

Max Bramer
Frederic Stahl (Eds.)

LNAI 15447

Artificial Intelligence XLI

44th SGAI International Conference
on Artificial Intelligence, AI 2024
Cambridge, UK, December 17–19, 2024
Proceedings, Part II

2
Part II



 Springer

Lecture Notes in Computer Science

Lecture Notes in Artificial Intelligence

15447

Founding Editor

Jörg Siekmann

Series Editors

Randy Goebel, *University of Alberta, Edmonton, Canada*

Wolfgang Wahlster, *DFKI, Berlin, Germany*

Zhi-Hua Zhou, *Nanjing University, Nanjing, China*

The series Lecture Notes in Artificial Intelligence (LNAI) was established in 1988 as a topical subseries of LNCS devoted to artificial intelligence.

The series publishes state-of-the-art research results at a high level. As with the LNCS mother series, the mission of the series is to serve the international R & D community by providing an invaluable service, mainly focused on the publication of conference and workshop proceedings and postproceedings.

Max Bramer · Frederic Stahl
Editors

Artificial Intelligence XLI

44th SGA I International Conference
on Artificial Intelligence, AI 2024
Cambridge, UK, December 17–19, 2024
Proceedings, Part II

Editors

Max Bramer
University of Portsmouth
Portsmouth, UK

Frederic Stahl
DFKI Niedersachsen
Oldenburg, Germany

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Artificial Intelligence
ISBN 978-3-031-77917-6 ISBN 978-3-031-77918-3 (eBook)
<https://doi.org/10.1007/978-3-031-77918-3>

LNCS Sublibrary: SL7 – Artificial Intelligence

© The Editor(s) (if applicable) and The Author(s), under exclusive license
to Springer Nature Switzerland AG 2025

This work is subject to copyright. All rights are solely and exclusively licensed by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

If disposing of this product, please recycle the paper.

Preface

Artificial Intelligence XLI comprises the refereed papers presented at the 44 SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence, held in December 2024. It is published as two volumes containing papers for the technical stream and the application stream, respectively. The conference was organised by SGAI, the British Computer Society Specialist Group on Artificial Intelligence. This year 80 papers were submitted and all were single-blind peer reviewed by either 2 or 3 reviewers plus the expert members of the Executive Program Committee for each stream of the conference.

This year's Donald Michie Memorial Award for the best refereed technical paper was won by a paper entitled 'NER Explainability Framework: Utilizing LIME to Enhance Clarity and Robustness in Named Entity Recognition' by Morten Grundetjern, Per-Arne Andersen, Morten Goodwin and Karl Audun Borgersen (University of Agder, Norway).

This year's Rob Milne Memorial Award for the best refereed application paper was won by a paper entitled 'Adaptive CNN Method For Prostate MR Image Segmentation Using Ensemble Learning' by Lars Jacobson, Mohamed Bader-El-Den, Adrian Hopgood, Shamsul Masum, Vincenzo Tamma (University of Portsmouth, UK), David Prendergast (Innovative Physics Ltd., UK) and Peter Osborn (Portsmouth Hospitals, University NHS Trust, UK).

The other technical stream full papers included are divided into sections on Neural Nets, Deep Learning, Large Language Models, Machine Learning, Evolutionary and Genetic Algorithms, and Knowledge Management. The other application stream full papers are divided into sections on Machine Vision, Evaluation of AI Systems, Applications of Machine Learning and Other AI Applications. Both volumes also include the text of short papers presented as posters at the conference.

On behalf of the conference Organising Committee, we would like to thank all those who contributed to the organisation of this year's programme, in particular the Program Committee members, the Executive Program Committees and our administrators Mandy Bauer and Bryony Bramer.

September 2024

Max Bramer
Frederic Stahl

Conference Administrator

Mandy Bauer BCS, UK

Paper Administrator

Bryony Bramer SGAI, UK

Technical Executive Program Committee

Max Bramer (Chair)	University of Portsmouth, UK
Frans Coenen	University of Liverpool, UK
Adrian Hopgood	University of Portsmouth, UK
John Kingston	Nottingham Trent University, UK
Jixin Ma	University of Greenwich, UK

Application Executive Program Committee

Frederic Stahl (Chair)	DFKI: German Research Center for Artificial Intelligence, Germany
Richard Ellis	RKE Consulting, UK
Rosemary Gilligan	SGAI, UK
Lars Nolle	Jade University of Applied Sciences, Germany
Richard Wheeler	University of Edinburgh, UK

Technical Program Committee

Per-Arne Andersen	University of Agder, Norway
Mercedes Arguello Casteleiro	University of Southampton, UK
Matt Armstrong-Barnes	HPE, UK
Juan Augusto	Middlesex University London, UK
Raed Sabri Hameed Batbooti	Southern Technical University/ Basra Engineering Technical College, Iraq
Karl Audun Borgersen	University of Agder, Norway
Soufiane Boulehouache	University of 20 Août 1955-Skikda, Algeria
Max Bramer	University of Portsmouth, UK
Ken Brown	University College Cork, Ireland
Marcos Bueno	Radboud University, The Netherlands
Darren Chitty	Aston University, UK

Frans Coenen	University of Liverpool, UK
Bertrand Cuissart	Université de Caen Normandie, France
Nicolas Durand	Aix-Marseille University, France
Frank Eichinger	DATEV eG, Germany
Michael Free	BT, UK
Martin Fyvie	Robert Gordon University, UK
Hossein Ghodrati Noushahr	University of Leicester, UK
Adrian Hopgood	University of Portsmouth, UK
Chris Huyek	Middlesex University London, UK
Mohamed Ihmeida	Buckinghamshire New University, UK
Stelios Kapetanakis	Distributed Analytics, UK
Mathias Kern	BT, UK
Ivan Koychev	University of Sofia, Bulgaria
Andrew Langworthy	BT, UK
Nicole Lee	University of Hong Kong, China
Haiming Liu	University of Southampton, UK
Jixin Ma	University of Greenwich, UK
Giovanna Martinez-Arellano	University of Nottingham, UK
Ken McGarry	University of Sunderland, UK
Silja Meyer-Nieberg	Universität der Bundeswehr München, Germany
Daniel Neagu	University of Bradford, UK
Lars Nolle	Jade University of Applied Sciences, Germany
Joanna Isabelle Olszewska	University of the West of Scotland, UK
Daniel O'Leary	University of Southern California, USA
Sanjib Raj Pandey	Royal Marsden NHS Foundation Trust, UK
Fernando Saenz-Perez	Universidad Complutense de Madrid, Spain
Pradeep Kumar Saraswathi	Salesforce, USA
Simon Thompson	GFT Technology, UK
M. R. C. van Dongen	University College Cork, Ireland

Application Program Committee

Manal Almutairi	University of Reading, UK
Saif Alzubi	University of Exeter, UK
Ines Arana	Robert Gordon University, UK
Vasileios Argyriou	Kingston University, UK
Juan Carlos Augusto	Middlesex University London, UK
Lakshmi Babu Saheer	Anglia Ruskin University, UK
Ken Brown	University College Cork, Ireland
Nikolay Burlutskiy	ContextVision AB, Sweden
Xiaochun Cheng	Swansea University, UK

Sarah Jane Delany	Technological University Dublin, Ireland
Tarek El-Mihoub	German Research Center for Artificial Intelligence GmbH (DFKI), Germany
Richard Ellis	RKE Consulting, UK
Ahmed Elsayed	German Research Center for Artificial Intelligence GmbH (DFKI), Germany
Xiaohong Gao	Middlesex University London, UK
Rosemary Gilligan	University of Hertfordshire, UK
John Gordon	AKRI Ltd, UK
Holmer Hemsen	German Research Center for Artificial Intelligence GmbH (DFKI), Germany
Chris Hinde	Loughborough University, UK
Chris Huyck	Middlesex University London, UK
Carl James-Reynolds	Middlesex University London, UK
Colin Johnson	University of Nottingham, UK
Stelios Kapetanakis	Distributed Analytics, UK
Mathias Kern	BT, UK
Daniel Lukats	German Research Center for Artificial Intelligence GmbH (DFKI), Germany
Christoph Manß	German Research Center for Artificial Intelligence GmbH (DFKI), Germany
Andre Miedtank	German Research Center for Artificial Intelligence GmbH (DFKI), Germany
Lars Nolle	Jade University of Applied Sciences, Germany
Sanjib Raj Pandey	Royal Marsden NHS Foundation Trust, UK
Jing Qi	University of Essex, UK
Robert Rettig	German Research Center for Artificial Intelligence GmbH (DFKI)
Sam Richardson	AstraZeneca, UK
Miguel A. Salido	Universitat Politècnica de València, Spain
Georgios Samakovitis	University of Greenwich, UK
Janina Schneider	German Research Center for Artificial Intelligence GmbH (DFKI), Germany
Frederic Stahl	German Research Center for Artificial Intelligence GmbH (DFKI), Germany
Daphne Theodorakopoulos	German Research Center for Artificial Intelligence GmbH (DFKI), Germany
Christoph Tholen	German Research Center for Artificial Intelligence GmbH (DFKI), Germany
Richard Wheeler	European Sustainable Energy Innovation Alliance, TU Graz, Austria

Contents – Part II

Application Papers

Adaptive CNN Method for Prostate MR Image Segmentation Using Ensemble Learning	3
<i>Lars E. O. Jacobson, Mohamed Bader-El-Den, Adrian A. Hopgood, Shamsul Masum, Vincenzo Tamma, David Prendergast, and Peter Osborn</i>	

Machine Vision

Optimizing Autonomous Vehicle Racing Using Reinforcement Learning with Pre-trained Embeddings for Dimensionality Reduction	21
<i>Martin Holen, Jayant Singh, Christian W. Omlin, Jing Zhou, Kristian M. Knausgård, and Morten Goodwin</i>	

Semantic Segmentation for Landslide Detection Using Segformer	35
<i>Hasnain Murtaza Syed, Mahdi Maktabdar Oghaz, and Lakshmi Babu Saheer</i>	

Vision-Based Human Fall Detection Using 3D Neural Networks	46
<i>Say Meng Toh, Na Helian, Kudiwa Pasipamire, Yi Sun, and Tony Pasipamire</i>	

Drone-Assisted Infrared Thermography and Machine Learning for Enhanced Photovoltaic Defect Detection: A Comparative Study of Vision Transformers and YOLOv8	59
<i>Ammar Memari and Tarek Debich</i>	

Evaluation of AI Systems

Evaluating Algorithms for Missing Value Imputation in Real Battery Data	75
<i>Dauda Nanman Sheni, Anton Herman Basson, and Jacomine Grobler</i>	

Using Pseudo Cases and Stratified Case-Based Reasoning to Generate and Evaluate Training Adjustments for Marathon Runners	88
<i>Ciara Feely, Brian Caulfield, Aonghus Lawlor, and Barry Smyth</i>	

Applications of Machine Learning

Emotion Detection in Hindi Language Using GPT and BERT 105
Ritika Agarwal and Noorhan Abbas

Classification and Recommendation of Mental Health Assistance Events
Using an RNN-LSTM, Fast-And-Frugal Trees and Weighted Sum System 119
Nathan R. Dickson and Nicholas H. M. Caldwell

Digit Detection: Localizing and Convoluting 133
Tyrell Martens and John Z. Zhang

Djinn—Data Journalism Interface for Newsgathering and Notifications 147
*Sara Elo Dean, Lars Adrian Giske, Herman Jangsett Mostein,
Silvia Podestà, Halvor Helland Barndon, Sara Stegane,
and Henrik Nordberg*

Advancing Financial Text Sentiment Analysis with Deep Learning
and Ensemble Models 162
Wei Liang Russell Tang

Other AI Applications

Explaining a Staff Rostering Problem Using Partial Solutions 179
*GianCarlo A. P. I. Catalano, Alexander E. I. Brownlee, David Cairns,
John A. W. McCall, Martin Fyvie, and Russell Ainslie*

Formalise Regulations for Autonomous Vehicles with Right-Open
Temporal Deontic Defeasible Logic 194
Pak Yin Chan, Xue Li, Yiwei Lu, Yuhui Lin, and Alan Bundy

SLANGO - The Initial Blueprint of Privacy-Oriented Legal Query
Assistance: Exploring the Potential of Retrieval-Augmented Generation
for German Law Using SPR 208
Jérôme Agater and Ammar Memari

Short Application Papers

An Ensemble Modelling of Feature Engineering and Predictions
for Enhanced Fake News Detection 225
Patricia Asowo, Sangeeta Lal, and Uchenna Daniel Ani

A Child-Robot Interaction Experiment to Analyze Gender Stereotypes in the Perception of Mathematical Abilities	232
<i>Madalina Croitoru, Pablo Laviron, Sio Bando, Eric Gilles, Amine Miled, Royce Anders, Nathalie Blanc, Gowrishankar Ganesh, and Emmanuelle Brigaud</i>	
Reinforcement Learning for Patient Scheduling with Combinatorial Optimisation	238
<i>Xi Liu, Changgang Zheng, Zhen Chen, Yong Liao, Ren Chen, and Shufan Yang</i>	
Nursing Activity Recognition for Automated Care Documentation in Clinical Settings	244
<i>Frank Wallhoff and Fenja T. Hesselmann</i>	
Exploring Efficient Job Shop Scheduling Using Deep Reinforcement Learning	251
<i>Reshma Maharjan, Per-Arne Andersen, and Lei Jiao</i>	
Respiratory Disease Detection Using Deep Convolutional Transformer Models	258
<i>Holly Burrows, Mahdi Maktabdar Oghaz, and Lakshmi Babu Saheer</i>	
Evaluating the Performance of LLMs When Translating Saudi Arabic as Low Resource Language	264
<i>Salwa Alahmari, Eric Atwell, Mohammad Alsalka, and Hadeel Saadany</i>	
Bi-directional LSTM Applied to the Maritime Target Motion Analysis Problem	270
<i>Lars Nolle, Nils Meinardus, Martin Kumm, and Christoph Tholen</i>	
Author Index	277

Contents – Part I

Technical Papers

NER Explainability Framework: Utilizing LIME to Enhance Clarity and Robustness in Named Entity Recognition	3
<i>Morten Grundetjern, Per-Arne Andersen, Morten Goodwin, and Karl Audun Borgersen</i>	

Neural Nets

Revealing Limitations of ResNet Models for Deep Evaluation in Chess	19
<i>Jakub Zeman and Ladislava Smítková Janků</i>	

Quasi Biologically Plausible Category Learning	33
<i>Christian Huyck</i>	

On the Development of a Pixel-Wise Plastic Waste Identification System for Multispectral Remote Sensing Applications	47
<i>Christoph Tholen, Eike Rodenbäck, Lars Nolle, Robert Rettig, and Frederic Stahl</i>	

Streamlining Attention for Text Classification: Sequence Length Reduction with Pooling Attention	61
<i>Daniel Biermann, Fabrizio Palumbo, Morten Goodwin, and Ole-Christoffer Granmo</i>	

LSTM for Modelling and Predictive Control of Multivariable Processes	74
<i>Krzysztof Zarzycki and Maciej Ławryńczuk</i>	

Structured Radial Basis Function Network: Modelling Diversity for Multiple Hypotheses Prediction	88
<i>Alejandro Rodriguez Dominguez, Muhammad Shahzad, and Xia Hong</i>	

Deep Learning

Bitcoin Forecasting Using Deep Learning and Time Series Ensemble Techniques	105
<i>Huma Zafar and Stylianos Kapetanakis</i>	

TRAPL: Transformer-Based Patch Learning for Enhancing Semantic Representations Using Aggregated Features to Estimate Patch-Class Distribution	116
<i>Sander Riisøen Jyhne, Per-Arne Andersen, Ivar Oveland, and Morten Goodwin</i>	
DATE: Derivative Alignment Training for Extrapolation with Neural Networks	130
<i>Enrico Lopedoto, Tillman Weyde, and Kizito Salako</i>	
Interactive Simulator Framework for XAI Applications in Aquatic Environments	144
<i>Ahmed H. Elsayed, Tarek A. El-Mihoub, Christoph Manss, Andre Miedtank, Lars Nolle, and Frederic Stahl</i>	
Detection of Vascular Leukoencephalopathy in CT Images	158
<i>Zuzana Cernekova, Viktor Sisik, and Fatana Jafari</i>	
Large Language Models	
PlanBERT: From Messy Zonal Plans to Informative Vector Embeddings	175
<i>Henrik Brådlund, Morten Goodwin, Per-Arne Andersen, and Alexander S. Nossun</i>	
ArgueMapper Assistant: Interactive Argument Mining Using Generative Language Models	189
<i>Mirko Lenz and Ralph Bergmann</i>	
Machine Learning	
Contextual Transformers for Goal-Oriented Reinforcement Learning	207
<i>Oliver Dippel, Alexei Lisitsa, and Bei Peng</i>	
Localized Affinity-Based Reinforcement Learning for Interpretable State-Specific Decision-Making	221
<i>Ajay Vishwanath and Christian Omlin</i>	
Navigating the Landscape of Case Fidelity and Competence in Case-Based Reasoning	235
<i>Adwait P. Parsodkar, Deepak P., and Sutanu Chakraborti</i>	

Evolutionary and Genetic Algorithms

Tree-Based Genetic Programming for Evolutionary Analog Circuit with Approximate Shapley Value	253
<i>Xinming Shi, Leandro L. Minku, and Xin Yao</i>	
A Dominance-Based Surrogate Classifier for Multi-objective Evolutionary Algorithms	268
<i>Tiwonge Msulira Banda and Alexandru-Ciprian Zăvoianu</i>	

Knowledge Management

A Homogeneous Approach to Reasoning Over Global Geographic Data	285
<i>Alia I. Abdelmoty and Abdurauf Satoti</i>	

Short Technical Papers

OK Google, What is the Stock Forecast for Next week? Leveraging Search Engines for Data Collection, Sentiment Analysis and Stock Predictions	301
<i>Nicholas Arthur Frederick-Preece and Noorhan Abbas</i>	
University News: A New Data Source for NLP Bias Research	307
<i>Rawan Bin Shiha, Eric Atwell, and Noorhan Abbas</i>	
Enhancing Nepali Text Understanding with Machine Translation and LoRA Fine-Tuning of Open-Source LLM	313
<i>Kshitiz Rimal and Noorhan Abbas</i>	
Audio-Visual Emotion Recognition Using Deep Learning Methods	320
<i>Mukhambet Tolegenov, Lakshmi Babu Saheer, and Mahdi Maktabdar Oghaz</i>	
Spatial Interpolation of Air Quality: A UK Case Study	327
<i>Lorenzo Garbagna, Praseed Melethil, Lakshmi Babu Saheer, and Mahdi Maktabdar Oghaz</i>	
Talk like a Local: Evaluating Large Language Models for Arabic Dialect Translation Using Similarity Scores	333
<i>Alaa Bouomar and Noorhan Abbas</i>	
On Monadic Binary, with Application to Machine Understanding	339
<i>M. J. Wheatman</i>	

A Proposed ELM Ensemble Approach for Predicting Railway Delays	346
<i>Matthew Day</i>	
Semantic Bone Structure Segmentation in 2D Image Data: Towards Total Knee Arthroplasty	352
<i>Tobias Neiss-Theuerkauff, Arne Schierbaum, Thomas Luhmann, Till Sieberth, and Frank Wallhoff</i>	
Author Index	359

Application Papers



Adaptive CNN Method for Prostate MR Image Segmentation Using Ensemble Learning

Lars E. O. Jacobson¹(✉), Mohamed Bader-El-Den¹, Adrian A. Hopgood¹, Shamsul Masum¹, Vincenzo Tamma¹, David Prendergast², and Peter Osborn³

¹ University of Portsmouth, Portsmouth, UK
{lars.jacobson,mohamed.bader,adrian.hopgood,shamsul.masum,
vincenzo.tamma}@port.ac.uk

² Innovative Physics Ltd., Portsmouth, UK
david.prendergast@inphys.com

³ Portsmouth Hospitals, University NHS Trust, Portsmouth, UK
peter.osborn@porthosp.nhs.uk

Abstract. In 2020, there were more than 1.4 million new cases of prostate cancer worldwide, and more than 375,000 deaths from the disease. The conventional diagnostic pathway hinges on the assessment of prostate-specific antigen (PSA) levels and the conduct of trans-rectal ultrasound (TRUS)-guided biopsies. However, the specificity of PSA as a biomarker is notably low, at approximately 36%, due to its elevation in benign prostatic conditions, underscoring the imperative for more precise diagnostic modalities. This research leverages a dataset comprising T2-weighted magnetic resonance (MR) images from 1,151 patients, totaling 61,119 images, to refine prostate cancer diagnostics. This paper introduces methodology that utilises knowledge-based artificial intelligence (AI) frameworks with image segmentation techniques to enhance the accuracy of prostate cancer detection. The approach in this paper focuses on the segmentation of MR images into distinct anatomical zones of the prostate - specifically, the transition zone (TZ) and peripheral zone (PZ). The variations of model produce a Dice Similarity Coefficient in the range of 0.373–0.544 in the 95th percentile. This segmentation is critical for the automation and augmentation of diagnostic precision in prostate cancer. This approach not only aims to improve the specificity and sensitivity of prostate cancer diagnostics but also to facilitate the exploitation of publicly accessible datasets for research advancements in this domain.

Keywords: Image segmentation · Prostate · Magnetic resonance imaging · U-net

1 Introduction

In 2020, there were more than 1.4 million new cases of prostate cancer worldwide, and more than 375,000 deaths from the disease [1]. The prevalence of this

malignancy notably increases in men aged 65 and above. There is a pressing need for the development of non-invasive diagnostic modalities that can accurately differentiate the severity of prostate cancer beyond the conventional approach of active surveillance [2]. The traditional diagnostic pathway for prostate cancer hinges on the assessment of prostate-specific antigen (PSA) levels and the execution of trans-rectal ultrasound (TRUS)-guided biopsies. Despite its widespread use, PSA screening is plagued by a low specificity rate of approximately 36%, due to the elevation of PSA levels in benign prostatic conditions [3]. This diagnostic ambiguity highlights the inherent limitations of PSA as a reliable biomarker, where elevated levels do not conclusively indicate the presence of a tumor, nor do normal levels definitively exclude it. Moreover, the TRUS-guided biopsy, which predominantly targets the peripheral aspects of the gland, suffers from methodological drawbacks. Given that 30–40% of prostate cancers originate in the anterior mid-line transition zone (TZ), a significant proportion of tumors may elude detection with this approach due to the systematic but random sampling of the peripheral zone (PZ), compounded by ultrasound’s poor capability in distinguishing cancerous tissues from benign ones [4]. Against this backdrop, magnetic resonance (MR) imaging emerges as an alternative, offering the potential to significantly enhance diagnostic accuracy.

MR imaging’s advanced capabilities in tumor detection both pre- and post-biopsy set the stage for a shift in prostate cancer diagnostics. The high-resolution imaging and detailed tissue characterisation afforded by MR imaging not only facilitate the precise localisation of tumors within the prostate gland but also aid in the assessment of tumor aggressiveness. This technological advancement underscores the imperative for integrating MR imaging into the diagnostic workflow, promising a leap towards more accurate, timely, and non-invasive detection of prostate cancer, thereby addressing the critical limitations of current diagnostic practices [5–8].

The process description and the semantics of the prostate cancer use case have been detailed in Fig. 1. The Prostate Imaging Reporting and Data System (PI-RADS) represents a significant advancement in the domain of prostate cancer diagnostics, providing a standardised framework for the evaluation and characterisation of prostate tumors via MR imaging. This scoring system, ranging from 1 to 5, facilitates the stratification of cancer risk based on MRI findings, with the score being adjudicated by clinicians through a detailed analysis of the imaging characteristics and anatomical location of lesions within the prostate gland. Specifically, the PI-RADS methodology involves the delineation of the prostate into its two main anatomical zones—the transition zone (TZ) and the peripheral zone (PZ)—as identified on MR images. The assignment of a PI-RADS score is contingent upon the clinician’s interpretation of the MR images, focusing on the distinctive features observed in these two zones. This approach enhances the precision of prostate cancer detection and characterisation, allowing for a more nuanced understanding of the disease’s severity and potential behavior.

The integration of knowledge-based artificial intelligence (AI) into the medical domain holds transformative potential for enhancing the early diagnosis of

various diseases, including cancer [9]. A pivotal area within AI that has demonstrated significant utility in diagnostic processes is computer vision, specifically through the extraction and analysis of features from medical imagery. Convolutional Neural Networks (CNNs), a class of deep neural networks, are at the forefront of this technological evolution. These networks are adept at learning disease-indicative features from vast datasets of medical images.

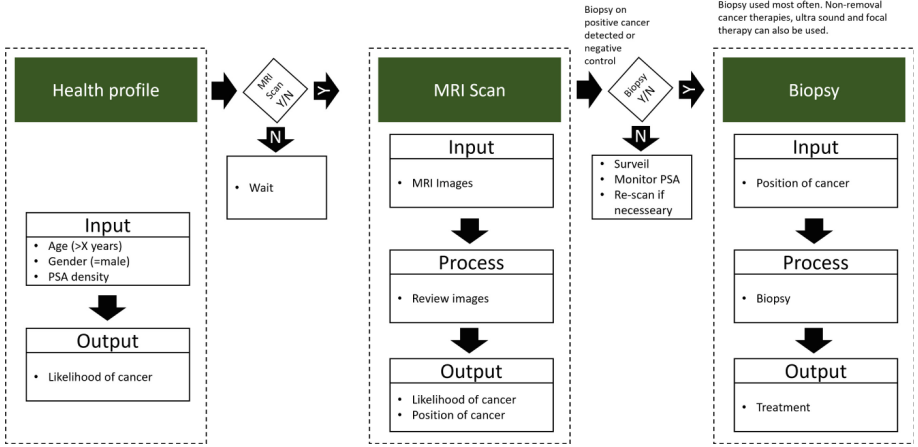


Fig. 1. Process description of prostate cancer procedure and diagnosis. The health profile of the patient is the initial input where properties e.g. age and PSA density are evaluated to determine if the patient should progress in the process. A MRI scan is executed and from this the likelihood and position of cancer is retrieved. For the final step, biopsy is performed with input from the previous steps [10].

The unique strength of CNNs lies in their ability to discern and interpret subtleties in imaging data that may be imperceptible to the human eye. This capability not only augments the accuracy of disease diagnosis but also posits the potential for CNN-driven diagnostic systems to surpass the diagnostic efficacy of human physicians [11]. Such advancements underscore the critical role of advanced AI methodologies in revolutionising medical diagnostics, enabling earlier detection of conditions like cancer with unprecedented precision [11].

The U-net architecture is particularly esteemed for its proficiency in medical imaging applications, including the segmentation of MR images. The U-net design is bifurcated into two distinct segments: the contracting (encoding) path and the expansive (decoding) path, catering to the nuanced demands of medical image analysis [12]. The contracting path mirrors a conventional CNN structure, where each block is composed of a sequence of operations starting with two consecutive convolutional layers. Each convolution is followed by the activation function, Rectified Linear Unit (ReLU), to introduce non-linearity and enhance feature learning capabilities. This is then succeeded by a max-pooling layer which

serves to down-sample the feature map, thereby reducing its dimensions while retaining the most salient features [13]. Transitioning to the expansive path, it is characterised by a series of up-sampling stages that progressively increase the spatial dimensions of the feature maps. This is achieved through 2×2 up-convolutions, which effectively reconstruct the image details from the compressed feature representation. The expansive path essentially reverses the operation of the contracting path, aiming to restore the image to its original resolution in the context of segmentation tasks. Both paths are conceptually described as encoding and decoding phases, respectively. Throughout these phases, multiple convolutional operations are employed, each with a specified kernel size and followed by a ReLU activation function. This sequence is designed to capture and refine the intricate patterns and structures within the medical images [14]. A notable feature of the U-net architecture is the implementation of skip connections between the encoding and decoding paths. These connections are pivotal in mitigating the loss of information typically associated with deep convolutional operations. By bridging the gap between the corresponding layers of the encoding and decoding paths, skip connections facilitate the direct flow of information, allowing the network to preserve and utilise fine-grained details essential for accurate segmentation. This architectural blueprint of U-net, with its synergistic components, underscores its unparalleled capability in medical image segmentation. It adeptly addresses the challenges of detail preservation and feature extraction, making it an indispensable tool in the enhancement of diagnostic accuracy through advanced imaging techniques [14].

In evaluating the performance of the medical imaging prostate segmentation model, several metrics are employed to assess the accuracy and robustness of the segmentation results. The Dice Similarity Coefficient (DSC) quantifies the overlap between the predicted and ground truth segmentation, calculated as $DSC = \frac{2 \times TP}{2 \times TP + FP + FN}$, where TP represents true positives, FP false positives, and FN false negatives. The Relative Volume Difference (RVD) measures the discrepancy in volume between the predicted and ground truth segmentation, expressed as $RVD = \frac{V_p - V_{gt}}{V_{gt}}$, where V_p and V_{gt} denote the volumes of the predicted and ground truth segmentation, respectively. The Hausdorff Distance (HD) captures the maximum distance between corresponding points in the predicted and ground truth segmentation, while the Average Surface Distance (ASD) computes the average distance between the surfaces of the two segmentation. These metrics collectively provide a comprehensive assessment of the segmentation model's performance in accurately delineating prostate boundaries in medical images [15].

Litjens et al. [16] leverage MR image segmentation combined with PI-RADS classification to investigate automated prostate cancer diagnosis. The results from the system were compared with the radiologists' opinions and were validated for 347 patients. The system did not show any significant difference in performance from the radiologists at high specificity but at lower specificity the radiologists performed significantly better. In order to evaluate this, ROC analysis was used to classify patient having prostate cancer or not. Masoudi et al. [17]

further support the potential of deep learning applications in prostate cancer research. Bardis et al. confirm deep learning methods can segment the prostate into TZ and PZ. Furthermore, they show that using three U-Nets can produce a near radiologist level of performance. To improve the highlighted zones' detection, pre-processing the MR images can enhance the system's specificity. The test data produced a mean Dice score of 0.940 (inter-quartile range, 0.930–0.961), and the Pearson correlation coefficient for volume was 0.981. Luo et al. [7] show that a weighted low-rank matrix restoration algorithm (RLRE) can improve MRI images' display effect and resolution.

Class imbalance is a prevalent issue in deep learning-based classifiers, particularly manifesting when certain classes within the training dataset significantly outnumber others. This imbalance not only hampers the model's convergence during the training phase but also affects its ability to generalise effectively during testing [18]. Garcia et al. [19] delineate two principal strategies to mitigate class imbalance: data level methods and classifier level methods. Data level methods aim to adjust the training set's class distribution directly through techniques such as under-sampling (reducing the number of examples in dominant classes) or oversampling (increasing the number of examples in minority classes). On the other hand, classifier level methods focus on modifying the training algorithm to better handle class imbalance, without altering the distribution of the training data [20]. In the realm of medical imaging, random minority oversampling is a widely adopted technique, which involves duplicating randomly selected samples from underrepresented classes to balance the dataset [21]. While effective in addressing class imbalance, this approach raises concerns about over-fitting [22], prompting the exploration of more sophisticated sampling methods. One such method is the Synthetic Minority Over-sampling Technique (SMOTE), which generates synthetic examples by interpolating between neighboring data points of minority classes. This technique aims to enhance the diversity within the underrepresented classes, thereby reducing the risk of over-fitting associated with traditional oversampling [23].

Further advancements in addressing class imbalance have leveraged machine learning algorithms like Random Forest to perform oversampling within a classification ensemble, providing a nuanced approach to generating samples for minority classes [24]. These innovative methods underscore the evolving landscape of strategies designed to tackle the challenges posed by class imbalance in deep learning models, particularly in the context of medical image analysis. Ensemble methods have emerged as a powerful strategy in machine learning to improve the predictive performance and robustness of models by combining multiple learners. Among these methods, majority voting is a simple yet effective technique, particularly in the domain of image segmentation, where the consensus across an ensemble of models is used to make the final prediction [25]. This approach leverages the diversity among the ensemble members to reduce over-fitting and increase the generalisation ability of the model. Specifically, in image segmentation tasks, majority voting has been applied to consolidate pixel-wise predictions from multiple segmentation models, thereby enhancing the accuracy and reliability of the segmen-

tation outcome [26]. This paper extends the work proposed by Jacobson et al. [10] including additional results and recent advancements.

Recent advancements have enabled the vectored implementation of majority voting, significantly reducing computational overhead and improving efficiency. The vectored approach takes advantage of modern hardware architectures and software optimisations to perform operations on entire arrays of data in a single step, as opposed to traditional iterative or loop-based methods. This enhancement is particularly beneficial in processing large datasets common in medical imaging and remote sensing applications, where timely and accurate segmentation is crucial [27]. The integration of vectored majority voting into image segmentation workflows represents a confluence of theoretical elegance and practical efficiency, offering a scalable solution to the challenges of contemporary segmentation tasks.

2 Methods

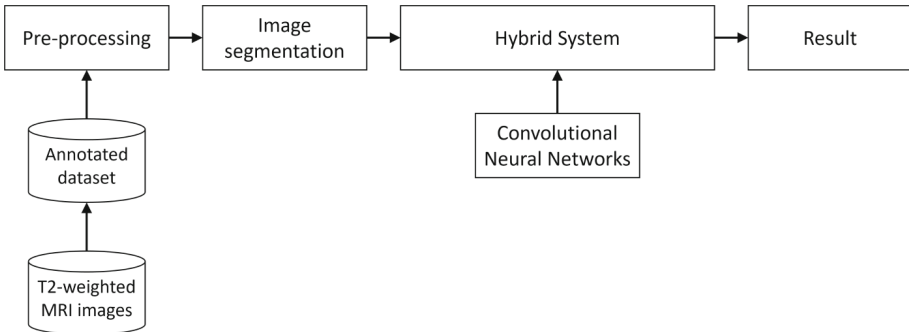


Fig. 2. Proposed model. The proposed model uses T2-weighted MR images together with annotations of the segmented prostate in TZ and PZ. Training of the model generates the model of which the test images can be processed and produce masks of the prostate zones.

The proposed method and algorithm flow is illustrated in Fig. 2, i.e. i) pre-processing of MR images to normalise quality, ii) segment the prostate into TZ and PZ using a single class. The t2-weighted MR images from the dataset are annotated and pre-processed for use as a training dataset.

2.1 Data Description and Preparations

The data relate to prostate cancer medical applications and have been sourced from The Cancer Imaging Archive (TCIA), which is publicly available [28]. The data set used for prostate cancer consists of 61,119 t2-weighted MR images for 1,151 patients. The MR images used are not pre-annotated with TZ or PZ.

Figure 3 displays the TZ and PZ on an MR image. The annotations used in this paper have been produced in collaboration with expert radiologists. Each recurrence of any of the zones is required to perform PI-RADS scoring [17, 29] and requires each MR image per patient to be labelled appropriately. Typically, in TCIA [28] data sets, each patient has a collection of MR images with a median of 60 slices or images. Additionally, each patient has been evaluated using the UCLA (University of California, Los Angeles) prostate cancer index following PI-RADS v2 [28].

A focal loss function has been implemented to address class imbalance. The focal loss function calculates a weighted loss based on the predicted probabilities of each class, emphasising challenging samples through the manipulation of a focusing parameter. This focal loss function aims to improve the model’s performance by penalising misclassifications, particularly focusing on the minority class. In addition to the focal loss function, class weights are calculated to further mitigate the effects of class imbalance. For each class i , the class weight w_i is computed as $w_i = \frac{N}{k \times N_i}$, where N is the total number of pixels across all masks, N_i is the number of pixels belonging to class i , and k is the total number of classes.

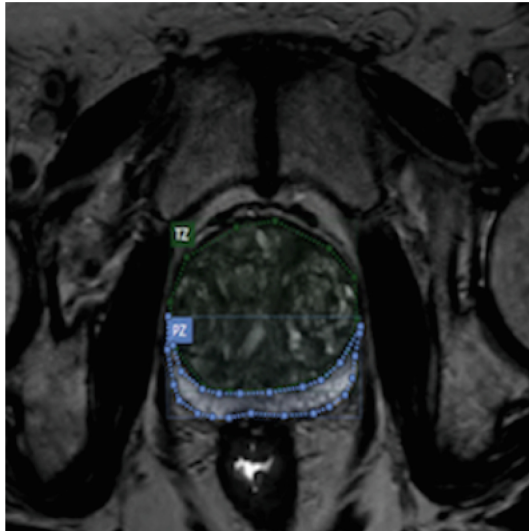


Fig. 3. Transition zone (TZ) and peripheral zone (PZ) annotated combined with the original image.

Some patients have been evaluated multiple times in the data set but at different times, and can then show a change in the scoring. A characteristic of the data is that most patients are evaluated with a PI-RADS score of 3 (intermediate), 4 (high), or 5 (very high), showing a bias towards positive cancer cases. The patients’ PSA levels are distributed mainly between 0 and 20 ng/mL.

For example, Fig. 4 shows a t2 weighted MR image that hints at a tumour in the highlighted area. It is difficult to confirm a tumour from this image alone. However, using this evidence to support other techniques provides the possibility of an MRI-guided prostate biopsy that allows for more accurate targeting [30].

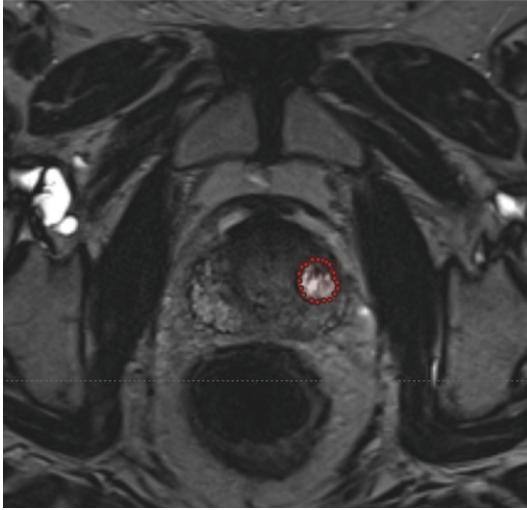


Fig. 4. Axial t2-weighted MR image showing subtle low-signal-intensity area in anterior peripheral zone. Source: [28].

2.2 Model Architecture and Training

The U-net architecture is encompassing nine distinct stages within both the encoding and decoding segments. This research capitalizes on the foundational U-net model, as illustrated in Fig. 5. The U-net architecture is encompassing nine distinct stages within both the encoding and decoding segments. Integral to this design is the incorporation of skip connections at every stage, bridging the encoding and decoding components. These connections serve a dual purpose: firstly, they expedite the model’s convergence by facilitating the direct flow of information across the network. This is critical for deep learning models, where the depth of the architecture can often slow down the training process. Secondly, the skip connections play a pivotal role in mitigating information loss throughout the network’s depth. As data traverses through the successive stages of encoding and decoding, there’s an inherent risk of diluting important features and details—skip connections counteract this by preserving and reintegrating essential information back into the network. This element ensures that the U-net model maintains a high degree of fidelity in the information processed, thereby enhancing its effectiveness in medical image segmentation tasks, such as those

central to this research. In this paper, five variations of deep learning models are explored for image segmentation, each tailored to address different aspects of the segmentation challenge. These models are differentiated by their handling of class types, class weighting, loss functions, and input data format.

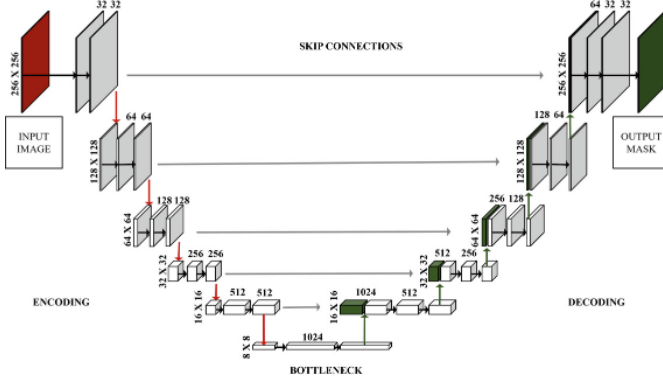


Fig. 5. The proposed U-Net neural network architecture localizes the prostate by creating a bounding box around it and narrowing the field of view. It consists of nine layers comprising convolutions. Both the contraction and expansion pathways utilize $1 \times 3 \times 3$ and $2 \times 1 \times 1$ filters for convolutional kernels. The image is downsampled to a $1 \times 1 \times 1$ matrix before upsampled.

Model 1: Single Class focuses on binary segmentation using a U-Net architecture. It employs a standard binary cross-entropy loss function and an Adam optimiser. The model is notable for its simplicity and is specifically designed for segmenting images into foreground and background classes without considering multiple object categories. The U-Net architecture utilised here is characterised by its encoding-decoding structure with skip connections, enabling precise localisation.

Model 2: Class Weights introduces class weights into the U-Net model to address class imbalance, a common issue in medical image segmentation. This model uses categorical cross-entropy loss and dynamically calculates class weights based on the training data distribution. By adjusting the loss function to emphasise minority classes, the model aims to improve segmentation performance on underrepresented classes.

Model 3: Focal Loss modifies the U-Net model by incorporating a focal loss function, designed to focus training on hard-to-classify examples. This approach was explored because of the significant imbalance between the foreground and background classes. The focal loss adds a modulating factor to the traditional cross-entropy loss, reducing the loss contribution from easy examples and allowing the model to focus on challenging areas of the image.

Model 4: Multiple Class (RGB) extends the U-Net model to handle multi-class segmentation with RGB input data. This model processes images and masks with multiple classes, utilising one-hot encoding for the masks and a softmax activation in the final layer for multi-class classification. The adaptation to RGB inputs allows the model to leverage color information because the dataset has color differences delineate class boundaries.

Model 5: Multiple Class (RGBA) further extends the multi-class segmentation approach by accommodating RGBA input images, effectively handling transparency in addition to color. This model incorporates data augmentation techniques and class weight calculation to address class imbalance and overfitting, increasing robustness for complex segmentation tasks with RGBA images. It also utilises a combination of callbacks including model checkpoints and learning rate reductions to optimise training outcomes.

Each model variant presents a unique approach to segmentation, from simple binary classification to complex multi-class segmentation with enhanced input data handling and loss function adjustments. The evolution from Model 1 through Model 5 demonstrates a progressive enhancement in addressing specific challenges such as class imbalance, hard examples, and multi-class segmentation, showcasing the adaptability of the U-Net architecture to various segmentation tasks. The segmentation framework employs an ensemble method to enhance prediction accuracy and robustness across a set of models. To accommodate variations in image and mask resolutions, a resising operation, `resize_mask`, is applied to standardise the dimensions of all masks to a predetermined output shape, typically (256, 256), though adjustable based on specific requirements. Moreover, the ensemble predictions are derived through a vectorised majority voting mechanism, `majority_vote_vectorised`, applied to model predictions. This method consolidates individual model outputs by selecting the most frequent prediction for each pixel across the ensemble, effectively mitigating outliers and leveraging collective model intelligence.

3 Results

This paper uses an MR image dataset obtained from patients with biopsy-confirmed prostate cancer. The annotations are stored in an annotation format with a link to the original DICOM file. The pre-processing of the MR images is presented in Fig. 6.

In comparison to existing research utilising similar evaluation metrics for medical imaging prostate segmentation, found in Table. 1, this study shows notable similarities and differences. This segmentation model achieved a Dice Similarity Coefficient (DSC) of 0.544 (Model 1: Single Class), which is lower than the 0.904 reported by RAU-Net [31] and the 0.901 reported by DenseU-Net [32]. This indicates inferior overlap between predicted and ground truth segmentations in this study. Moreover, the Relative Volume Difference (RVD) in this Model 5: Multiple Class (RGBA) was 0.339, which is comparable to the previous studies but still higher than the 0.026 reported by ConvLSTMs and

Table 1. Results of evaluation metrics. This table provides detailed performance metrics for each model variation used in this study. The models include Single Class, Class Weights, Focal Loss, Multiple Class (RGB), Multiple Class (RGBA), and Ensemble predictions. The evaluation metrics include Dice Similarity Coefficient (DSC, %), Relative Volume Difference (RVD, %), Hausdorff Distance (HD, mm), and Average Surface Distance (ASD, mm). Model 1: Single Class achieves a DSC of 0.544, RVD of 0.822, HD of 9.825 mm, and ASD of 91.182 mm. Model 5: Multiple Class (RGBA) shows a notable RVD of 0.339 and an ASD of 39.278 mm. The Ensemble model achieves the best HD of 3.000 mm. These results illustrate the comparative performance of each model variant, providing insights into the strengths and limitations of each approach.

	DSC	RVD (%)	HD (mm)	ASD (mm)
Model 1: Single Class				
Mean	0.379	0.822	9.825	91.182
(CI) 95%	0.544	0.861	11.059	94.827
Model 2: Class Weights				
Mean	0.297	0.824	11.171	91.182
(CI) 95%	0.427	0.861	11.277	94.827
Model 3: Focal Loss				
Mean	0.230	0.867	11.144	69.878
(CI) 95%	0.395	0.911	11.273	94.109
Model 4: Multiple Class (RGB)				
Mean	0.226	0.871	11.063	63.015
(CI) 95%	0.374	0.911	11.270	93.864
Model 5: Multiple Class (RGBA)				
Mean	0.170	0.339	22.775	39.278
(CI) 95%	0.239	0.576	31.649	49.815
Ensemble predictions				
Mean	0.162	0.565	3.000	55.899
(CI) 95%				
Results from previous research				
RAU-Net [31]				
Mean	0.904		10.962	
(CI) 95%				
DenseU-Net [32]				
Mean	0.901		8.846	
(CI) 95%				
RDAU-Net [33]				
Mean	0.898		7.872	
(CI) 95%				
U-Net [34]				
Mean	0.897		8.916	
(CI) 95%				
nnUNet (3D) [35]				
Mean	0.823		6.046	
(CI) 95%	0.804, 0.842		5.333, 6.759	
ConvLSTMs and GGNN [36]				
Mean	0.918	0.026	10.36	1.73
(CI) 95%				

CGNN [36], indicating a closer agreement in volume estimation compared to prior studies. Additionally, the Hausdorff Distance (HD) and Average Surface Distance (ASD) in these results were notably increased. This study’s ensemble model recorded an HD of 3.000 mm and an ASD of 55.899 mm, compared to RAU-Net’s 10.962 mm HD and nnUNet (3D)’s 6.046 mm ASD. These increased values indicate reduced precision in delineating prostate boundaries.

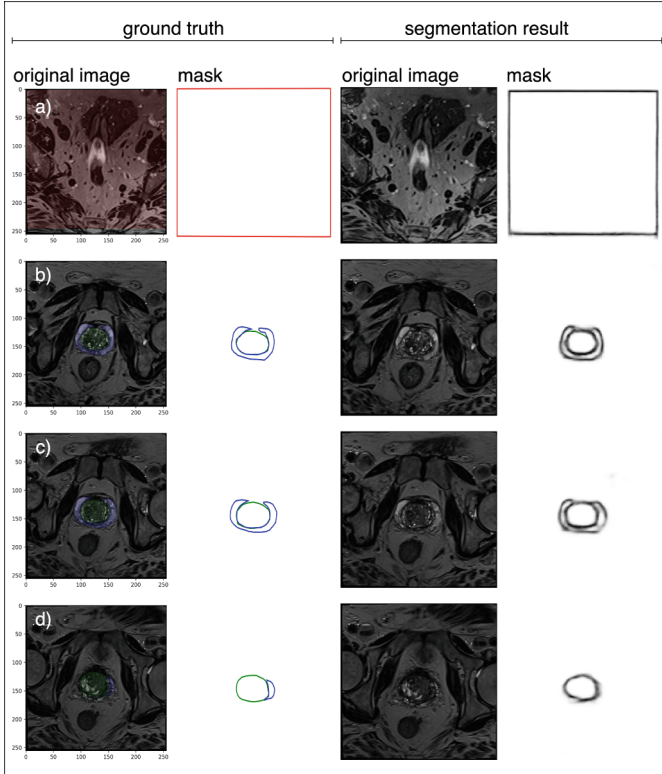


Fig. 6. (a) & (d) are the top & bottom part of the prostate where (a) successfully predicts that no prostate is present & (d) predicts the TZ and traces of PZ. The prediction of presence of prostate is visualised with a annotated square tracing the outskirts of the image. (b) & (c) are the middle part of the prostate [28].

This comparative analysis highlights the effectiveness of this proposed segmentation approach, underscoring its potential for enhancing clinical decision-making and patient care in prostate cancer diagnosis and treatment planning [31,32]. One of the key characteristics regarding the data sets of previous studies is that either the MR images not containing a prostate are excluded from the processed data set or excluded manually as part of the segmentation process. This notably affects the evaluation of the model and is predominantly noticeable in DSC metrics. In this study, Model 1: Single Class outperforms the other

variations of models produced regarding DSC and HD, achieving a DSC of 0.544 and an HD of 9.825 mm, which follows the accuracy of the greyscale zones. However, Model 5: Multiple Class (RGBA) notably outperforms the other models on RVD and ASD, with an RVD of 0.339 and an ASD of 39.278 mm. The ensemble predictions produce a combined result and present an improvement in HD with a value of 3.000 mm. Since the models are using different image types, the ensemble learning is not beneficial for all other evaluation metrics.

4 Conclusion

Complete and reliable segmentation into TZ and PZ is required in order to automate and enhance the process of localising prostate cancer. This paper proposes an approach to applying a knowledge base of domain expertise and visual properties together with image segmentation to improve the diagnosis of prostate cancer using publicly available data. The contribution of this paper is to provide a system to analyse and classify prostate cancer using MR images. The trained model is based on domain expert knowledge and a developed set of rules. It brings together existing work on image segmentation and industry knowledge. By combining the complementary advantages of the approaches, this research aims to overcome the limitations of single-sided segmentation methods and achieve increased performance in prostate MR image segmentation. The trained model can using a single class detect I) identify if a prostate is present & II) segment the prostate into TZ and PZ. Experimental results demonstrate that the proposed model has the potential to produce satisfactory results and, together with expert knowledge, achieve additional useful understanding of the field. Future work includes improving the image segmentation with additional classes to further mitigate class imbalance. This approach can be further extended and refined to address other similar challenges in medical imaging research.

References

1. WCRF International. Prostate cancer statistics: World Cancer Research Fund International (2022). <https://www.wcrf.org/cancer-trends/prostate-cancer-statistics/>
2. Schröder, F.H., et al.: Screening and prostate-cancer mortality in a randomized European study. *New Engl. J. Med.* **360**(13), 1320–1328 (2009)
3. Barentsz, J.O., et al.: ESUR prostate MR guidelines 2012. *Eur. Radiol.* **22**(4), 746–757 (2012)
4. Ahmed, H.U., et al.: Is it time to consider a role for MRI before prostate biopsy? *Nat. Rev. Clin. Oncol.* **6**(4), 197–206 (2009)
5. Huang, S., Yang, J., Fong, S., Zhao, Q.: Artificial intelligence in cancer diagnosis and prognosis: opportunities and challenges. *Cancer Lett.* **471**, 61–71 (2020)
6. Murphy, G., Haider, M., Ghai, S., Sreeharsha, B.: The expanding role of MRI in prostate cancer. *AJR Am. J. Roentgenol.* **201**(6), 1229–38 (2013)
7. Luo, R., Zeng, Q., Chen, H.: Artificial intelligence algorithm-based MRI for differentiation diagnosis of prostate cancer. *Comput. Math. Methods Med.* (2022)






8. Lawrentschuk, N., et al.: 'Prostatic evasive anterior tumours': the role of magnetic resonance imaging. *BJU Int.* **105**(9), 1231–1236 (2010)
9. Ardila, D., et al.: End-to-end lung cancer screening with three-dimensional deep learning on low-dose chest computed tomography. *Nat. Med.* **25**(6), 954–961 (2019)
10. Jacobson, L.E., Hopgood, A.A., Bader-El-Den, M., Tamma, V., Prendergast, D., Osborn, P.: Hybrid system for prostate MR image segmentation using expert knowledge and machine learning. In: Bramer, M., Stahl, F. (eds.) *SGAI 2023*. LNCS, vol. 14381, pp. 493–498. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-47994-6_43
11. Anwar, S.M., Majid, M., Qayyum, A., Awais, M., Alnowami, M., Khan, M.K.: Medical image analysis using convolutional neural networks: a review. *J. Med. Syst.* **42**(11), 1–13 (2018)
12. Chahal, E.S., Patel, A., Gupta, A., Purwar, A.: Unet based Xception model for prostate cancer segmentation from MRI images. *Multimed. Tools Appl.* **81**(26), 37333–37349 (2022)
13. Siddique, N., Paheding, S., Elkin, C.P., Devabhaktuni, V.: U-net and its variants for medical image segmentation: a review of theory and applications. *IEEE Access* **9**, 82031–82057 (2021)
14. Aldoj, N., Biavati, F., Michallek, F., Stober, S., Dewey, M.: Automatic prostate and prostate zones segmentation of magnetic resonance images using DenseNet-like U-net. *Sci. Rep.* **10**(1), 1–17 (2020)
15. Ma, J.J., et al.: Diagnostic image quality assessment and classification in medical imaging: opportunities and challenges. In *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*, pp. 337–340. IEEE (2020)
16. Litjens, G., Debats, O., Barentsz, J., Karssemeijer, N., Huisman, H.: Computer-aided detection of prostate cancer in MRI. *IEEE Trans. Med. Imaging* **33**(5), 1083–1092 (2014)
17. Masoudi, S., et al.: Quick guide on radiology image pre-processing for deep learning applications in prostate cancer research. *J. Med. Imaging* **8**(1), 010901 (2021)
18. Buda, M., Maki, A., Mazurowski, M.A.: A systematic study of the class imbalance problem in convolutional neural networks. *Neural Netw.* **106**, 249–259 (2018)
19. He, H., Garcia, E.A.: Learning from imbalanced data. *IEEE Trans. Knowl. Data Eng.* **21**(9), 1263–1284 (2009)
20. Yap, B.W., Rani, K.A., Rahman, H.A.A., Fong, S., Khairudin, Z., Abdullah, N.N.: An application of oversampling, undersampling, bagging and boosting in handling imbalanced datasets. In: Herawan, T., Deris, M.M., Abawajy, J. (eds.) *Proceedings of the First International Conference on Advanced Data and Information Engineering (DaEng-2013)*. LNEE, vol. 285, pp. 13–22. Springer, Singapore (2014). https://doi.org/10.1007/978-981-4585-18-7_2
21. Haixiang, G., Yijing, L., Shang, J., Mingyun, G., Yuanyue, H., Bing, G.: Learning from class-imbalanced data: review of methods and applications. *Expert Syst. Appl.* **73**, 220–239 (2017)
22. Wang, K.J., Makond, B., Chen, K.H., Wang, K.M.: A hybrid classifier combining SMOTE with PSO to estimate 5-year survivability of breast cancer patients. *Appl. Soft Comput.* **20**, 15–24 (2014)
23. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **16**, 321–357 (2002)
24. Bader-El-Den, M., Teitei, E., Perry, T.: Biased random forest for dealing with the class imbalance problem. *IEEE Trans. Neural Netw. Learn. Syst.* **30**(7), 2163–2172 (2018)

25. Safdar, K., Akbar, S., Shoukat, A.: A majority voting based ensemble approach of deep learning classifiers for automated melanoma detection. In: 2021 International Conference on Innovative Computing (ICIC), pp. 1–6. IEEE (2021)
26. Ju, C., Bibaut, A., van der Laan, M.: The relative performance of ensemble methods with deep convolutional neural networks for image classification. *J. Appl. Stat.* **45**(15), 2800–2818 (2018)
27. Litjens, G., et al.: A survey on deep learning in medical image analysis. *Med. Image Anal.* **42**, 60–88 (2017)
28. The Cancer Imaging Archive (TCIA). (2022). Prostate-MRI-US-Biopsy [Data file]. Retrieved from <https://wiki.cancerimagingarchive.net/pages/viewpage.action?pageId=68550661>
29. Bardis, M., et al.: Segmentation of the prostate transition zone and peripheral zone on MR images with deep learning. *Radiol. Imaging Cancer*, **3**(3) (2021)
30. Bonekamp, D., Jacobs, M.A., El-Khouli, R., Stoianovici, D., Macura, K.J.: Advancements in MR imaging of the prostate: from diagnosis to interventions. *RadioGraphics* **31**, 677–703 (2011)
31. Wang, Z., Wu, R., Xu, Y., Liu, Y., Chai, R., Ma, H.: A two-stage CNN method for MRI image segmentation of prostate with lesion. *Biomed. Signal Process. Control* **82**, 104610 (2023)
32. Hassanzadeh, T., Hamey, L.G., Ho-Shon, K.: Convolutional neural networks for prostate magnetic resonance image segmentation. *IEEE Access* **7**, 36748–36760 (2019)
33. Negi, A., Raj, A.N.J., Nersisson, R., Zhuang, Z., Murugappan, M.: RDA-UNET-WGAN: an accurate breast ultrasound lesion segmentation using wasserstein generative adversarial networks. *Arab. J. Sci. Eng.* **45**, 6399–6410 (2020)
34. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28
35. Isensee, F., Jaeger, P.F., Kohl, S.A., Petersen, J., Maier-Hein, K.H.: nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation. *Nat. Methods* **18**(2), 203–211 (2021)
36. Tian, Z., Liu, L., Fei, B.: Deep convolutional neural network for prostate MR segmentation. *Int. J. Comput. Assist. Radiol. Surg.* **13**(11), 1687 (2018)

Machine Vision



Optimizing Autonomous Vehicle Racing Using Reinforcement Learning with Pre-trained Embeddings for Dimensionality Reduction

Martin Holen¹(✉) , Jayant Singh² , Christian W. Omlin¹ , Jing Zhou²,
Kristian M. Knausgård² , and Morten Goodwin¹ 

¹ Centre for Artificial Intelligence Research, University of Agder,
4879 Grimstad, Norway
martin.holen@uia.no

² Top Research Centre Mechatronics, University of Agder, 4879 Grimstad, Norway
<https://cair.uia.no/>

Abstract. In the rapidly evolving domain of reinforcement learning (RL), which has applications in computer vision and games, our research presents an RL-based Embedding algorithm (EmbRL) that, applied to an autonomous car racing environment, allows for rapid algorithm training with significant results.

EmbRL addresses the challenge of processing high-dimensional camera inputs, which is common in advanced game environments like OpenAI Five and AlphaStar. By employing a pre-trained supervised learning model, our algorithm efficiently transforms these inputs into a set of 1000 class features, which are then processed by a fully connected network (FCN) acting as the RL model.

This method effectively separates the task of understanding the vehicle's state from the core path-finding and control tasks performed using a separate RL network, simplifying the task of autonomous car racing. Our findings show a remarkable reduction in training time, speeding up training by 230% compared to traditional end-to-end convolutional networks, as well as a significant boost to the reward, highlighting EmbRL's potential in enhancing the real-time applicability of vision models. This study integrates concepts from established methodologies, incorporating minor modifications that result in significant enhancements in performance.

1 Introduction

Reinforcement learning (RL) is one of the main machine learning paradigms, where agents learn by interacting with an environment and maximizing a cumulative reward [28]. RL challenges include sparse rewards, high-dimensional states, action spaces, and increasingly large compute requirements due to these highly complex RL environments [1, 31]. Various methods for enhancing learning efficiency and performance, such as transfer learning, dimensionality reduction, and knowledge distillation, have emerged as promising techniques for RL [2, 3, 14].

Proximal policy optimization (PPO) [23] is an algorithm that aims to improve training for highly complex RL environments. PPO uses experiences containing (action a_t , reward r_t and state s_t) and updates with minibatches, which has some of the benefits of trust region policy optimization while being more general and easy to implement. It does this by calculating the probability ratio $r_t(\theta)$ (shown in Eq. 1) using the stochastic policy $\pi_\theta(a_t|s_t)$ as well as the old policy $\pi_{\theta_{old}}(a_t|s_t)$. It then clips the probability ratio ($r_t(\theta)$) between $1 - \epsilon$ and $1 + \epsilon$ and chooses the minimum between the clipped and unclipped ratio (refer to Eq. 2), where \hat{A}_t refers to the advantage function at timestep.

$$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)} \quad (1)$$

$$L^{CLIP}(\theta) = \hat{E}_t[\min(r_t \hat{A}_t, \text{clip}(r_t, 1 - \epsilon, 1 + \epsilon) \hat{A}_t)] \quad (2)$$

Traditional RL requires an agent to learn from scratch for each environment or task, requiring extensive resources and time for training the algorithm. Transfer learning (TL) offers a solution by allowing knowledge transfer from one task to a similar task [33], such as an agent learning how to navigate a maze being moved to a task of navigating in a complex environment [32]. TL can also be done between the paradigms; a method uses unsupervised learning (UL) to RL; another method is behavior transfer [3].

RL often has high-dimensional state spaces, such as camera inputs through robotics tasks or game environments. Training in these spaces requires vast amounts of computational resources, with the possibility of the curse of dimensionality. Dimensionality reduction techniques aim to reduce the high-dimensional states into a lower-dimensional space while containing all the information from the higher-dimensional state [6]. Decreasing the dimensions increases the computational efficiency [2].

As Deep-RL models keep growing, their deployment in resource-constrained settings becomes challenging. Knowledge distillation addresses this by training a larger model, commonly called the teacher, and training a smaller model or a student to predict the same values for the same input. When the student model is done, its performance should be similar to the larger model while being significantly faster to compute [14].

In the realm of autonomous vehicles, tasks are time-sensitive, demanding real-time algorithms for safety-critical functions [13, 15, 27]. These vehicles' operations can be categorized into five tasks [16]:

- **Sensing:** Involves sensors for environmental perception.
- **Perceiving and Localizing:** Encompasses supervised learning (SL) objectives such as object detection and SLAM.
- **Scene Representation:** Integrates sensor fusion, behavior prediction, and object mapping.
- **Planning and Deciding:** Entails path planning, trajectory optimization, and driving policy formulation.
- **Control:** Dictates velocity, steering, acceleration, and braking.

Of these five tasks, the first three tasks, sensing, perceiving and localizing, and scene representation, are referred to as scene understanding in the context of this work. The last three tasks of Scene representation, Planning & deciding and Control are referred to as decision-making and planning in accordance with Kiran et al. [16].

Natural language processing (NLP) based autonomous vehicle tasks aim to increase the explainability of the autonomous vehicles [4, 7, 9]. NLP commonly use embeddings, which has been used to calculate the similarity of different words, embedding information from text as well as other tasks [4, 5, 20]. Among these Chen, Sinavski et al. [4] embeds information from the environment, then use the NLP model to give the actions to control the vehicle. The embedding is done by taking information such as the route, velocity of other vehicles, potential pedestrians, and velocity of the ego vehicle, sending it through as a vector encoder, and merging the resulting embedding with a prompt embedding [4].

Our work introduces a system, shown in Fig. 1a, that integrates the above concepts. We employ a pre-trained supervised learning (SL) network for dimensionality reduction, creating embeddings from camera inputs. These embeddings represent the SL model’s interpretation of the image, serving as an input for the compact fully connected network (FCN).

According to [33], Transfer learning is a machine learning framework where data and algorithms from one task may be leveraged in a new related one. By this definition, a task involving RL to train an autonomous racing car using continuous actions would likely not be considered a “related task” to a supervised learning classification task that predicts discrete classes and, therefore, not considered TL. However, our approach integrates several commonly used methods of TL with minor modifications. These modifications lead to a significant performance improvement, making the system’s application to autonomous car racing more feasible.

By only updating a smaller network using PPO and leveraging the frozen pre-trained SL model for scene representation, our approach simplifies the RL task to planning and vehicle control. This paper delves into our methodology, experimental setup, and the promising results obtained.

2 Background

In the sphere of state embeddings in reinforcement learning (RL), the primary research avenues are divided into two distinct themes: world models and state aggregation via bisimulation metrics [21].

World models employ supervised learning (SL) to develop an environment model using sampled experiences. This approach either leads to compressed state representations [11] or enables agent training directly with the model [10, 22]. Notably, world models have demonstrated remarkable improvements in sample efficiency, especially in complex environments like Atari 2000 domains [12].

An additional facet of world models is their capability to enhance training exploration through these compressed state representations [8, 29]. This enhance-

ment is often achieved by evaluating transition prediction errors or examining the distances between state embeddings.

In a similar context, Munk et al. [19] have explored the utilization of environment models for offering state representations to RL agents, further extending the application scope of world models.

On the other hand, bisimulation focuses on aggregating states with similar behaviors, a strategy that can accelerate convergence by grouping akin states into abstract categories [18]. This process can be implemented using various metrics, with recent advancements introducing scalable deep learning methodologies to this end.

Another approach in this field is vision-based state estimation [30]. This method employs cameras to ascertain the state values of objects, integrating visual data directly into the state estimation process. Shen et al. [26] did this by creating modules which estimated the state of the vehicle using three different sensors, a monocular camera, a stereo camera setup and an IMU. These were used to estimate where the vehicle was, in its surroundings, what its pose was and fixing it when there was an error compared to where it was supposed to be. For testing their setup they used a rotorcraft which was set to hover, and they then tested the estimated state and compared it to a “sub-millimeter accurate Vicon motion tracking system”, they also tested it for a more complex outdoor environment and saw some deviation which could be caused by the surroundings changing because of wind and other factors [26].

More contemporary research ventures into leveraging pre-trained networks for embedding visual data, working to decrease the state representation while still retaining the important information. Pritz et al. [21] uses the state and action to estimate the next state similarly to world models, they then use this trained model to train the embedding model and action embedding model, which are both trained using supervised learning. Shah et al. [25] removes the last layer of the network that predicts the ImageNet classes, gathers data from human demonstrators and uses behaviour cloning to initialize the RL network, after which they train the model until it reaches their metric requirement. Another new model integrates multiple modules from Natural Language Processing (NLP) and vision domains [24], allowing them to specify what the goal is via text. This allows a user to type where to go, which an NLP algorithm preprocesses, and sends it to a Language to vision model that creates a path. The path is updated based on new experiences, this is then sent to a vision module which decides the action to take.

Our methodology distinguishes itself from world models as it does not rely on constructing a separate world model for state representation. In comparison to bisimulation, our approach shares some similarities but diverges in its lack of state aggregation. Instead, it develops an embedding in a continuous space. Our method differs from others in terms of vision state estimation as it does not explicitly estimate the state of objects, though information such as vehicle positioning and road shape is likely included in the embedding. The closest resemblance to our work lies in the embedding of visual information. However,

unlike other studies, our method does not necessitate extensive alterations to pre-trained models, adding extra information from other sensors, or training the model to handle state embedding. Temporal aspects are often managed by variants of recurrent neural networks (RNNs) in other models, making our approach notably simpler and more straightforward to implement, requiring minimal code adjustments. The application of NLP and vision modules in other research diverges from our method, as these typically embed their respective data before integration into the final network layers.

3 Methods

3.1 EmbRL

EmbRL operates on inputs compatible with a pre-trained convolutional network, predicting the 1000 ImageNet classes. This prediction serves as input for a compact FCN (see Fig. 1a). The pre-trained model’s predictions remain consistent for identical inputs. Once processed, the model estimates the likelihood of each of the 1000 classes being present in the input. This prediction is effectively an embedding representing the scene understanding for the camera inputs. A subsequent compact FCN predicts actions based on the pre-trained model’s output (refer to Algorithm: 1). Owing to its reliance on embeddings rather than extensive camera inputs, EmbRL is faster to train than conventional convolutional models, as backpropagation is limited to the smaller network (refer to Fig. 1b).

Algorithm 1. EmbRL

```

1: weights = parameter_server.get_weights()
2: for  $epoch = 1, 2, \dots, N_{epochs}$  do
3:   for  $agent = 1, 2, \dots, N$  do
4:     agent.set_weights(weights)
5:     while  $step < max\_steps$  and not done do
6:       embedding = pre-trained(state)
7:       action = PPO(embedding)
8:       next_state, reward, done = env.step(action)
9:       memory.save(embedding, reward, done, action)
10:      gradients = PPO.calc_gradients(memory)
11:    end while
12:  end for
13:  new_weights = parameter_server.
14:    sum_gradients(gradients)
15: end for

```

The algorithm starts by obtaining the weights from a parameter server using the `parameter_server.get_weights()` function. The algorithm enters a training loop that runs for a specified number of epochs, denoted as ‘ N_{epochs} ’. Within each epoch, there is another loop that iterates over individual agents, indexed

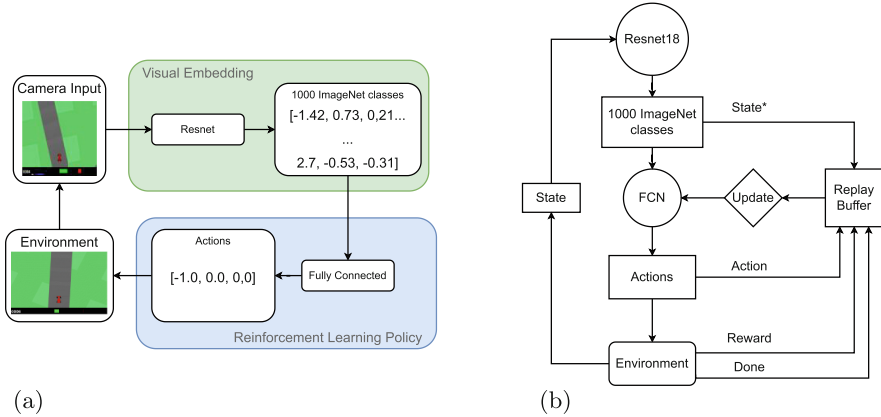


Fig. 1. (a) Shows how the EmbRL System runs, with the environment giving the camera input, which is sent to the pre-trained network for embedding. The embedding is then used as the input for the RL algorithm, with which it predicts the action to perform in the environment. (b) Illustrates the EmbRL update process: Images are input to the visual embedded. The resulting embedding is fed to the FCN to predict actions and is stored in memory. The predicted action is executed in the environment and stored in the memory, yielding a reward, episode status, and a new camera view. Both the reward and episode status are also stored in memory. After two episodes, the FCN undergoes an update based on the accumulated data.

from 1 to ‘ N ’. For each agent in the inner loop, the agent’s neural network weights are set to the same weights obtained from the parameter server using `agent.set_weights(weights)`. Within each agent’s context, there is a while loop that continues until a maximum number of steps (`max_steps`) is reached or a termination condition (`done`) is met. This loop represents an episode of interaction between the agent and its environment. In each step of the episode, the current state is passed through a pre-trained model (referred to as `pre-trained(state)`), which generates an embedding for the state. This embedding is a lower-dimensional representation of the state space. The agent selects an action using the PPO algorithm based on the state embedding obtained in the previous step. This action represents the agent’s policy decision. The selected action is applied to the environment (`env.step(action)`), leading to transitions to the next state, a reward signal, and a flag indicating whether the episode is done (`done`). The state embedding, reward, termination flag, and action taken in each step are stored in a memory buffer using `memory.save(embedding, reward, done, action)`. Gradients for updating the agent’s policy are calculated using the stored experiences in the memory buffer. This involves calling `PPO.calc_gradients(memory)` to compute the policy gradients. After all agents have completed their episodes, the calculated gradients are summed across agents using `parameter_server.sum_gradients(gradients)`, resulting in updated weights

(new_weights). The outer loop advances to the next epoch, and the process is repeated for the specified number of epochs.

The EmbRL algorithm combines elements of distributed reinforcement learning, where multiple agents interact with their environments, with policy optimization using PPO. It leverages pre-trained state embeddings to represent states and aims to train agents to perform tasks in a reinforcement learning setting. The specific details of how these components are implemented and interact with each other would depend on the actual code implementation.

The reasoning behind our system, is to offload part of the work commonly done by a RL-agent, to a pre-trained network using ImageNet weights. Looking at the three tasks of Scene Understanding in autonomous driving from Kiran et.al., we give the task of Scene Understanding to the pre-trained network, leaving only the two last tasks of path planning and control [16]. This leads to a significantly simpler environment for the RL algorithm to learn from, as instead of having to understand what those shapes are in the picture and then needing to learn how to drive, it can now simply aim to learn the controls given the embedding. It still needs to learn what the embeddings represent, but we show that the embeddings are representable of the environment later with our results. In essence, our system reduces the dimensionality of the input, offloading one of the tasks, and leaving less work for the RL algorithm.

3.2 System Overview

Our system adopts a parameter server setup, utilizing eight agents for experience collection. Each agent gathers experience from two episodes in the Gym CarRacing-v2 environment. Post-experience collection, agents compute their gradients using the PPO update algorithm based on their experiences. These gradients are relayed to the parameter server for further processing. The server averages the gradients, updates the model, and dispatches the updated model to agents for subsequent data collection.

Hardware and Architecture

- **Hardware** Training of the models was conducted on an Nvidia DGX-2 equipped with 1.5 TB RAM, 48 CPU Cores, and 16 Nvidia Tesla V100 GPUs; Max latency was done on a AMD Ryzen threadripper 3960 × 24-core processor CPU with an Nvidia 3090 GPU.
- **Neural Network** We adopted torchvision’s models across all three implementations and incorporated a three-layer FCN using torch for the EmbRL approach.

3.3 Implementation

Three distinct implementations were explored:

- A ResNet18 model using RL to train from scratch without pre-trained weights (referred to as ‘ResNet18’).

- A ResNet18 model leveraging pre-trained weights for transfer learning with RL into the Gym environment (termed ‘ResNet18_transfer’).
- Our own EmbRL employs a pre-trained input embedding model, followed by training on a three-layer FCN; the three-layer FCN is then trained in the RL environment while the pre-trained network is not updated.

Environment. CarRacing-v2 is an environment part of gyms box2d created by Oleg Klimov. CarRacing is a top down, car simulator with realistic physics, where the vehicle is rear wheel drive [17]. The environment randomly generates the track, and allows for training with domain randomization. Actions controlling the vehicle includes, steering, acceleration and breaking, with the state being a $96 \times 96 \times 3$ camera input. The reward system, includes a reward ($1000/N$, where N is the number of tiles) and a small penalty for each action performed of -0.1 . As for ending the episode, this happens if the vehicle drives off the map or finishes all tiles; though it is common to have a maximum amount of actions to perform.

Distributed Training. To speed up training, distributed learning was used with a worker and parameter server setup. The distributed learning was done using the Python library ray, with eight worker agents and one parameter server. The workers run two episodes each, gathering experiences, after which they use said experiences to calculate the gradients and send them with their respective metric (loss/reward) to the parameter server. The parameter server takes the gradients of the workers and sums them, after which it updates the weights and sends the updated weights to each of the workers. The workers and parameter server repeat the updates n number of times, after which they gather the next set of experiences. The code can be found at¹.

4 Results and Discussion

In this section we will look into the results of our system comparing it with different pre-trained network and two ResNet implementation, one which is pre-trained on ImageNet and another which is learning from randomly initialized weights. The results of the ResNet implementations are averaged for ten runs, the same goes for the EmbRL algorithm with ResNet18 as its pre-trained model; the EmbRL pre-trained models are averaged over four runs.

Our experiments reveal that the EmbRL approach demonstrates superior learning speed and efficiency compared to traditional ResNet implementations as shown in Fig. 2a.

Learning Speed and Epoch Performance: As depicted in Fig. 2a, EmbRL rapidly converges to solve the environment, while the ResNet models lag in their learning trajectories. As for the runtime, the EmbRL method using a ResNet18 model finishes in 206.56 h or 230.23% faster than an end-to-end ResNet18, as

¹ <https://github.com/marho13/EmbeddingInput>.

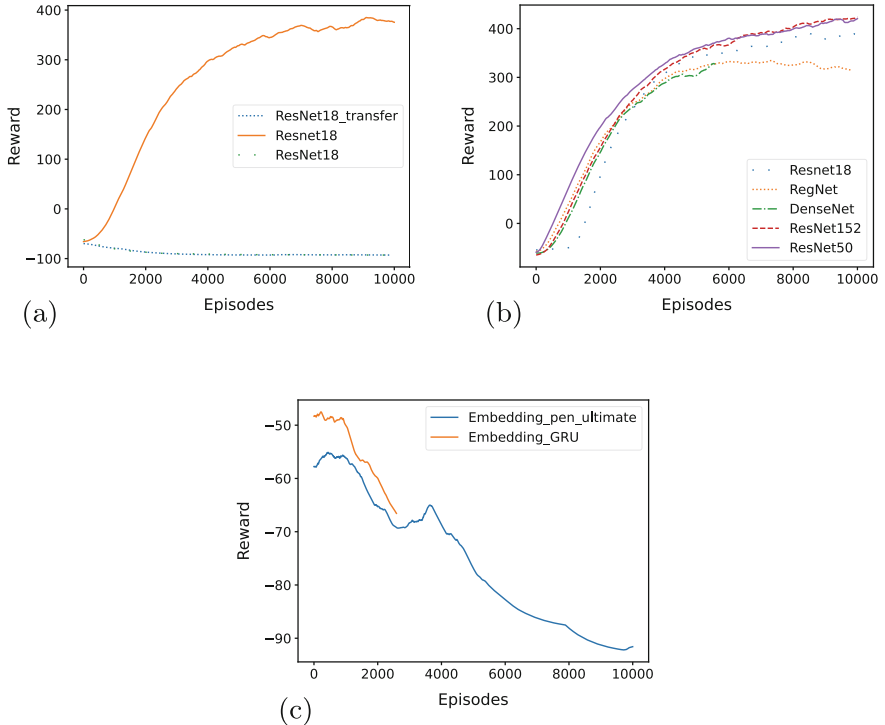


Fig. 2. (a) Shows the performance of two methods which use end-to-end RL where one uses ImageNet weights and transfer learns on them (ResNet18_transfer), the next uses randomly initialized weights (ResNet18) and lastly comparing it to our method EmbRL. The x-axis being the number of epochs, and is y-axis the reward. (b) Shows the performance of EmbRL with different pre-trained networks (ResNet18/50/152, RegNet-32GF and Densenet201), with the x-axis being the number of epochs, and the y-axis being the reward. (c) Shows the results of using the penultimate layer and a GRU model

shown in Tables: 3. Interestingly, when substituting ResNet18 with another pre-trained model, performance remains consistent (refer to Fig. 2b), though there are some algorithms which perform slightly better than ResNet18, such as Resnet50 and ResNet152. This suggests that optimizing the choice of pre-trained network could yield further improvements.

In our comparison of EmbRL methods versus traditional ResNet18 methods, we conducted 10 runs to minimize randomness. However, when evaluating different EmbRL pre-trained models, we performed only 4 runs. This discrepancy arises because comparing traditional methods necessitates a larger number of runs to ensure robust performance evaluation of EmbRL. In contrast, when comparing various pre-trained models, the primary objective is to assess whether scaling up the network yields significant performance improvements.

Our tests indicate that while there may be a slight improvement, there is little benefit in using a considerably larger model compared to ResNet50. ResNet50, which has a maximum latency of 22.84 ms as detailed in Table 2, demonstrates the best performance overall, as illustrated in Fig. 2b. Given that ResNet50 not only exhibits the highest performance but also has the lowest variance among the top-performing algorithms, further testing is unlikely to vastly change its performance compared to the other algorithms.

When testing the TL method of using the pen ultimate layer, we found that the model was unable to learn how to control the vehicle. Our GRU implementations resulted in similar results (refer to Fig. 2c). Note that though the GRU implementation did not finish all 10 000 episodes for each agent, it did run for significantly longer than the pen ultimate implementation.

Our metrics of comparison include variance (refer to Table: 1, this is the variance in reward for the FCN where we find that using a ResNet18 pre-trained network has the highest variance; This could be due to more runs of ResNet18, though there is still large variance when comparing it to ResNet50 and ResNet152. As for DenseNet and RegNet they have a lower variance, but they still have a lower reward.

Table 1. Name of model tested, and their respective summed variance. The ResNet18 refers to the randomly initialized ResNet18 model, and the ResNet18_transfer is the pre-trained ResNet18 TL into the RL task.

Model name	Variance
EmbRL_ResNet18	96.85
EmbRL_ResNet50	56.14
EmbRL_ResNet152	73.32
EmbRL_DenseNet	41.23
EmbRL_RegNet	47.90
EmbRL_Pen_Ultimate	5.99
EmbRL_GRU	0.64
ResNet18	3.26
ResNet18_transfer	1.35

4.1 Latency and Model Size

A critical observation pertains to latency. As indicated in Tables: 2, the maximum time interval between two consecutive actions varies significantly across different pre-trained models. The smallest network ResNet18, has a maximum latency of approximately 1/8th of DenseNet201; this means that ResNet18 can perform about eight actions for every action that DenseNet201 can perform, leading to a faster reaction time. In real-world scenarios, such as autonomous driving, this latency could be the difference between avoiding an obstacle and a potential collision. Consequently, the choice of model might be influenced by the computational capabilities of the vehicle’s hardware. More powerful systems could accommodate larger models for both pre-trained networks and the

Table 2. Name of pre-trained networks used with EmbRL, their performance on ImageNet, their parameter count, and maximum latency in milliseconds(ms) between two actions for 100 epochs. Note that the maximum latency is excluding the update time.

Name of network	Top1 Performance	Top5Performance	Parameter Count	Max Latency
ResNet18	69.758	89.078	11.7M	17.49 ms
ResNet50	80.858	95.434	25.6M	22.84 ms
ResNet152	82.284	96.002	60.2M	32.77 ms
DenseNet	76.896	93.37	20.0M	134.28 ms
RegNet	86.838	98.362	145M	111.65 ms

Table 3. Max runtime during training, in hours, for EmbRL and the end-to-end Resnet18

Algorithm	Runtime
EmbRL_ResNet18	206.56 h
Resnet18	475.56 h

RL model. Looking at the model size and their effects, we see that our system utilizes significantly less memory during training, as it only stores the embeddings during training instead of images. When training with ResNet18, EmbRL uses 1777 MB on the GPU with the stored states being of size [x, 1000]. The base ResNet18 implementation uses 4200 MB GPU memory during training, with a state shape of [x, 96, 96, 3]. As for how much total memory is used, each image includes 27,648 numbers, compared to the state which only has 1000 floating points. This means that storing the camera states uses approximately 27.64 times more memory during training compared to storing the embeddings. As such, the algorithm does not only converge faster compared to a baseline ResNet model, but it also trains faster and maintains a low max latency during inference.

4.2 Future Work

Future work includes testing this for more RL-based autonomous vehicle environments. Attempting to estimate the state values of the vehicle using subsequent camera input, feeding it to EmbRL. Testing pre-trained network embeddings from tasks other than image classification, including transformers, image captioners, and different Neural Networks for the RL part of the system.

5 Conclusion

This paper introduces EmbRL, which applies an embedding system that uses transfer learning-like methods on two different tasks with some changes.

In a typical TL approach, the last layer in the NN is modified and updated on the new data. This paper, instead, uses the full network and adds a separate network at the end for the RL task. This approach leads to vast improvements compared to standard TL methods. In this way, the system facilitates efficient training of a compact NN, addressing the challenges of computationally demanding environments while conserving resources.

Our experimental results underscore the efficacy of EmbRL, as evidenced by its rapid convergence within 160,000 episodes in the CarRacing-v2 gym environment. This performance significantly surpasses that of traditional methods, including ResNet trained with RL from randomly initialized parameters and the pre-trained ResNet model utilizing direct TL. Notably, our method achieves 230.23% faster training when using ResNet18 as its pre-trained network, compared to a standard ResNet18, while also delivering superior performance in the car racing game environment CarRacing-v2 from gym.

The observed performance of EmbRL is consistent not only with the ResNet18 pre-trained classifier network but also with the majority of other tested pre-trained classifier networks. However, latency considerations may narrow the range of classifiers, especially in time-sensitive environments.

As future work, there is considerable potential to enhance EmbRL through the exploration of alternative network architectures of varying sizes and by implementing algorithmic modifications. Our objective is to achieve consistent performance across a range of image fidelities and multiple environments including real-world scenarios.

References

1. Arulkumaran, K., Cully, A., Togelius, J.: Alphastar: an evolutionary computation perspective. In: GECCO 2019 Companion - Proceedings of the 2019 Genetic and Evolutionary Computation Conference Companion, pp. 314–315 (2019). <https://doi.org/10.1145/3319619.3321894>
2. Becker, M., Lippel, J., Stuhlsatz, A.: Regularized nonlinear discriminant analysis - an approach to robust dimensionality reduction for data visualization. In: VISIGRAPP 2017 - Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, vol. 4, pp. 116–127 (2017). <https://doi.org/10.5220/0006167501160127>
3. Campos, V., et al.: Beyond fine-tuning: transferring behavior in reinforcement learning
4. Chen, L., et al.: Driving with LLMs: fusing object-level vector modality for explainable autonomous driving. <https://github.com/wayveai/Driving-with-LLMs>
5. Chiu, B., Crichton, G., Korhonen, A., Pyysalo, S.: How to train good word embeddings for biomedical NLP, pp. 166–174 (2016). <http://www.ncbi.nlm.nih.gov/pmc/>
6. Dai, B., Shen, X., Wang, J.: Embedding learning. *J. Am. Stat. Assoc.* **2022**(537), 307–319 (2020). <https://doi.org/10.1080/01621459.2020.1775614>
7. Dong, J., Chen, S., Miralinaghi, M., Chen, T., Li, P., Labi, S.: Why did the AI make that decision? Towards an explainable artificial intelligence (XAI) for autonomous driving systems. *Transp. Res. Part C: Emerg. Technol.* **156**, 104358 (2023). <https://doi.org/10.1016/J.TRC.2023.104358>

8. Ermolov, A., Sebe, N.: Latent world models for intrinsically motivated exploration. In: *Advances in Neural Information Processing Systems*, vol. 34 (2020). <https://github.com/htdt/lwm>
9. Gao, F., Ping, Q., Thattai, G., Reganti, A., Wu, Y.N., Natarajan, P.: Transform-retrieve-generate: natural language-centric outside-knowledge visual question answering. In: *Computer Vision Pattern Recognition* (2022). <https://github.com/JaidedAI/EasyOCR>
10. Ha, D., Urgan Schmidhuber, J.: World Models. <https://worldmodels.github.io>
11. Ha Google Brain Tokyo, D., Schmidhuber, J.: Recurrent world models facilitate policy evolution. In: *Advances in Neural Information Processing Systems*, vol. 31 (2018). <https://worldmodels.github.io>
12. Hafner, D., Research, G., Lillicrap, T., Norouzi, M., Ba, J.: Mastering atari with discrete world models
13. Hanzinger, T.A., Sifakis, J.: The embedded systems design challenge. In: Misra, J., Nipkow, T., Sekerinski, E. (eds.) *FM 2006. LNCS*, vol. 4085, pp. 1–15. Springer, Heidelberg (2006). https://doi.org/10.1007/11813040_1
14. Hinton, G., Dean, J.: Distilling the knowledge in a neural network (2015)
15. Kato, S., et al.: Autoware on board: enabling autonomous vehicles with embedded systems. In: *Proceedings - 9th ACM/IEEE International Conference on Cyber-Physical Systems, ICCPS 2018*, pp. 287–296 (2018). <https://doi.org/10.1109/ICCPS.2018.00035>
16. Kiran, B.R., et al.: Deep reinforcement learning for autonomous driving: a survey. *IEEE Trans. Intell. Transp. Syst.* **23**(6), 4909–4926 (2022). <https://doi.org/10.1109/TITS.2021.3054625>
17. Klimov, O.: Car Racing - Gym Documentation. https://www.gymlibrary.dev/environments/box2d/car_racing/
18. Li, L.: Towards a Unified Theory of State Abstraction for MDPs (2006). <http://anytime.cs.umass.edu/aimath06/proceedings/P21.pdf>
19. Munk, J., Kober, J., Babuska, R.: Learning state representation for deep actor-critic control. In: *2016 IEEE 55th Conference on Decision and Control, CDC 2016*, pp. 4667–4673 (2016). <https://doi.org/10.1109/CDC.2016.7798980>
20. Patil, R., Boit, S., Gudivada, V., Nandigam, J.: A survey of text representation and embedding techniques in NLP. *IEEE Access* **11**, 36120–36146 (2023). <https://doi.org/10.1109/ACCESS.2023.3266377>
21. Pritz, P.J., Ma, L., Leung, K.K.: Jointly-learned state-action embedding for efficient reinforcement learning. In: *International Conference on Information and Knowledge Management, Proceedings*, pp. 1447–1456 (2021). <https://doi.org/10.1145/3459637.3482357>
22. Schmidhuber, J.: On Learning to Think: Algorithmic Information Theory for Novel Combinations of Reinforcement Learning Controllers and Recurrent Neural World Models Technical report (2015)
23. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Openai, O.K.: Proximal Policy Optimization Algorithms (2017). <https://arxiv.org/abs/1707.06347v2>
24. Shah, D., Osiński, B., Ichtter, b., Levine, S.: LM-Nav: robotic navigation with large pre-trained models of language, vision, and action. In: Liu, K., Kulic, D., Ichnowski, J. (eds.) *Proceedings of The 6th Conference on Robot Learning. Proceedings of Machine Learning Research*, vol. 205, pp. 492–504. PMLR (2023). <https://proceedings.mlr.press/v205/shah23b.html>
25. Shah, R., Kumar, V.: RRL: ResNet as representation for reinforcement learning. In: *Proceedings of the 38th International Conference on Machine Learning*, vol. 38 (2021)

26. Shen, S., Mulgaonkar, Y., Michael, N., Kumar, V.: Vision-based state estimation for autonomous rotorcraft MAVs in complex environments. In: Proceedings - IEEE International Conference on Robotics and Automation, pp. 1758–1764 (2013). <https://doi.org/10.1109/ICRA.2013.6630808>
27. Stankovic, J.A.: Real-time and embedded systems. *ACM Comput. Surv.* **28**(1) (1996)
28. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction, 2nd edn. The MIT Press, Cambridge (2015). <https://inst.eecs.berkeley.edu/~cs188/sp20/assets/files/SuttonBartoIPRLBook2ndEd.pdf>
29. Tao, R.Y., François-Lavet, V., Pineau, J.: Novelty search in representational space for sample efficient exploration. In: Advances in Neural Information Processing Systems, vol. 34 (2020). <https://github.com/taodav/nsrs>
30. Webb, T.P., Prazhenica, R.J., Kurdila, A.J., Lind, R.: Vision-based state estimation for autonomous micro air vehicles. **30**(3), 816–826 (2007). <https://doi.org/10.2514/1.22398>
31. Wurman, P.R., et al.: Outracing champion Gran Turismo drivers with deep reinforcement learning. *Nature* **602**(7896), 223–228 (2022). <https://doi.org/10.1038/s41586-021-04357-7>
32. Xu, Y., Hansen, N., Wang, Z., Chan, Y.C., Su, H., Tu, Z.: On the feasibility of cross-task transfer with model-based reinforcement learning. In: The Eleventh International Conference on Learning Representations (ICLR) (2023). <https://nicklashansen.github.io/xtra>
33. Zhuang, F., et al.: A comprehensive survey on transfer learning. *Proc. IEEE* **109**, 43–76 (2021). <https://doi.org/10.1109/JPROC.2020.3004555>



Semantic Segmentation for Landslide Detection Using Segformer

Hasnain Murtaza Syed^(✉), Mahdi Maktabdar Oghaz,
and Lakshmi Babu Saheer

Anglia Ruskin University, East Road, Cambridge CB1 1PT, UK
hs1008@student.aru.ac.uk, {mahdi.maktabdar,lakshmi.babu-saheer}@aru.ac.uk

Abstract. Landslides pose a significant threat to human life and infrastructure which urges the need for efficient techniques for identifying and categorising them. The advent of deep segmentation models such as the Segformer has shown a remarkable empirical performance for semantic segmentation tasks on well-known benchmark datasets, such as ADE20k and Cityscapes. Therefore, this research proposes utilising Segformer on the benchmark Chinese Academy of Sciences (CAS) Landslide Dataset, which features high-quality aerial images of areas impacted or prone to landslides. Taking advantage of the multi-scale attention mechanism and long-range dependency modeling characteristics of the Segformer architecture, this research aims to achieve state-of-the-art results for landslide segmentation using aerial images. Experimental results show the advantage of the Segformer model in segmenting landslide areas, with the largest Segformer variant achieving an Intersection over Union (IoU) score of 87.795% on the Unmanned aerial vehicle (UAV) dataset, surpassing the previous state-of-the-art model, Multiscale Feature Fusion and Enhancement Network (MFFENet), by 3.4%. On the Satellite (SAT) dataset, Segformer attained an IoU score of 79.300%, outperforming the previous best model, DeepLabv3+, by 11.163%. For the combined UAV&SAT dataset, Segformer achieved an IoU score of 85.157%, surpassing DeepLabv3+, the best previous model by 5.032%.

Keywords: segformer · landslide detection · semantic segmentation

1 Introduction

In recent years there has been a rise in the frequency of landslides causing significant dangers to human lives and infrastructure across the world [1]. Globally, landslides cause an estimated 4,600 deaths annually, highlighting the significant impact of this natural hazard on human lives [6]. Thus, accurate mapping of landslides is essential for emergency response and risk identification. Many studies attempted to use various image segmentation techniques to automate the

challenging task of landslide mapping using remote sensing and aerial imagery data. Although there has been much research on automated mapping of landslides using remote sensing images, the accuracy is hindered by the quality of the dataset and model performance. Notable studies in this field include: [7] who proposed a fully convolutional network within pyramid pooling (FCN-PP) method for landslide inventory mapping, [3] who developed a You Only Look Once (YOLO) model for detection from satellite images, and [1] who applied U-Net to Landsat 8 satellite images for landslide segmentation. Despite these advances, challenges remain in achieving high accuracy across diverse landslide scenarios. The advent of high-performance segmentation models like Segformer [12] and high-quality datasets such as the CAS [13] offer opportunities to further enhance the accuracy and efficiency of landslide detection and segmentation. Segformer has proven its great performance on similar case studies and datasets like ADE20K [14] and Cityscapes [4] indicating its potential for applications such as landslide detection [12]. This novel research applies the state-of-the-art Segformer architecture to landslide detection using the high-quality CAS landslide dataset. This work is the first to combine Segformer’s capabilities with such a comprehensive and well-annotated dataset. By evaluating Segformer across multi-sensor data, this study aims to improve the accuracy and efficiency of automated landslide mapping.

The rest of this paper is structured as follows: Sect. 2 provides an overview of related work in semantic segmentation for landslide detection. Section 3 describes the methodology. Section 4 presents the experimental results and discussion. Finally, Sect. 5 concludes the paper and outlines potential future research directions.

2 Related Work

Landslide detection using remote sensing imagery has evolved significantly over the past decade. The advent of deep learning techniques marks a significant leap in detection and segmentation accuracy.

Most automatic landslide detection techniques have recently relied on utilizing Convolutional neural networks (CNNs), which have proven highly effective at analyzing image data and extracting relevant features [8]. For instance, [7] conducted a series of experiments to validate their proposed FCN-PP (Fully Convolutional Network with Pyramid Pooling) method for landslide inventory mapping (LIM). [3] conducted experiments to validate their proposed small attentional YOLO model for landslide detection but they faced major issues as small models are efficient but often compromise on performance. [1] proposed using a U-Net with Landsat 8 satellite images but faced issues regarding data imbalance and the quality of available data. Despite these challenges, recent advancements in deep learning have introduced transformer-based models, which have shown good performance on other tasks [12] but there has been little research on their application in detecting landslides.

2.1 Transformer Based Models

Vision Transformers (ViT) [5] is the first transformer-based model used to experiment with image classification resulting in state-of-the-art performance. Since then, there has been research utilizing transformers for image segmentation tasks. SegFormer, a Transformer-based semantic segmentation model, was utilized for the identification of landslides by conducting extensive experiments to compare it with other models such as High-Resolution Network (HRNet), DeepLabv3, Attention-UNet, U2Net, and Fast Semantic Segmentation Network (FastSCNN) [11]. Chinese Academy of Sciences [13] developed the CAS Landslide Dataset, a large-scale and multisensory dataset specifically designed for deep learning-based landslide detection. They also compared the results of models such as FCN, U-Net, DeepLabv3+, and MFFENet.

2.2 Landslide Datasets

Access to various and well-documented landslide datasets is crucial for progress in landslide detection models. It is worth mentioning that this kind of multi-sensor, well annotated and large-scale dataset was missing before introducing the CAS (Chinese Academy of Sciences) landslide dataset by [13]. The dataset efficiently overcomes these challenges by providing annotations at a level, for various types of landslide situations, as illustrated in Fig. 1.

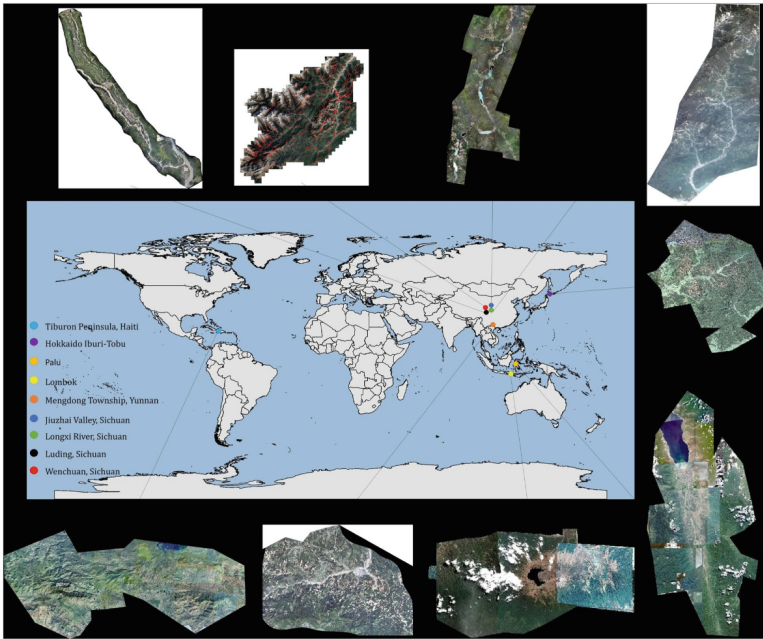


Fig. 1. Location map of the study areas.

2.3 Research Gaps and Opportunities

While segmentation techniques demonstrate a high detection rate for landslides, there remain significant opportunities for further exploration. One of the main challenges is the lack of high-quality and diverse datasets with comprehensive annotations, which hinders the effective evaluation of segmentation models. The complex and irregular characteristics of landslide features require computational models that can adeptly capture long-range dependencies and manage intricate spatial interactions. Transformer-based models like Segformer have shown potential by outperforming human experts in segmentation tasks, highlighting their capability in the landslide detection process. This paper addresses these research gaps by advancing landslide detection techniques through the use of the Segformer architecture and the CAS landslide dataset.

3 Methodology

This section outlines the approach taken in this study to explore how effective the Segformer architecture is, for detecting landslides using the CAS landslide dataset. Starting with the explanation of the dataset and its features then heading towards discussing the Segformer architecture and its key elements. Following that we delve into the specifics of the training and evaluation processes covering aspects such as division, data enhancement methods, and optimization configurations. We then present the setup, which includes three scenarios; UAV, SAT only, and UAV+SAT to evaluate how different data sources and their integration impact landslide detection performance. Finally, implementation specifics like the frameworks and libraries used, along with making our code available, for reliability and further study. The GitHub repository can be found here: www.github.com/syedddhasnainn/landslide-detection-segformer.

3.1 Dataset

In this research, the Segformer model was evaluated using the CAS [13] landslide datasets. These datasets include annotations, for high-resolution satellite (SAT) images and unmanned aerial vehicle (UAV) images to aid in landslide detection. The dataset covers a variety of landslide scenarios making it a valuable resource for developing and testing models, for detecting landslides. It contains 19,756 images from nine regions, most from various publicly available datasets and collaborative partners. The image and label were cropped into 512×512 TIFF format [13]. The dataset was divided into training, test, and validation sets with proportions of 64:20:16, as can be seen in Table 1, to assess the model's performance.

Table 1. Distribution of Images Across Datasets in Training, Validation, and Test Sets

Images Distribution			
	Train	Validation	Test
UAV	8603	2151	2689
SAT	4040	1010	1263
UAV + SAT	12643	3161	3952

3.2 Segformer Architecture

The Segformer, first invented by [12], was allegedly designed for this type of semantic segmentation task. It uses a hierarchical architecture with local and global attention which is specialized to extract both low-level details and long-ranged dependencies from the input photo. The architecture structure is an encoder-decoder, with the former taking MiT (the Mix Transformer) backbone and the latter employing a lightweight version of an MLP (Multi-Layer Perceptron) as the decoder, as illustrated in Fig. 2.

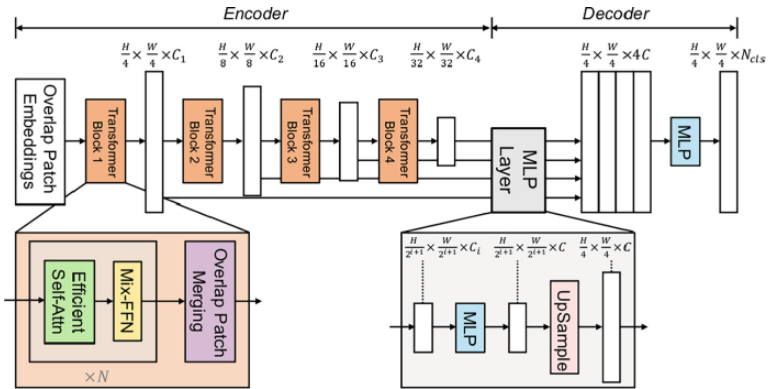


Fig. 2. A Segformer comprises two components; a Transformer encoder, for capturing both general and detailed features and a compact All MLP decoder, for integrating these various levels of features and forecasting the semantic segmentation mask

The SegFormer models were fine-tuned using the CAS landslide dataset. Different versions of Segformer models were analyzed, ranging from MIT-B0 to MIT-B5 to investigate how the model size and complexity were configured for this landslide detection research.

3.3 Training and Evaluation

The Segformer model was modified according to the CAS landslide dataset and UAV and SAT images were used for tuning the model. The training, test, and

validation datasets were split in a 64:20:16 ratio. Models were trained using the AdamW optimizer with a learning rate of 0.00006. The performance of the modified models was evaluated using a range of semantic segmentation metrics, including IoU, F1 score, recall, and accuracy.

3.4 Experimental Setup

Three sets of experiments were conducted to test the Segformer architecture for landslide detection. The models were trained and evaluated using UAV imagery, then SAT imagery, and finally both UAV and SAT combined. By comparing the results, this research aims to determine which approach provides the most accurate landslide detection.

3.5 Implementation Details

PyTorch [9] which is a popular deep learning framework, was used to carry out all the experiments for this work running on 4 NVIDIA L4 GPUs and it took around 24 h for the full training. The implementation of the Segformer models in the Huggingface library was utilized to establish a common framework for semantic segmentation. The source code including the training and test scripts and dataset preprocessing scripts will be made publicly available for open research and further development in this field.

4 Experimental Results and Discussions

This section demonstrates the experimental results using Segformer on the CAS landslide dataset and compares the performance with the latest and semantic segmentation models like FCN, U-Net, DeepLabV3+, and MFFENet [2, 8, 10, 15].

4.1 Comparison with State-of-the-Art Methods

The efficacy of the Segformer architecture is proven by comparing it with the leading semantic segmentation approaches. Table 2 compares the results of the Segformer model for UAV, SAT, and UAV+SAT scenarios with those of FCN, U-Net, DeepLabV3+, and MFFENet [12]. In the case where only UAV data is used the Segformer model has obtained an IoU score of 87.795% surpassing the leading model, MFFENet by 3.4%. This shows an enhancement in identifying landslide areas from high-resolution UAV images. The Segformer also excels in precision, recall, F1 score, mean IoU (mIoU), and overall accuracy (OA) compared to models assessed for UAV segmentation. For the SAT dataset, the Segformer model outperforms DeepLabv3+ by 11.163% in terms of IoU score with a score of 79.300% the Segformer model demonstrates its performance in accurately segmenting landslide areas from satellite imagery. In cases where both UAV and SAT datasets are combined the Segformer model achieves a score of 85.157% surpassing DeepLabv3+ by 5.032% as a unique model.

Table 2. Performance metrics

UAV						
Model	Precision	Recall	IoU	F1 score	mIoU	OA
FCN [13]	75.045%	84.016%	65.057%	86.724%	77.456%	91.468%
Unet [13]	73.694%	86.394%	65.991%	87.136%	78.019%	91.658%
DeepLabv3+ [13]	89.289%	93.739%	84.261%	94.715%	90.142%	96.721%
MFFENet [13]	89.326%	93.839%	84.375%	94.756%	90.214%	96.746%
Segformer	96.178%	95.923%	87.795%	96.050%	92.516%	97.695%
SAT						
FCN [13]	62.981%	84.142%	55.716%	84.391%	75.173%	94.972%
Unet [13]	61.795%	78.550%	51.179%	82.316%	72.619%	94.410%
DeepLabv3+ [13]	74.275%	89.187%	68.137%	89.675%	82.397%	96.881%
MFFENet [13]	74.141%	89.141%	67.998%	89.621%	82.318%	96.862%
Segformer	94.118%	93.333%	79.300%	93.720%	88.646%	98.135%
UAV&SAT						
FCN [13]	70.847%	84.014%	61.757%	85.864%	76.515%	92.848%
Unet [13]	67.479%	82.360%	60.115%	85.311%	75.697%	92.653%
DeepLabv3+ [13]	86.128%	92.013%	80.125%	93.563%	88.316%	96.687%
MFFENet [13]	86.133%	92.121%	80.088%	93.608%	88.299%	96.754%
Segformer	95.483%	95.147%	85.157%	95.314%	91.242%	97.681%

The research findings show the efficiency of the Segformer architecture in exactly identifying the affected areas by landslides. The transformer-based architecture of Segformer is characterized by the multi-scale attention mechanism and the long-range dependency modeling that are superior to traditional convolutional neural networks. This has relevance to disaster management and mitigation giving the Segformer the potential to be a valuable tool for accurate landslide detection from high-resolution aerial and satellite imagery. In general, this investigation helps the progress of remote sensed imagery for the landslide segmentation.

4.2 Qualitative Results

To visually assess the performance of the Segformer models, the ground truth is used to compare with our prediction to investigate the power of the Segformer model in segmenting the satellite imagery datasets (SAT, UAV, and UAV+SAT). The final prediction closely matches the ground truth masks and indicates how well the model could differentiate between landslide and non-landslide areas. The fact that the Segformer model can accurately detect and segment the different features in the aerial visuals is an indication that it can be very reliable for landslide mapping and detection.

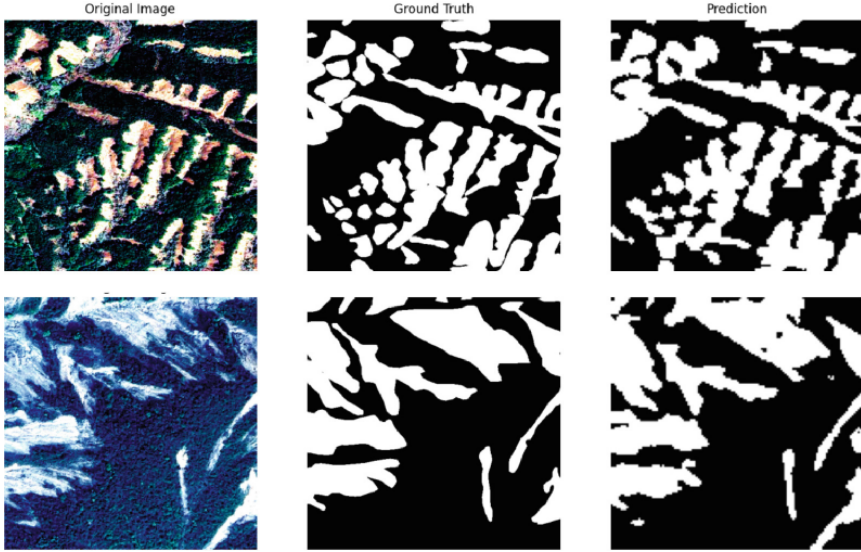


Fig. 3. Inaccurate Predictions

We noticed that the images that display high contrast and clear distinctions between different terrain features demonstrate very strong performance. The ground truth segmentation reflects these clear boundaries, and the model's predictions closely align with this ground truth as shown in Fig. 3. The success in these cases can be attributed to the well-defined patterns in the original image, which facilitate accurate segmentation. The distinct differences in color and texture provide the model with clear cues for differentiation and making accurate predictions.

On the other hand, the images that show low contrast and dark regions were challenging to predict accurately. The subtle features and dark, noisy backgrounds are likely misinterpreted by the model as significant features, leading to incorrect segmentation. The ground truth indicates that there are no significant segmented regions, suggesting that the features to be detected are either very subtle or entirely absent. However, the model's prediction includes several segmented regions that are not present in the ground truth, resulting in false positives (see Fig. 4).

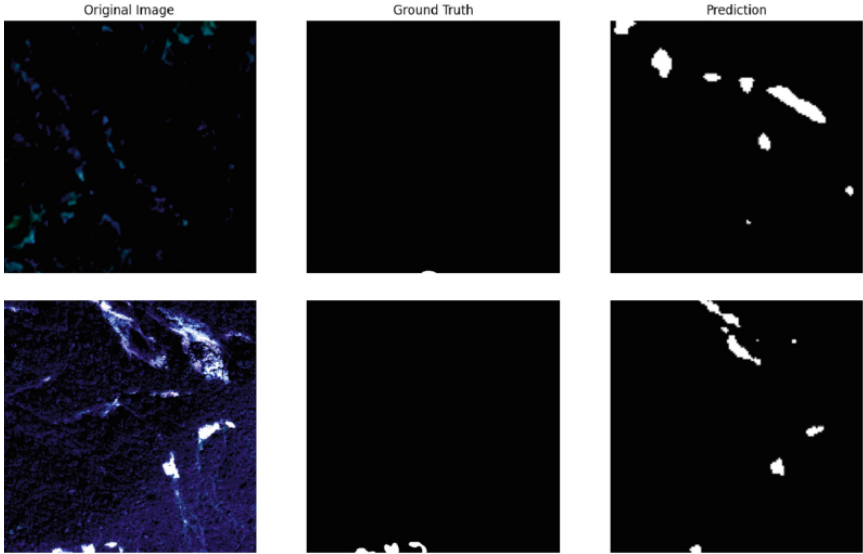


Fig. 4. Inaccurate Predictions

4.3 Discussion and Future Work

The data-based results presented in this section demonstrate the success of the Segformer architecture for landslide detection using high-resolution UAVs and satellite imagery. The Segformer models outperform other latest semantic segmentation methods, such as FCN, U-Net, DeepLabV3+, and MFFENet, across various evaluation metrics and scenarios. The superior performance of Segformer can be attributed to its transformer-based design, which enables it to capture long-range dependencies and model both local and global features effectively. The hierarchical structure of Segformer, combining the MiT backbone and the All-MLP head, allows it to learn multi-scale representations and efficiently merge information from different scales for accurate landslide detection. Nevertheless, the areas of noticeable weak points, as well as prospects for future research, should be mentioned.

The study utilizes a dataset based on the CAS landslide dataset, which covers a wide range of geographic areas. Future research could address common challenges in remote sensing data, such as insufficient content in cropped images due to boundary issues, low proportion of target objects, obstruction by cloud cover, and discontinuities from image stitching.

5 Conclusion

While our study has made significant strides in landslide detection using the Segformer architecture and the CAS landslide dataset, it also highlights several

areas for future research. One key challenge in this field remains the scarcity of high-quality, diverse datasets with comprehensive annotations, which limits the thorough evaluation of segmentation models. The complex and irregular nature of landslide features necessitates advanced computational models capable of capturing long-range dependencies and managing intricate spatial interactions effectively.

Our research demonstrates the potential of transformer-based models like Segformer in addressing these challenges, showcasing their ability to outperform human experts in segmentation tasks. By developing an efficient Segformer environment for identifying landslides in high-resolution UAV and satellite images, we demonstrated superior accuracy in landslide region detection compared to existing systems. This success underscores the importance of further exploring and refining such architectures for landslide detection. In the future, research ought to be extended to assess diversified datasets and enhance data modalities. The role of domain-specific refinements of the Segformer architecture should be investigated to further improve the accuracy of landslide detection.




References

1. Bragagnolo, L., et al.: Convolutional neural networks applied to semantic segmentation of landslide scars. *CATENA* **201** (2021). <https://doi.org/10.1016/j.catena.2021.105189>
2. Chen, L.C., et al.: Encoder-decoder with atrous separable convolution for semantic image segmentation. *arXiv* (2018)
3. Cheng, L., Li, J., Duan, P., Wang, M.: A small attentional YOLO model for landslide detection from satellite remote sensing images. *Landslides* **18**(8), 2751–2765 (2021). <https://doi.org/10.1007/s10346-021-01694-6>
4. Cordts, M., et al.: The cityscapes dataset for semantic urban scene understanding. *arXiv* (2016)
5. Dosovitskiy, A., et al.: An image is worth 16x16 words: transformers for image recognition at scale. *arXiv* (2021)
6. Froude, M.J., Petley, D.N.: Global fatal landslide occurrence from 2004 to 2016. *Nat. Hazards Earth Syst. Sci.* **18**(8) (2018). <https://doi.org/10.5194/nhess-18-2161-2018>
7. Lei, T., et al.: Landslide inventory mapping from bitemporal images using deep convolutional neural networks. *IEEE Geosci. Remote Sens. Lett.* **16**(6) (2019). <https://doi.org/10.1109/LGRS.2018.2889307>
8. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. *arXiv* (2015)
9. Paszke, A., et al.: Pytorch: an imperative style, high-performance deep learning library. *arXiv* (2019)
10. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. *arXiv* (2015)
11. Tang, X., et al.: Automatic detection of coseismic landslides using a new transformer method. *Remote Sens.* **14**(12) (2022). <https://doi.org/10.3390/rs14122884>
12. Xie, E., et al.: Segformer: Simple and efficient design for semantic segmentation with transformers. *arXiv* (2021)

13. Xu, Y., et al.: Cas landslide dataset: a large-scale and multisensor dataset for deep learning-based landslide detection. *Sci. Data* **11**(1) (2024). <https://doi.org/10.1038/s41597-023-02847-z>
14. Zhou, B., et al.: Scene parsing through ade20k dataset. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE (2017). <https://doi.org/10.1109/CVPR.2017.544>
15. Zhou, W., et al.: Mffenet: multiscale feature fusion and enhancement network for RGB-thermal urban road scene parsing. *IEEE Trans. Multimed.* **24** (2022). <https://doi.org/10.1109/TMM.2021.3086618>



Vision-Based Human Fall Detection Using 3D Neural Networks

Say Meng Toh¹ , Na Helian¹  , Kudiwa Pasipamire² , Yi Sun¹ ,
and Tony Pasipamire²

¹ University of Hertfordshire, Hatfield AL10 9AB, UK
{s.m.toh, n.helian, y.2.sun}@herts.ac.uk

² Delight Supported Living, Letchworth SG6 3EG, UK
{kudiwa.pasi, tonypasi}@delightsupportedliving.co.uk

Abstract. The use of Machine Learning to monitor old people is crucial in providing immediate assistance and potentially life-saving interventions. With the rapid innovation in the field of Artificial Intelligence and Computer Vision, fall detection has seen significant improvements in accuracy and efficiency. Traditionally, 2D Convolutional Neural Networks (CNN) have been the main focus in fall detection research. However, these approaches have several drawbacks, including that 2D CNNs are primarily designed for spatial feature extraction and may not fully capture the temporal dynamics across multiple frames. This is because, for 2D CNN, the video frames are averaged out on time dimension within a time window. This project aims to explore and validate the use of 3D Convolutional Neural Networks (CNN) for fall detection, specifically in care home settings. The proposed 3D CNN keeps all frames in the time dimension (without averaging out video frames) and therefore can capture spatiotemporal dynamics of fall events more effectively, potentially enhancing detection accuracy. Experiment results indicate that the proposed 3D CNN achieved a G-Means, the geometric mean of recall and specificity, of 96.92%, an improvement of 1.9% over the 2D CNN.

Keywords: Neural Network · Fall Detection · Deep Learning · Machine Learning · Computer Vision · Elderly Care

1 Introduction

People become more vulnerable and likely to fall as they age, especially if they have a long-term health condition. According to the NHS, around 1 in 3 adults over 65 and half of people over 80 will have at least one fall a year [1]. While most falls do not cause serious injuries, there are risks for falls to become recurrent and result in head or hip injuries. More than 95% of hip fractures are caused by falling [2]. Falls are estimated to cost the NHS more than £2.3 billion per year [3].

The medical outcomes of falls depend on the swiftness with which individuals receive treatment or care. In this sense, an automated fall detection system can improve the response time of carers to help mitigate the consequences of falls. Fall detection systems fall under two categories: vision and non-vision based. A vision approach relies on

cameras recording videos while a non-vision approach relies on wearables and sensors of data, like acceleration [4]. Non-vision approaches have their drawbacks, especially in the context of elderly monitoring because of the intrusiveness of wearable sensors. Therefore, a constant vision-based approach to fall detection was used. A paper authored by Espinosa et al., uses a 2D CNN to perform fall detection [5]. This project aims improve the algorithm to enhance model performance by leveraging 3D CNNs to capture temporal information.

2 Previous Works

There have been many applications of fall detection systems throughout the years. These systems mainly fall under two categories: sensor-based and vision-based. Given that this study utilizes a camera system for fall detection, the primary focus will be on vision-based detection methods. However, a brief review of sensor-based methods is also provided for context and comparison.

2.1 Sensor-Based

Sensor-based approaches typically involve wearable devices or ambient sensors to monitor motion and detect falls. These sensors can be placed at various locations in the environment and the human body. From a commercial standpoint, wearable sensor technology is the most utilized type due to its low costs. Yin et al. [6] utilized wearable sensors to detect abnormal activities by training a Support Vector Machine (SVM) on normal activity data, classifying deviations as anomalies. Three sensors were placed on the shoulder to capture motion data from different body parts. Their approach achieved an Area under ROC Curve (AUC) of 0.985. Additionally, Kangas et al. [7] used accelerometers and gyroscopes for fall detection. While the results demonstrate the effectiveness of these approaches, the use of wearable sensors can be intrusive.

2.2 Vision-Based

Vision-based fall detection systems utilize computer vision and image processing techniques with data from various types of cameras, including RGB cameras, motion camera systems, and Kinect cameras. Previous research relied on traditional machine learning techniques. Harrou et al. [8] combined a Multivariate Exponentially Weighted Moving Average (MEWMA) chart [9] with an SVM for fall detection, achieving 96.66% accuracy on the UR Fall Detection dataset (URFD) [10] and 97.02% on the Fall detection dataset (FDD) [11].

Multiple studies have also extracted human skeleton features using pose estimation algorithms to classify falls. Mobasheri et al. [19] and Ramirez et al. [20] extracted the key points on a human body and used a long-short-term memory (LSTM) network, achieving 98% accuracy and 81% accuracy respectively.

Many studies have also explored the use of 2D CNNs for fall detection. Espinosa et al. [5] used optical flow and windowing techniques to capture and analyze movement data. Experiments determined that a 1-s window with a 0.5-s overlap provided the best

results. Optical flow data was averaged across frames within each window and classified using a 2D CNN achieving 95.64% accuracy and a 97.43% F1-Score. Additionally, they implemented traditional machine learning techniques, such as Support Vector Machines (SVM), to evaluate performance differences.

Several studies have shown that 3D CNNs offer improvements over 2D CNNs. Particularly in the task of action recognition [16–18]. Given these advantages, this study proposes using 3D CNNs for fall detection due to the limitation of 2D CNN for this application.

3 Methodology

3.1 Dataset

The UP-Fall dataset [13] is used. It contains a collection of video data that depict various activities, including both fall and non-fall incidents. The dataset comprises of 17 subjects (9 males and 8 females) ranging from 18 to 24 years of age performing 11 activities. Each activity was repeated 3 times taken from two Microsoft Life-Cam Cinema cameras from two different perspectives: a lateral view and a frontal view. This combination gives a total of 1122 videos. The length of the videos can be seen in Table 1, each second in the videos contains 18 frames.

The dataset provides labels from 1–11, each describing a different action.

Table 1. UP-Fall dataset labels.

Activity ID	Description	Duration(s)
1	Falling forward using hands	10
2	Falling forward using knees	10
3	Falling backwards	10
4	Falling sideward	10
5	Falling sitting in empty chair	10
6	Walking	60
7	Standing	60
8	Sitting	60
9	Picking up an object	10
10	Jumping	30
11	Laying	60

3.2 Preprocessing

Windowing

The windowing approach divides the falls into smaller data segments of time series.

Segmentation techniques can be categorized into three main groups: activity-defined windows, event-defined windows, and sliding windows [14].

A sliding window approach – the approach described in [5] – was used in this research to capture the temporal dependency between samples. Each window is formed by segmenting the video data into fixed-length intervals. These windows can overlap in time to ensure continuous monitoring and capture of sequential movements. A window size of 1 s with a 0.5-s overlap was chosen as it gave the best performance [5]. This means that each new window starts halfway through the previous window, creating multiple overlapping windows. The outputs of this windowing are multiple 1-s window length series of 3D images which are to be classified.

Optical Flow

Feature extraction is a method to transform raw data into features that can be processed and analyzed while preserving the relevant information. There are many techniques to perform it and optical flow is well used for extracting features from videos. Optical flow is a technique used to estimate motion between two consecutive frames in a video sequence. Using this technique, we can detect the motion that a subject takes over time. Among various optical flow estimation methods, the Farneback method, introduced by Farneback in 2003 [15], is a popular choice due to its efficiency and robustness. The Farneback method is based on polynomial expansion. It approximates the neighbourhood of each pixel in both frames with quadratic polynomials using convolution kernels. By comparing the polynomial coefficients from consecutive frames, the displacement field is derived. This displacement field indicates how each pixel moves between frames. The algorithm then refines the flow estimations over several iterations, considering large neighbourhoods to capture larger motions.

Dataset Downsampling

Upon further investigation of the dataset, there is a large imbalance between the number of fall and non-fall samples. Specifically, the dataset contained a total of 1910 fall windows and 62562 non-fall windows. The non-fall windows are downsampled to the same number of fall windows (1910) and then split into two different sets for training and testing. 80% of the falls are taken along with the same number of non-falls creating a new balanced dataset for training. This training dataset contained 1526 for each set of falls and non-falls, leaving the remaining 59184 non-falls and 384 falls as the test dataset.

3.3 Neural Network Architecture

The research by Espinosa et al. used a 2D Convolutional Neural Network for feature extraction and classification. This 2D architecture serves as the foundation for the proposed 3D CNN architecture.

A 3D Convolutional Neural Network was developed to enhance the model architecture and better capture the temporal dynamics in the video data. In contrast to the original 2D CNN approach, which requires averaging optical flow across a window of 18 frames to produce a 2D feature map – one channel as shown in Fig. 1, the 3D CNN implementation retains more detailed motion information by processing each of the optical flow

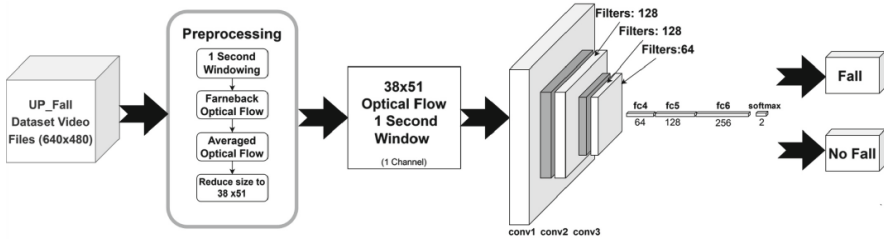


Fig. 1. Espinosa et al. 2D CNN architecture.

frames without the need for averaging – removing the averaged optical flow process. The architecture of the 3D CNN is as follows:

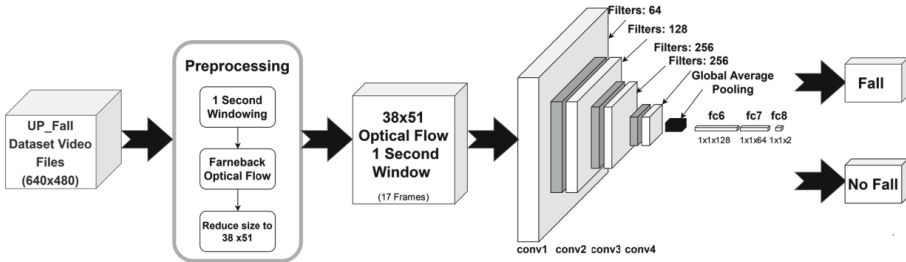


Fig. 2. Proposed 3D CNN Architecture.

Figure 1 and 2 illustrate the difference between the two architectures. The 3D CNN in Fig. 2. Does not average the optical flow within a window. This results in 3D data, incorporating 18 frames to capture spatiotemporal dynamics. Lastly, the 3D CNN has 4 3D Convolutional layers instead of three.

3.4 Evaluation Metrics

In evaluating the performance of the 3D CNN for fall detection, we considered several key metrics: Accuracy, Recall (Sensitivity), Specificity, F1-Score, and G-Means.

Accuracy gives a general sense of the model's performance but is not sufficient on its own. Recall (or Sensitivity) is important in fall detection because the application requires as many falls are detected as possible, minimizing the risk and consequences of a missed fall. Specificity is another important metric that indicates the number of wrongly predicted non-falls. For a commercial product, maintaining high specificity is important to reduce the number of false alarms, which can erode user trust and increase operational costs. F1-Score is a metric that combines both precision and recall, which is particularly useful when dealing with imbalanced datasets. Finally, G-Means is a metric that highlights how well-balanced a model is in identifying falls while minimizing wrongly predicted non-falls.

4 Experimentation and Results

This section presents a detailed analysis of the experimental setup, including comparisons between 2D and 3D CNN models, the impact of hyperparameters and video frame resolution, and the effect of class balancing on model performance. All experiments were conducted using an NVIDIA RTX 4060 and PyTorch framework.

To ensure robust evaluation of the models, all experiments were conducted using 5-fold cross-validation. This technique divides the training dataset into five equal parts, with four parts used for training and one part for validation in each fold. This process is repeated five times, with each part used exactly once for validation. The model used for the final test evaluation is the best model obtained from the cross-validation process. Specifically, the model with the highest F1-Score across the folds is selected and then evaluated on the test set.

4.1 2D vs 3D CNN

This experiment aims to compare the performance of the Espinosa et al. 2D CNN and the proposed 3D CNN in detecting falls, with the hypothesis that the 3D CNN will provide improved detection accuracy. In this experiment, the hyperparameters of the 3D CNN are set to a learning rate of 0.0001, a batch size of 16, and a kernel size of 3x3x3. These values were selected as they are generally considered to be a good baseline to start from.

Figure 3 displays the confusion matrices derived from the models on the test dataset. Table 2 compares the proposed 3D CNN with Espinosa et al. 2D CNN [5]. The results in this table are different from the ones produced in [5] due to the difference in how the dataset was split. To ensure comparability, the dataset methodology described in Sect. 3.2 was used.

The experiment results indicate that the proposed 3D CNN outperforms the 2D CNN across all key metrics. The notable improvements, particularly in F1-Score highlight the 3D CNN’s capability to capture spatiotemporal dynamics, leading to better fall detection performance. However, the 3D CNN took three times longer to train compared to the 2D CNN – 10 min for 2D CNN and 30 min for 3D CNN.

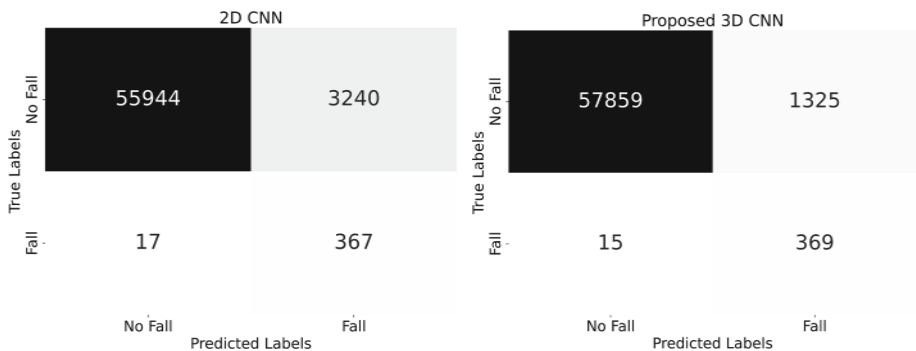


Fig. 3. 2D CNN vs 3D CNN confusion matrices.

Table 2. 2D vs 3D CNN test performance metrics.

	Accuracy	Recall	Specificity	F1-Score	G-Means
2D CNN	94.53	95.57	94.53	18.39	95.04
Proposed 3D CNN	97.75	96.09	97.76	35.51	96.92

4.2 Hyperparameters

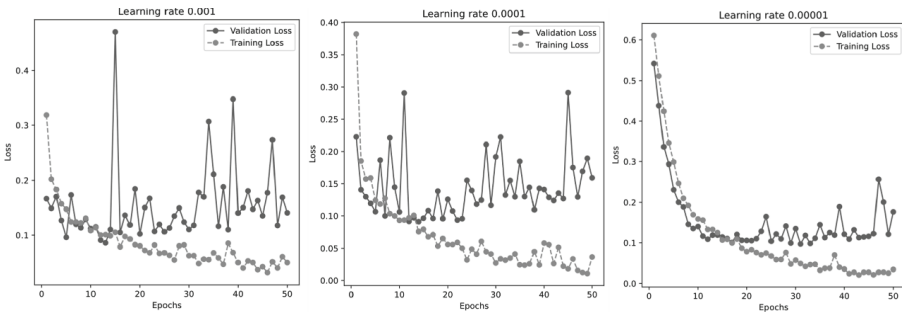
In this section, the impact of hyperparameters on the performance of the 3D CNN is explored. Specifically, the focus is on learning rate, batch size, and kernel size. For all experiments, the best-performing hyperparameter value from each test is used in subsequent experiments.

Learning Rate

Three learning rates, including 0.001, 0.0001, and 0.00001, were tested to determine the optimal value for model convergence.

Figure 4 displays the training and validation loss of the proposed 3D CNN with 3 different learning rates. Table 3 compares the three learning rates across different metrics.

A learning rate of 0.0001 achieves the highest test accuracy, specificity, F1-Score, and G-means with recall slightly lower by 0.8%. However, a learning rate of 0.0001 is a strong contender as it provides the smoothest training graph as seen in Fig. 4.

**Fig. 4.** Training and validation loss for learning rates.

Batch Size

The next hyperparameter tested was batch size including 8, 16, 32, and 48.

Figure 5 displays the confusion matrices from the models on the test dataset. Table 4 shows the performance metrics for the different batch sizes.

Looking at Table 4, a batch size of 16 achieves the overall best performance with the highest accuracy, highest precision, and a balanced recall, resulting in the highest F1-Score and G-Means.

Table 3. Test performance metrics on learning rates.

Learning rate	Accuracy	Recall	Specificity	F1-Score	G-Means
0.001	96.93	96.88	96.93	28.90	96.75
0.0001	97.75	96.09	97.76	35.51	96.92
0.00001	96.61	96.88	96.61	26.95	96.74

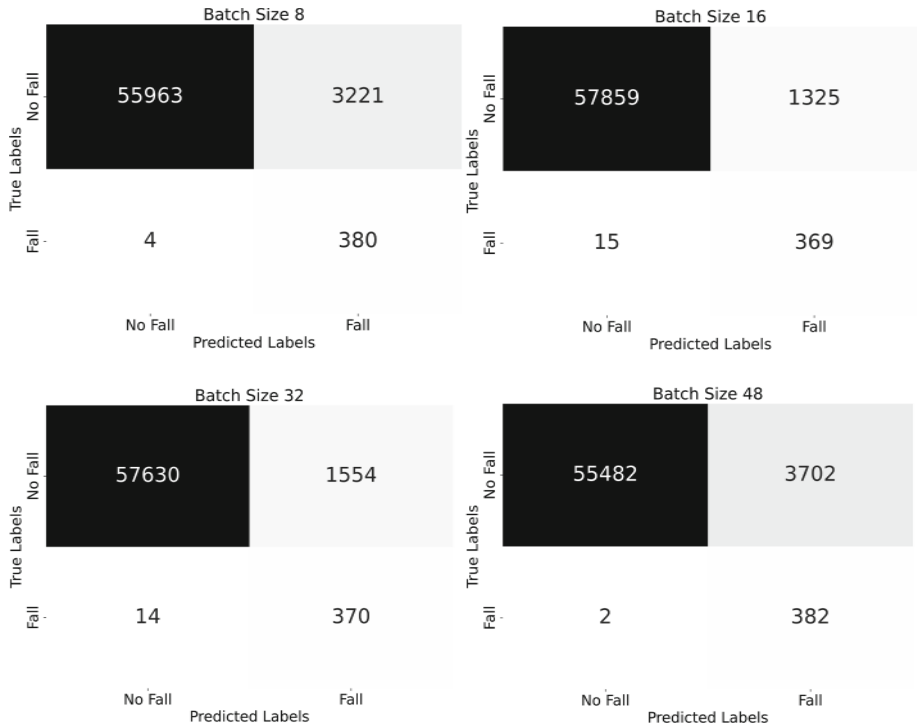


Fig. 5. Confusion Matrices for batch sizes.

Table 4. Test performance metrics for batch sizes.

Batch Size	Accuracy	Recall	Specificity	F1-Score	G-Means
8	94.59	98.96	94.56	19.07	96.73
16	97.75	96.09	97.76	35.51	96.92
32	97.37	96.35	97.37	32.06	96.86
48	93.78	99.48	93.74	17.10	96.57

Kernel Size

The next hyperparameter tested was kernel size, including $3 \times 3 \times 3$ and $2 \times 2 \times 2$.

Figure 6 is the confusion matrices for the model on the test dataset. Table 5 shows the performance metrics on the test dataset.

Table 5 shows that a kernel size of $3 \times 3 \times 3$ produces better results. While accuracy and specificity remain similar, with a difference of 0.3% and 0.16% respectively, precision, recall, F1-Score, and G-Means see a difference of 2–3%.

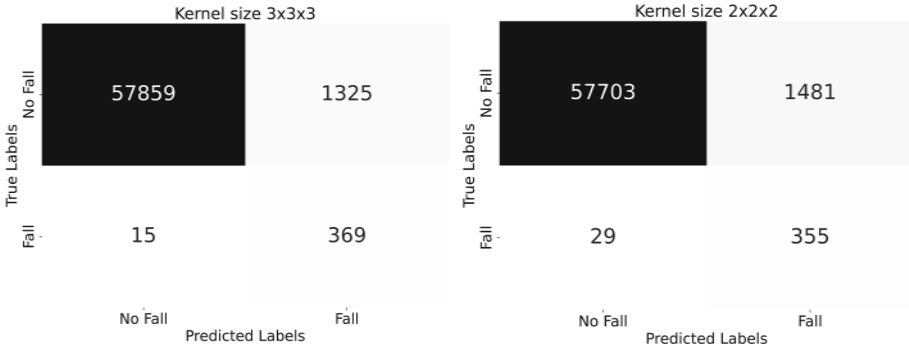


Fig. 6. Confusion matrices for kernel size.

Table 5. Test performance metrics for kernel size.

Kernel Size	Accuracy	Recall	Specificity	F1-Score	G-Means
$3 \times 3 \times 3$	97.75	96.09	97.76	35.51	96.92
$2 \times 2 \times 2$	97.47	92.45	97.50	31.98	94.94

4.3 Input Image Resolution

The resolution of input images is another critical factor that can impact the model's performance. Higher resolutions may provide more detailed information, potentially improving accuracy, but at the cost of increased computation resources and time.

To determine the optimal resolution for the baseline CNN model, several resolutions were tested, including 38×51 , 57×76 , and 76×102 .

Figure 7 is the confusion matrices for the model on the test dataset. Table 6 shows the performance metrics on the test dataset.

Increasing resolution was hypothesized to provide more detailed information for the model, potentially leading to improved performance. However, the results show that the performance decreased. This might be because high resolution focuses more on detailed local information, which may not be necessary for fall detection. In addition, increasing resolution significantly increased training times. For 5-fold cross-validation, a resolution of 38×51 took roughly 30 min to train, and 57×76 trained in an hour while increasing the resolution to 76×102 increased training time to 20 h.

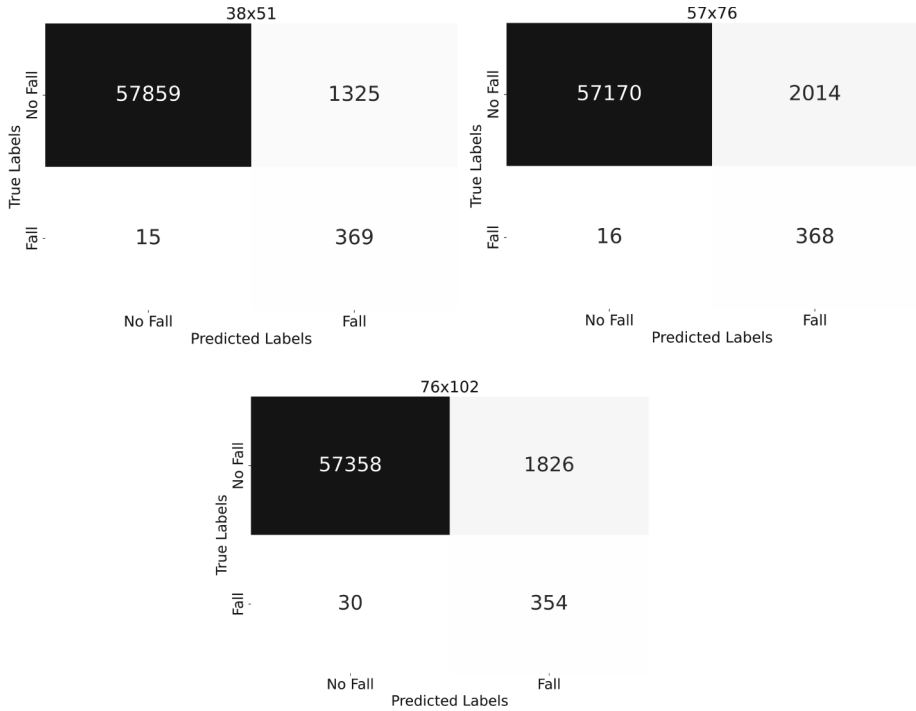


Fig. 7. Confusion matrices for input image resolutions.

Table 6. Test performance metrics for input image resolution.

Resolution	Accuracy	Recall	Specificity	F1-Score	G-Means
38 x 51	97.75	96.09	97.76	35.51	96.92
57x76	96.59	95.84	96.60	26.61	96.22
76x102	96.88	92.19	96.91	27.61	94.52

4.4 Ratio of Falls to Non-falls

The ratio of falls to non-falls in the dataset is a crucial factor that can significantly impact the performance and generalization of the model. In real-world scenarios, falls are rare events compared to non-falls, leading to highly imbalanced datasets. Training models on imbalanced datasets can cause the model to overpredict the majority class (non-falls), leading to poor detection of falls. On the other hand, using a perfectly balanced dataset is not representative of real-world conditions and might not reflect the model's true performance in practical applications.

To investigate the effect of the different ratios of falls to non-falls on the model's performance, several experiments were conducted with varying degrees of class imbalance, including 1:1, 1:3, 1:5, and 1:10.

Figure 8 is the confusion matrices for the model on the test dataset. Table 7 shows the performance metrics on the test dataset.

As the results show, the number of non-falls detected increased, and the number of falls detected decreased as the imbalance ratio increased. This is reflected in both the confusion matrices and Table 7. The optimal ratio depends on the acceptable level of risk and specific application context. For most practical applications, a mildly imbalanced ratio may be a sufficient approach, reducing false positives while maintaining a good detection rate for falls.

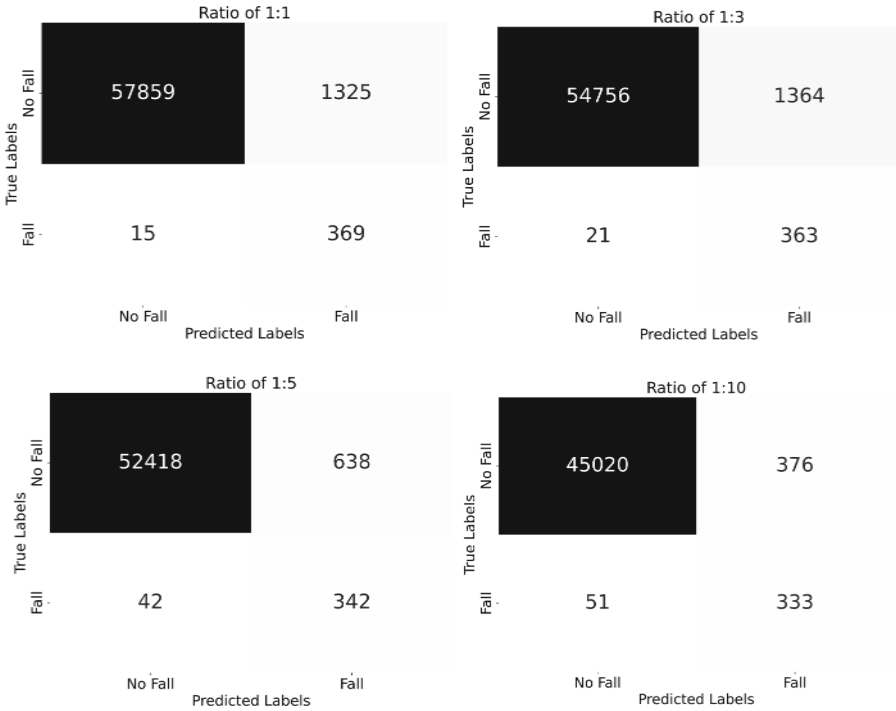


Fig. 8. Confusion matrices of ratio of falls to non-falls.

Table 7. Test performance metrics for ratio of falls to non-falls.

Ratio of falls to non-falls	Accuracy	Recall	Specificity	F1-Score	G-Means
1:1	97.75	96.09	97.76	35.51	96.92
1:3	97.55	94.53	97.57	34.39	96.04
1:5	98.73	89.06	98.90	50.15	93.85
1:10	99.07	86.72	99.17	60.93	92.99

4.5 Comparative Analysis

The proposed vision-based fall detection system is comparable to other state-of-the-art studies, as discussed in Sect. 2.2. This comparative analysis will provide an understanding of the proposed model's performance relative to different approaches.

Espinosa et al. [5] implemented a Support Vector Machine (SVM) and Random Forest (RF) achieving a G-Means of 35.63% and 36.12% respectively. The proposed 3D CNN outperforms both traditional methods significantly, likely due to the inability of SVMs and RFs to capture complex temporal and spatial patterns. Similarly, Ramirez et al.'s LSTM [20] achieved an accuracy of 81.14% but with a recall of 31.82%, potentially suggesting the LSTM's limited feature extraction capabilities compared to 3D CNNs. Mobasheri et al.'s LSTM's achieved an accuracy of 98.5%, around 1% higher than the proposed CNN. This may be due to the difference in the preprocessing method, as Mobasheri uses skeletal information.

However, it is important to note that direct comparisons between these models may be inconclusive due to the differences in evaluation methods and dataset handling, which can significantly impact the performance metrics.

5 Conclusion

This research proposed a 3D CNN for human fall detection. The experiment results demonstrated significant improvements over the 2D CNN. However, the 3D CNN took three times longer to train compared to the 2D CNN. Nevertheless, it is worth noting that there is an unnoticeable difference in time cost between 2D CNN and 3D CNN when deploying a model for prediction.

By carefully tuning hyperparameters and improving preprocessing, the model achieved better performance and generalization, achieving an accuracy of 97.55%, 3% better than 2D CNNs. Parameter optimization played a crucial role in achieving these results, as adjustments in learning rate, batch size, and data resolution contributed to the model's overall effectiveness. However, there is an inherent trade-off between fall and non-fall prediction performance. Achieving a balance between detecting falls and minimizing false alarms remains a challenge.

Further work is needed to validate the robustness of the models by testing them on diverse datasets. This will help provide a more comprehensive evaluation of the model's effectiveness. Additionally, deploying the model in real-world settings is crucial to identify improvements to handle variations in real-life data.

Acknowledgments. This work was funded by UK Research and Innovation, Knowledge Transfer Partnership project – partnership number: 13139.

Disclosure of Interests. The authors have no competing interests.

References

1. NHS, Falls. <https://www.nhs.uk/conditions/falls>. Accessed 23 May 2024

2. Pakkari, J., et al.: Majority of hip fractures occur as a result of a fall and impact on the greater trochanter of the femur: a prospective controlled hip fracture study with 206 consecutive patients. *Calcified Tissue Int.* **65**, 183–187 (1999)
3. NICE. Falls in older people: assessing risk and prevention, NICE Clinical guideline. <https://www.nice.org.uk/guidance/cg161>. Accessed 20 June 2024
4. Salimi, M., Machado, J.J.M., Tavares, J.M.M.S.: Using deep neural networks for human fall detection based on pose estimation. *Sensors* **22**, 4544 (2022)
5. Espinosa, R., Ponce, H., Gutiérrez, S., Martínez-Villaseñor, L., Brieva, J., Moya-Albor, E.: A vision-based approach for fall detection using multiple cameras and convolutional neural networks: A case study using the UP-Fall detection dataset. *Comput. Biol. Med.* **115** (2019)
6. Yin, J., Yang, Q., Pan, J.J.: Sensor-based abnormal human-activity detection. *IEEE Trans. Knowl. Data Eng.* **20**, 1082–1090 (2008)
7. Kangas, M., Konttila, A., Lindgren, P., Winblad, I., Jämsä, T.: Comparison of low-complexity fall detection algorithms for body attached accelerometers. *Gait Posture* **28**, 285–291 (2008)
8. Harrou, F., Zerrouki, N., Sun, Y., Houacine, A.: Vision-based fall detection system for improving safety of elderly people. *IEEE Instrum. Meas. Mag.* **20**, 49–55 (2017)
9. Lowry, C.A., Woodall, W.H., Champ, C.W., Rigdon, S.E.: A multivariate exponentially weighted moving average control chart. *Technometrics* **34**, 46–53 (1992)
10. Kwolek, B., Kepski, M.: Improving fall detection by the use of depth sensor and accelerometer. *Neurocomputing* **168**, 637–645 (2015)
11. Adhikari, K., Bouchachia, H., Nait-Charif, H.: Fall detection Dataset. <https://falldataset.com>. Accessed 22 June 2024
12. Rougier, C., Meunier, J., St-Arnaud, A., Rousseau, J.: Monocular 3D head tracking to detect falls of elderly people. *IEEE Eng. Med. Biol. Soc.* 6384–6387 (2006)
13. Martínez-Villaseñor, L., Ponce, H., Brieva, J., Moya-Elbor, E., Núñez-Martínez, J., Peñafort-Asturiano, C.: UP-Fall detection dataset: a multimodal approach. *Sensors (Basel)*, **19** (2019)
14. Banos, O., Galvez, J.M., Damas, M.: Window size impact in human activity recognition. *Sensors* **14**, 6474–6499 (2014)
15. Farnebäck, G.: Two-frame motion estimation based on polynomial expansion. In: Bigun, J., Gustavsson, T. (eds.) SCIA 2003. LNCS, vol. 2749, pp. 363–370. Springer, Cham (2003). https://doi.org/10.1007/3-540-45103-x_50
16. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3D convolutional networks. In: IEEE International Conference on Computer (ICCV), pp. 4489–4497 (2015)
17. Ji, S., Xu, W., Yang, M., Yu, K.: 3D Convolutional neural networks for human action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **35**, 221–231 (2013)
18. Carreira, J., Zisserman, A.: Quo Vadis, action recognition? A new model and the kinetics dataset. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 4724–2733 (2017)
19. Mobasheri, B., Tabbakh, S.R.K., Forghani, Y.: An approach for fall prediction based on kinematics of body key points using LSTM. *Int. J. Environ. Res. Public Health* **19**(21) (2022)
20. Ramirez, H., Velastin, S.A., Aguayo, P., Fabregas, E., Farias, G: Human activity recognition by sequences of skeleton features. *Sensors* **22**(11) (2022)



Drone-Assisted Infrared Thermography and Machine Learning for Enhanced Photovoltaic Defect Detection: A Comparative Study of Vision Transformers and YOLOv8

Ammar Memari^(✉)  and Tarek Debich

Jade University of Applied Sciences, Friedrich-Paffrath-Straße 101, 26389
Wilhelmshaven, Germany
ammam.memari@jade-hs.de, tarek.debich@student.jade-hs.de
<https://www.jade-hs.de/>

Abstract. This paper presents a comparative study on the application of drone-assisted infrared thermography coupled with state-of-the-art machine learning models, including Vision Transformers (ViTs) and YOLOv8, for efficient and accurate defect detection in Photovoltaic (PV) systems. The research outlines the methodology for on-site inspections and details the integration of drones to capture high-resolution thermal imagery, identifying minute anomalies indicative of potential system failures. By employing advanced image processing techniques and training AI models, the study compares the performance of these models in accurately identifying and classifying PV defects. A segmentation model based on TensorFlow was trained and used to detect the location of PV panels in the camera imagery, followed by the separate application of YOLOv8 and ViTs to classify defects in the detected PV panels. The dataset used to train the models was carefully curated and prepared specifically for this study, representing another contribution of this paper. This approach not only enhances detection capabilities but also streamlines the processes of data collection, analysis, and reporting, ultimately leading to improved decision-making for system operators and stakeholders.

Keywords: Solar energy · Drones · Thermography · Object detection · Object classification · YOLOv8 · Vision Transformers

1 Introduction and Background

Global energy demand is rising due to economic growth and population increase. The International Energy Agency (IEA) projects 305 GW of renewable energy capacity will be added annually between 2021 and 2026, a 60% increase [15]. A large portion comes from PV systems, whose performance depends on individual

modules. Defective modules reduce efficiency, profitability, and pose safety risks. Defects can occur during manufacturing, installation, or through degradation, making regular inspections crucial.

This project leverages computer vision and deep learning to inspect PV systems. Machine learning models, including TensorFlow [1], YOLOv8 [19], and ViTs [13], identify defects in drone-captured images of PV panels (see Fig. 1 and Table 1).

Manual inspections using thermal cameras are laborious and expensive, taking up to 200 h for a 3 MW solar farm. Drone technology with infrared thermography reduces inspection time from days to hours. This research aims to develop AI-based methods for defect detection and reporting in PV imagery, streamlining maintenance and monitoring.

Fundamentals of Photovoltaics. PV technology converts sunlight into electricity, playing a key role in renewable energy systems [15,21]. Solar cells generate electricity via the PV effect, where photons free electrons in semiconductor materials [27]. PV systems include inverters to convert DC to AC, integrating into domestic applications and public grids.

Routine inspection maintains PV system efficiency, reliability, and safety. Thermography, in accordance with DIN EN 62446-1 (VDE 0126-23-1), helps detect issues early, extending system lifespan and ensuring optimal performance [12].

Drone Technology and Thermography. Drones equipped with infrared sensors detect temperature anomalies, enhancing inspection efficiency. Thermal imaging in solar panel inspections enables early fault detection, efficient maintenance, and valuable documentation [29].

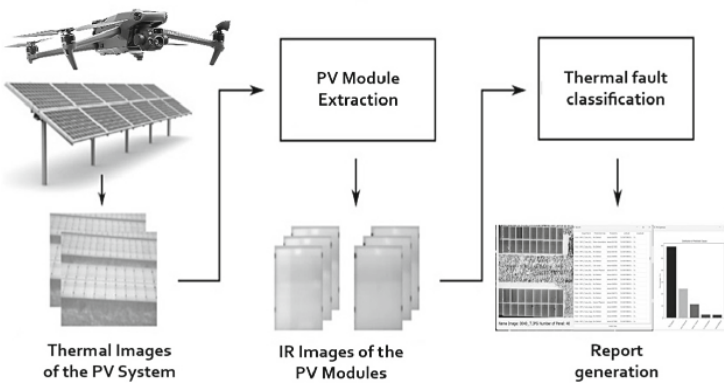
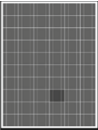
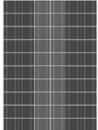
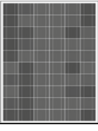
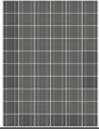


Fig. 1. Workflow of the proposed thermal fault detection system

Table 1. Shape of a character in dependence on its position in a word

Thermal Anomaly Type	Causes	Production Losses	Pattern
Hot Spot	Typically resulting from shadow effects, soiling, or physical damage.	May reduce a module's production by up to 90%.	
Bypassed String	Damage to bypass diodes often results from lightning strikes, causing them to conduct current continuously.	Loss could reach 33% or 66% of a module's production, depending on the affected area.	
Single and Multi-Diode Issues	Similar causes and effects to Bypassed String but isolated to specific diodes.	Losses can amount to up to a third of production per affected string.	
Potential Induced Degradation (PID)	Emerging PID signs without definitive evidence.	Lower production losses than fully developed PID, as it's in early stages.	

Manual inspections are time-consuming, labor-intensive, and potentially hazardous, often overlooking subtle issues. Ground-based camera inspections also fall short in precision and efficiency. Drones, however, offer increased speed and efficiency, enhanced safety, cost-effectiveness, comprehensive coverage, and high-resolution thermal imaging capabilities, making them preferable for solar panel inspections. The IEC TS 62446-3 standard underscores the importance of thermal imaging in PV module maintenance, advocating drone use for accurate and detailed thermographic inspections [16].

2 Related Work

The landscape of defect detection in PV systems has evolved significantly with the advent of advanced machine learning (ML) and image processing techniques. This section reviews pertinent studies that have informed and influenced the methodology and findings of our research on drone-assisted infrared thermography and machine learning models, specifically ViTs and YOLOv8, for PV defect detection.

2.1 Recent Relevant Literature

Prabhakaran et al. [25] conducted an extensive analysis of various methods, emphasizing the crucial need for accurate fault identification to maintain PV panel efficiency. Among the ML models evaluated, AlexNet stood out with an F1-score of 0.86 and 85.56% accuracy, setting a benchmark for PV defect detection.

Bo et al. [5] reviewed ML applications for health monitoring in renewable energy systems, providing a comprehensive overview of the current state and future directions in PV fault detection.

Oviedo et al. [24] used a bibliometric approach to analyze AI applications in PV fault diagnosis, offering insights into prevalent methodologies and the necessity for robust AI solutions in the sector. Fiorese et al. [14] developed a deep learning-based semantic segmentation model using Deeplabv3 with a ResNet-50 backbone for detecting defects in electroluminescence images of silicon PV cells, achieving high accuracy and showcasing the potential of deep learning for precise defect localization.

Li et al. [22] developed an online defect detection system for large-scale PV plants using drones and edge computing, integrating deep learning and transfer learning to enhance detection accuracy. Chen et al. [8] reviewed remote sensing (RS) technology applications in PV system development, emphasizing its utility in large-scale, high-resolution data collection for efficient fault detection.

Bommes et al. [6] proposed an anomaly detection framework for infrared images of PV modules using supervised contrastive learning, achieving high AUROC scores and demonstrating strong generalization capabilities.

Açıköz [2] improved the YOLOv7 model for automatic crack detection in PV cells, indicating ongoing advancements in object detection algorithms. Li et al. [20] reviewed artificial neural networks (ANN) for PV fault detection, providing insights into model configurations and performance metrics, while Ali et al. [4] developed an SVM model with a hybrid feature vector for detecting hotspots in PV panels using infrared thermography, achieving high accuracy.

Ishak et al. [17] reviewed image processing algorithms for detecting hotspots in PV modules, suggesting future research directions for improved accuracy. Ahmed et al. [3] utilized isolated and transfer learned deep neural models to classify PV panels using infrared thermographic images, with their isolated model outperforming traditional deep learning models in defect detection.

These studies collectively highlight the critical role of advanced ML techniques and image processing methods in enhancing the accuracy and efficiency of PV defect detection. Our research builds upon these foundational works to further improve the detection and classification of PV defects.

2.2 Comparative Analysis of Related Studies to Our Approach

Our study integrates drone-assisted infrared thermography with advanced machine learning models, specifically ViTs and YOLOv8, for defect detection in PV systems. We compare our approach with relevant studies by focusing on some key aspects:

Vision Transformers. *Our Approach:* We use ViTs for classifying defects in PV panels detected by our custom segmentation model. ViTs are effective in capturing global image contexts, crucial for accurate defect detection in high-resolution thermal images. We also compare the performance of ViTs and YOLOv8, demonstrating their respective strengths and limitations.

Related Studies: Fioresi et al. [14] use a Deeplabv3 model with a ResNet-50 backbone for semantic segmentation but do not explore ViTs. Bommès et al. [6] employ a ResNet-34 CNN for anomaly detection, focusing on supervised contrastive learning instead of ViTs.

Custom Dataset. *Our Approach:* We built a custom dataset of high-resolution thermal imagery of PV systems using drones, capturing a range of defect types and conditions. This dataset addresses the scarcity of publicly available, high-quality thermal image datasets for PV defect detection.

Related Studies: Fioresi et al. [14] introduced the UCF EL Defect dataset with 17,064 EL images. Li et al. [22] developed a solution using images from PV plants but with limited dataset details. Ali et al. [4] and Ishak et al. [17] emphasize the need for extensive datasets for improved defect detection.

YOLOv8. *Our Approach:* We use YOLOv8 for real-time defect classification in PV panels. Our study compares the performance of YOLOv8 and ViTs, highlighting their strengths in different defect detection aspects.

Related Studies: Açıkgöz [2] employs an improved YOLOv7 model for crack detection in PV cells, without comparing it with ViTs. Zhang et al. [31] use YOLOv5 for defect identification, showcasing YOLO models' effectiveness.

Custom Segmentation Model. *Our Approach:* We developed a TensorFlow-based segmentation model to detect PV panels' locations in thermal imagery. This model ensures accurate defect localization, integrating with YOLOv8 and ViTs for defect classification.

Related Studies: Fioresi et al. [14] use Deeplabv3 for segmentation but do not combine it with multiple classification models. Bommès et al. [6,7] highlight robust segmentation and detection models' importance, but focus on anomaly detection using CNNs.

Streamlined Process from Data Collection to Inference. *Our Approach:* We present a comprehensive end-to-end process, including drone data collection, image preprocessing, model training, and inference. Our approach features a graphical user interface (GUI) tool for data collection and analysis, alongside drone operation guidance for consistent, high-quality data capture.

Related Studies: Li et al. [22] propose an edge computing solution with UAVs for visual inspection but lack detailed end-to-end workflow guidance. Chen et al. [8] review remote sensing techniques but do not provide a specific workflow.

Oliveira et al. [23] and Ahmed et al. [3] emphasize the need for automation and advanced machine learning integration.

Our study presents a novel and comprehensive approach to PV defect detection by integrating ViTs and YOLOv8 with a custom-built dataset and segmentation model. The streamlined process from data collection to inference, including drone operation guidance and a GUI tool, sets our research apart, enhancing the efficiency, accuracy, and practicality of PV defect detection.

3 Materials and Methods

The full workflow of the proposed method including training the models and preparing the dataset is depicted in details in Fig. 2.

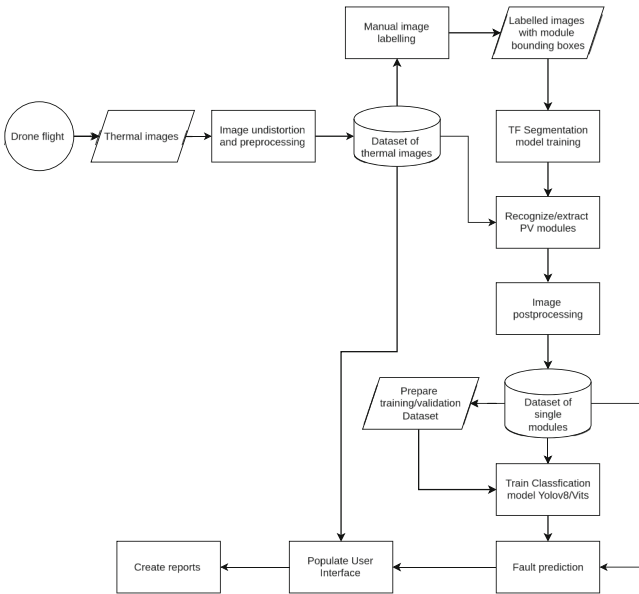
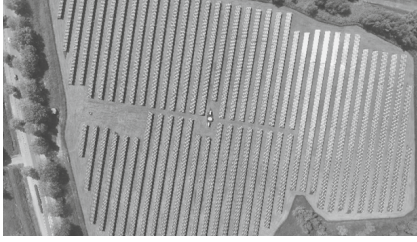


Fig. 2. Workflow diagram of the system

3.1 Study Area and Data Collection

The study was conducted at a solar park in Wilhelmshaven, Germany, commissioned by Volksbank Wilhelmshaven in 2013, an aerial view of the park is shown in Fig. 3. The park covers 55,000m² and has an installed capacity of approximately 3 megawatts. It features 11,914 solar modules (CSG pvtech) connected to the local power grid. The site was selected for its considerable scale, accessibility, and the potential issues related to its age and other factors such as vegetation. This provided a rich dataset for analysis.



(a) Aerial view of the PV system inspection area.



(b) DJI Mavic 3T drone used for thermal imaging and data collection in PV system inspections

Fig. 3. Study area and the used Mavic 3T drone

3.2 Drone and Image Acquisition

Automated drone flights offer significant advantages over traditional inspection methods. The drone inspection approach markedly enhances safety by keeping inspectors on the ground, thus eliminating personal risk. Documentation is digitized, making the process more objective and repeatable than paper-based methods. Additionally, the potential for errors is greatly reduced, and digital record-keeping allows for better accountability of past inspections.

The Mavic 3T's thermal camera supports radiometric thermal video recording, which captures temperature data for each pixel, providing precise thermal analysis during inspections of PV parks. Our chosen drone for this mission is therefore the robust and automated Mavic 3T depicted in Fig. 3.

Employing this drone, which features an advanced mission control system, we conducted an aerial survey of the study area. The resulting dataset, encompassing numerous high-resolution photographs, is described in Table 2

Table 2. Acquisition details of collected thermal imagery

Attribute	Description
Total Images	2,028 aerial images
Imaging Technique	Thermal imaging
Camera Specifications	DJI Mavic 3T, 640×512 pixel resolution, thermal imaging camera
Field of View	61° diagonal
Accuracy	$\pm 2^\circ\text{C}$ or $\pm 2\%$, whichever is higher
Image Format	JPEG, with embedded metadata (timestamp, GPS coordinates, ambient temperature)
Data Collection Period	Single session, 70 min

3.3 Image Distortion Correction and Preprocessing

Thermal images were initially processed to correct lens distortions using the undistort functions of OpenCV [18], which rectify radial and tangential distortions. Our developed UI application for this step contains sliders to adjust the correction parameters, while displaying below them a comparison of images before and after barrel distortion correction. The parameters were specifically calibrated for the Mavic 3T thermal camera and are automatically applied to all photos taken with it. However, this application also allows users to experiment with different parameters for other cameras.

After correction, the images underwent standard preprocessing steps, including cropping, resizing, and normalization, to prepare them for segmentation model training.

3.4 Segmentation Model Training and Extraction

We performed data annotation (labeling) using the Computer Vision Annotation Tool (CVAT [9]), marking the precise locations of the PV modules within the images, resulting in a dataset of labeled images.

After that, the TensorFlow Object Detection API, equipped with the pre-trained model “EfficientDet-D0” [28], was utilized to train a model capable of identifying individual PV modules within the thermal images. Following that, we had manually selected and curated a dataset of images of single modules as listed in Table 3 for subsequent training of the classification models. All images are in JPEG format and have a resolution of 64×96 pixels.

Table 3. Number of Single-Module Images per Class after Segmentation

Severe Physical Damage	Cell-Level Defects	Minor Anomalies	Systemic Issues	No Defect	Total
300	310	309	311	315	1545

3.5 Classification Models Training

Two advanced machine learning models were employed for defect classification:

YOLOv8: The model specifically used was the YOLOv8m-cls. This medium-sized model offers a balanced combination of training time savings and performance, which made it the optimal choice for our implementation. The YOLOv8 algorithm was trained to classify the types of defects observed in PV panels. This adaptation allowed for the precise differentiation of various forms of degradation.

ViTs: these models approach image analysis through the lens of transformer models, which are traditionally used in natural language processing. By treating

image patches as sequences, ViTs provide a unique mechanism to capture contextual relationships in data, which promise to enhance the model’s ability to focus on subtle features indicative of early-stage defects not readily apparent to other algorithms. The specific variant used was “ViT-B/16,” a mid-sized Vision Transformer that processes images by treating 16×16 pixel patches as sequence elements.

Training of these models was conducted on Google Colab using a Tesla T4 GPU. The models were trained exclusively using a carefully curated, manually annotated dataset, which consists of 1,545 thermal images representing various types of defects in photovoltaic (PV) modules. In order to ensure the robustness and accuracy of the study, no external datasets were used.

4 Results

In this section, we evaluate our proposed models and compare the classification models to each other based on evaluation metrics and confusion matrices.

4.1 Evaluation Indicators

To validate the performance of all used models, precision (P), recall (R), and F1-score, as suggested in [30], were chosen as experimental evaluation metrics. These were calculated as follows:

$$P = \frac{TP}{TP + FP} \cdot 100\% \quad R = \frac{TP}{TP + FN} \cdot 100\% \quad F1 = \frac{2 \cdot P \cdot R}{P + R} \cdot 100\%$$

where **TP** (true positives) denotes the number of targets detected correctly, **FP** (false positives) denotes the number of backgrounds detected as targets, and **FN** (false negatives) denotes the number of targets detected as backgrounds.

4.2 Performance of the Models

Performance of the Segmentation Model. In addition to the aforementioned general metrics, the performance of our segmentation model was evaluated based on an additional one, namely Intersection over Union (IoU).

In image recognition, the IoU value is a critical indicator for the accuracy of object recognition. It measures the correspondence between the predicted bounding boxes and the actual frames defined as ground truth. The IoU value divides the area of overlap between the two bounding boxes by the area of their union [26].

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}} \quad (1)$$

A predicted frame is typically regarded as a true positive (TP) if its IoU value exceeds a predefined threshold, which in our work is 0.7.

The segmentation model had achieved the following values for the metrics:

- **Precision:** With an accuracy of approximately 96.8%, most of our model predictions are correct.
- **Recognition (Recall):** Our model has achieved a complete detection rate of 100%. It did not miss a single true bounding box.
- **F1-score:** An almost perfect score of approximately 98.4% indicates a good balance between precision and recall and points to the high functionality of the model.

The segmentation model demonstrates high precision, recall, and F1-score, making it suitable for accurate and reliable detection.

Performance of the YOLOv8 Classification Model. Various defect types were successfully identified during defect classification using the trained YOLOv8 model as detailed in Table 4. The metrics were calculated using the test dataset to evaluate the models’ generalization capabilities. The model’s effectiveness was confirmed by its alignment with manually classified defects. Although Table 3 shows a balanced category distribution, some defect types, like cell-level defects, were sometimes misclassified as “no defect”. This suggests the model struggles to identify subtle variations of specific defects, despite having comprehensive data for each category. Additional feature engineering could further enhance the model’s ability.

Table 4. Performance evaluation of defect classification using the YOLOv8 model

Class	Precision (%)	Recall (%)	F1-score (%)
Class 1 (Cell-Level Defects)	86.41	89.90	88.12
Class 2 (Minor Anomalies)	90.00	91.84	90.91
Class 3 (No Defect)	89.54	87.00	88.25
Class 4 (Severe Physical)	93.81	93.81	93.81
Class 5 (Systemic Issues)	100.00	96.91	98.43

The confusion matrix in Fig. 4 illustrates the efficacy of YOLOv8 in categorizing diverse defects in PV systems. The model demonstrates high accuracy in identifying instances of “No Defect” and “Systemic Issues”, indicating its robust predictive capabilities in these categories. The confusion between “Cell-Level Defects” and “No Defect” indicates potential areas for refinement to enhance the model’s precision further.

Performance of the Vision Transformers Classification Model. The ViTs model was also successful in identifying the various classes, as seen in

Table 5. Based on the evaluation, the model has demonstrated more consistent performance in terms of validation metrics.

The confusion matrix in Fig. 4 demonstrates the high accuracy of the ViTs model in defect classification, particularly for “Systemic Issues” and “Minor Anomalies”, with an accuracy rate exceeding 90%. It is evident that there is some confusion between “Cell-Level Defects” on the one hand, and both “Minor Anomalies” and “No Defects” on the other, which indicates areas for improvement. In conclusion, the model displays considerable potential for generalization and reliability.

Table 5. Performance evaluation of defect classification using the ViTs model

Class	Precision (%)	Recall (%)	F1-score (%)
Class 1 (Cell-Level Defects)	88.00	85.00	87.00
Class 2 (Minor Anomalies)	89.00	87.00	88.00
Class 3 (No Defect)	91.00	93.00	92.00
Class 4 (Severe Physical)	92.00	90.00	91.00
Class 5 (Systemic Issues)	93.00	92.00	92.00

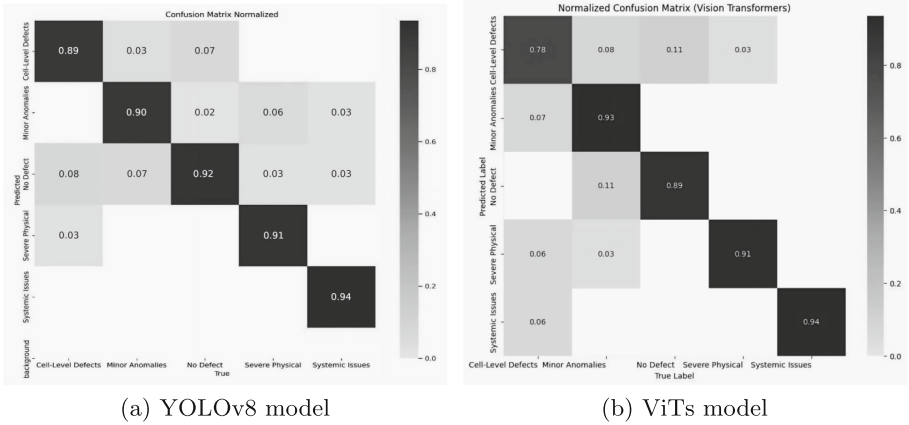


Fig. 4. Confusion matrix of the classification models

Comparative Analysis of Results. Although the primary aim of this paper is to compare ViTs with YOLOv8 models, we can also benchmark our results against those from related studies.

In this regard, we observe the following: Prabhakaran et al. [25] reported an F1-score of 86% using AlexNet, while our YOLOv8 and ViTs models achieved higher F1-scores of 93.81% for “Severe Physical Defects” and 98.43% for “Systemic Issues”. Fiorese et al. [14] achieved an F1-score of 69% and an IoU of 57.3% with Deeplabv3, compared to our TensorFlow model’s IoU of 70%, and the F1-scores mentioned above. They had targeted different defect classes however and used a different imaging technology. Bommers et al. [6] reported high precision and recall using ResNet-34 CNN, but used a different evaluation metric, namely AUROC due to unbalanced dataset. Our models showed similarly high precision and recall, demonstrating strong performance.

However, overfitting must be considered, as our dataset consists of modules from the same manufacturer and model. The performance of our models on other types of modules, despite the technological similarities, is not assured.

5 Discussion and Conclusion

All three developed models demonstrated excellent performance in detecting and classifying defects in PV modules.

Both YOLOv8 and ViTs models exhibited high accuracy in the classification of defects in PV modules. YOLOv8 exhibited slightly higher precision in certain defect categories, whereas ViTs demonstrated more consistent performance across all categories. In light of the comparable performance metrics, the selection of the optimal model should be based on additional considerations, such as the availability of computational resources and the specific requirements of the intended application. In scenarios where lower computational demand is required, YOLOv8 may be the preferred option. Conversely, for applications where consistent performance across various defect types is of paramount importance, ViTs could be the better choice.

We argue that cropping individual modules before applying the detection algorithm is more effective for identifying faults within single modules. However, considering the overall series can be beneficial for detecting other types of faults. For instance, when an entire module is offline or connected in reverse. Such significant faults can occur during installation or manufacturing and may not be detectable by observing individual modules in separation.

Future work includes optimizing rare defect detection in the YOLOv8 model through improved dataset augmentation, exploring hybrid models combining ViTs and YOLOv8 for enhanced classification, utilizing the RGB photos captured by the drone for further classification refinement, and integrating real-time defect detection and classification into the user interface.

Our results demonstrate that combining our models with drone-assisted infrared thermography significantly improves the accuracy and efficiency of PV module inspections over manual inspection, offering a comprehensive automated solution for defect detection and classification.

The dataset [11] and the code [10] are available under open source licenses.

References

1. Abadi, M., et al.: TensorFlow: large-scale machine learning on heterogeneous systems (2015)
2. Acikgoz, H.: An automatic detection model for cracks in photovoltaic cells based on electroluminescence imaging using improved YOLOv7. *SIViP* **18**(1), 625–635 (2024)
3. Ahmed, W., Hanif, A., Kallu, K.D., Kouzani, A.Z., Ali, M.U., Zafar, A.: Photovoltaic panels classification using isolated and transfer learned deep neural models using infrared thermographic images. *Sensors* **21**, 5668 (2021). <https://doi.org/10.3390/s21165668>
4. Ali, M.U., Khan, H.F., Masud, M., Kallu, K.D., Zafar, A.: A machine learning framework to identify the hotspot in photovoltaic module using infrared thermography. *Sol. Energy* **208**, 643–651 (2020)
5. Bo, R., et al.: Machine learning applications in health monitoring of renewable energy systems (2024). <https://doi.org/10.1016/j.rser.2023.114039>
6. Bommès, L., et al.: Anomaly detection in IR images of PV modules using supervised contrastive learning. *Progress Photovoltaics* (2022). <https://doi.org/10.1002/pip.3518>
7. Bommès, L., Pickel, T., Buerhop-Lutz, C., Hauch, J., Brabec, C., Peters, I.M.: Computer vision tool for detection, mapping, and fault classification of photovoltaics modules in aerial IR videos. *Prog. Photovoltaics Res. Appl.* **29**, 1236–1251 (2021). <https://doi.org/10.1002/pip.3448>
8. Chen, Q., et al.: Remote sensing of photovoltaic scenarios: techniques, applications and future directions (2023). <https://doi.org/10.1016/j.apenergy.2022.120579>
9. CVAT.ai Corporation: Computer Vision Annotation Tool (CVAT), November 2023. <https://github.com/cvat-ai/cvat>
10. Debich, T.: Analyse von drohnen pv-thermobildern (2024). <https://github.com/TarDeb/MA>
11. Debich, T.: Photovoltaic defects thermal images dataset (2024). <https://www.kaggle.com/datasets/tarekdebich/data-class-pv>
12. DIN: DIN EN 62446-1 - 2019-04 (2019). <https://www.beuth.de/en/standard/din-en-62446-1/299821500>
13. Dosovitskiy, A., et al.: An image is worth 16x16 words: transformers for image recognition at scale. *CoRR abs/2010.11929* (2020). <https://arxiv.org/abs/2010.11929>
14. Fiorese, J., et al.: Automated defect detection and localization in photovoltaic cells using semantic segmentation of electroluminescence images. *IEEE J. Photovoltaics* (2022). <https://doi.org/10.1109/jphotov.2021.3131059>
15. IEA: World Energy Outlook 2021 - Analysis - IEA (2021). <https://www.iea.org/reports/world-energy-outlook-2021>
16. IEC: IEC TS 62446-3:2017 — IEC Webstore (2017). <https://webstore.iec.ch/publication/28628>
17. Ishak, N.H.B., Isa, I.S.B., Osman, M.K.B., Daud, K., Jadin, M.S.B.: Hotspot detection of solar photovoltaic system: a perspective from image processing. In: 2023 IEEE 3rd International Conference in Power Engineering Applications (ICPEA), pp. 263–267. IEEE (2023)
18. Itseez: Open source computer vision (2015). <https://github.com/itseez/opencv>
19. Jocher, G., Chaurasia, A., Qiu, J.: Ultralytics YOLO, January 2023. <https://github.com/ultralytics/ultralytics>

20. Li, B., Delpha, C., Diallo, D., Migan-Dubois, A.: Application of artificial neural networks to photovoltaic fault detection and diagnosis: a review. *Renew. Sustain. Energy Rev.* (2020). <https://doi.org/10.1016/j.rser.2020.110512>
21. Li, G., Wang, F., Feng, F., Wei, B.: Hot spot detection of photovoltaic module based on distributed fiber bragg grating sensor. *Sensors* **22**(13), 4951 (2022). <https://doi.org/10.3390/s22134951>. <https://www.mdpi.com/1424-8220/22/13/4951>
22. Li, X., Li, W., Yang, Q., Yan, W., Zomaya, A.Y.: Building an online defect detection system for large-scale photovoltaic plants. *BuildSys@SenSys* (2019). <https://doi.org/10.1145/3360322.3360835>
23. de Oliveira, A.K.V., Aghaei, M., R  ther, R.: Automatic inspection of photovoltaic power plants using aerial infrared thermography: a review. *Energies* **15**, 2055 (2022). <https://doi.org/10.3390/en15062055>
24. Oviedo, E.H.S., Trav  -Massuy  s, L., Subias, A., Pavlov, M., Alonso, C.: Fault diagnosis of photovoltaic systems using artificial intelligence: a bibliometric approach (2023). <https://doi.org/10.1016/j.heliyon.2023.e21491>
25. Prabhakaran, S., Uthra, R.A., Preetharoselyn, J.: Comprehensive analysis of defect detection through image processing and machine learning for photovoltaic panels (2023). https://doi.org/10.1007/978-981-19-7169-3_23
26. Rezatofighi, S.H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I.D., Savarese, S.: Generalized intersection over union: a metric and a loss for bounding box regression. *CoRR* abs/1902.09630 (2019). <http://arxiv.org/abs/1902.09630>
27. SolarPower, E.: EU market outlook for solar power 2021-2025 - SolarPower Europe (2021). <https://www.solarpowereurope.org/insights/market-outlooks/eu-market-outlook-for-solar-power-2021-2025>
28. Tan, M., Pang, R., Le, Q.V.: EfficientDet: scalable and efficient object detection (2020)
29. Thobe, I., Lehr, U., Edler, D.: Betrieb und Wartung von Anlagen zur Nutzung von erneuerbaren Energien - Kosten und Struktur in der Literatur. GWS Discussion Paper Series (2015). <https://ideas.repec.org/p/gws/dpaper/15-4.html>. Number: 15-4 Publisher: GWS - Institute of Economic Structures Research
30. Zhang, L., Wang, M., Liu, K., Xiao, M., Wen, Z., Man, J.: An automatic fault detection method of freight train images based on BD-YOLO. *IEEE Access* **10**, 39613–39626 (2022). <https://doi.org/10.1109/ACCESS.2022.3165835>
31. Zhang, X., et al.: Inspection and classification system of photovoltaic module defects based on UAV and thermal imaging. In: 2022 IEEE The 7th International Conference on Power and Renewable Energy, pp. 905–909. IEEE (2022). <https://doi.org/10.1109/ICPRE55555.2022.9960506>

Evaluation of AI Systems



Evaluating Algorithms for Missing Value Imputation in Real Battery Data

Dauda Nanman Shen¹ , Anton Herman Basson¹  , and Jacomine Grobler² 

¹ Department of Mechanical and Mechatronic Engineering, Stellenbosch University, Stellenbosch, South Africa
ahb@sun.ac.za

² Department of Industrial Engineering, Stellenbosch University, Stellenbosch, South Africa

Abstract. With the growing number of data-driven services, effective methodologies to identify errors and rectify missing values within data are required to ensure data quality. This paper examines the performance of three supervised prediction and two imputation techniques for missing value imputation on real, structured data from high-voltage batteries in the automotive industry. The cost of incorrect data can be very high in this application.

In this study, starting with an error-free data set, missing values are generated according to the missing-at-random mechanism. Thereafter, the decision tree, multilayer perceptron, k-nearest neighbour, and support vector machine algorithms are evaluated for classification and regression tasks. Further, the k-nearest neighbour imputer and the multiple imputation by chained equations algorithm are evaluated as imputation techniques. The performance of these algorithms is compared based on their reliability and error metrics for three categorical and seven continuous features.

The paper shows that features, where the standard deviation is less than the mean, are predicted more reliably than features with larger spreads. It is seen that more complex algorithms, i.e. support vector machine and multi-layer perceptron, perform better for more complex features, i.e. features with a higher cardinality and a wide range. The k-nearest neighbour algorithm emerges as the best performing, demonstrating efficient learning and generalisation across all features.

The approach used in this research identifies the strengths and weaknesses of various machine learning algorithms in handling imputation tasks. This research informs the selection and combination of methods for better error correction and data management, ultimately supporting more reliable and efficient operations. This paper has shown that machine learning algorithms can be used for missing value imputation and error correction, with k-nearest neighbour and multiple imputation by chained equations providing efficient and reliable estimates for missing values, with minimal training effort.

Keywords: Error Correction · Missing Value Imputation

1 Introduction

In large data systems, errors and inconsistencies significantly reduce data quality, impacting business decisions and regulatory compliance, particularly in automotive systems. Accurate and reliable storage of process and product information is essential. Consequently, there is a need for effective methodologies to identify and rectify errors, including handling missing values. The increased use of electric vehicles has further highlighted the importance of high-quality data. The European Union (EU) has mandated that by 2026, all industrial or electric vehicles in the EU with a capacity over 2 kWh will require a battery passport, including data from various sources. The battery passport underscores the necessity of managing data inconsistencies and missing values effectively.

This paper explores the use of artificial intelligence (AI) and machine learning (ML) methods for the correction of detected errors, here represented by synthetically generated missing values. The study evaluates classification, regression, and missing value imputation (MVI) methods based on predictive accuracy. Specifically, the paper assesses the reliability of supervised imputation and predictive algorithms on a dataset of high-voltage batteries (HVBs) for an electric vehicle.

The evaluated methods are intended for use with digital twins of motor vehicles, which are virtual representations of vehicles that facilitate real-time monitoring and performance optimisation. These digital twins are used throughout the vehicle lifecycle for, inter alia, product and material traceability [3]. Given the high volume of data involved in these digital twins, automated error detection and correction are essential. The findings are significant in the context of regulatory compliance and the transition to a climate-neutral economy, as mandated by the EU [11].

This paper is structured as follows: Section 2 summarises related research on the approaches to error correction and MVI for large data systems, missing value mechanisms, and the algorithms employed for this research. Section 3 describes the HVB dataset with its feature characteristics. Section 4 details the case study approach, including data preparation, feature modelling, and hyperparameter tuning. The experiment is evaluated in Sect. 5 and the paper is concluded in Sect. 6, including areas of future work.

2 Foundations and Related Work

The approaches to error correction are discussed in Sect. 2.1, the mechanisms that generate missing values in Sect. 2.2, and the employed algorithms in Sect. 2.3. The experimental setup (Sect. 5) is informed by the literature reviewed in this section.

2.1 Error Correction in Large Data Systems

Researchers have demonstrated the effectiveness of decision trees in imputing missing values generated by the missing-at-random mechanism that uses other features in the dataset [3, 14, 23]. Twala [23] found that model-based approaches, such as the expectation maximisation algorithm, performed best on 21 complete datasets from the UCI repository. Burgette and Reiter [3] applied the classification and regression tree (CART)

algorithm within the “multiple imputation by chained equations” (MICE) framework, offering a flexible and robust nonparametric approach for handling various data types and complex relationships, although it may create undesirable discontinuities at partition boundaries.

Random forests have also been found to provide reliable results in multiple imputation by aggregating decisions from multiple decision trees [16, 24]. Pantanowitz and Marwala [16] attributed the increased accuracy of random forests, compared to auto-associate neural networks and hybrid methods, to their robustness in handling large datasets and managing both continuous and categorical variables efficiently. Stekhoven and Bühlmann [22] introduced MissForest, which addresses the problem of imputing missing values in datasets with complex interactions and non-linear relationships.

Silva-Ramirez et al. [19] highlighted the effectiveness of multilayer perceptrons (MLPs) and support vector machines (SVMs) in imputing missing data and correcting erroneous records on 21 UCI datasets with both categorical and numerical features. They found that MLPs were particularly effective in improving decision-making processes [19, 20].

Jerez et al. [6] enhanced prognosis accuracy for early cancer relapse in a dataset of 3678 women by comparing MVI carried out by MLPs, self-organising maps (SOMs), and k-nearest neighbour (k-NN) against traditional statistical methods. ML methods outperformed statistical imputation methods and improved prognosis accuracy. Other neural network-based and hybrid methods have also been employed [12, 13, 18, 24].

Automated error correction in large data systems often relies on domain experts to annotate and verify detected anomalies and corrections to erroneous data [4]. Alwan et al. [7] categorise data quality management methods into three primary categories: mathematical models, data mining methods, and technical solutions. Data mining methods enable the auto-discovery of knowledge from large volumes of data using techniques such as anomaly detection, predictive analysis, and clustering. Technical solutions extend these methods with technologies for data profiling, cleansing, validation, and integration [12–14, 18, 23].

2.2 Missing Value Mechanisms

Rubin and Little [10] defined missingness as the occurrence of missing values within a dataset and that missingness can occur in different ways and can be related to various mechanisms. There are three commonly identified missing value mechanisms: missing-completely-at-random (MCAR), missing-at-random (MAR), and not-missing-at-random (NMAR) [10, 22].

MCAR occurs when the probability of an instance having a missing value for an attribute does not depend on either the known values or the missing data [22]. MAR occurs when the probability of an instance having a missing value for an attribute may depend on the value of that attribute. In other words, MAR occurs when the distribution of an instance having missing values for an attribute depends on the observed data but does not depend on the missing data [8]. NMAR occurs when the probability of an instance having a missing value for an attribute may depend on the value of that attribute [22].

Depending on the mechanism that has generated the missing value, which should be confirmed by a domain expert, a missing value may be treated as an error that can be corrected [8]. Predictions can be carried out reliably when the missingness depends on the observed data, i.e. missingness generated by the MAR mechanism [22], although research has also been conducted on MCAR data [21]. Predictive models can be trained on the observed data to estimate or impute missing values. Similarly, if there is no informative value in an error, an error can be converted to a missing value generated by a MAR mechanism, and a reliable prediction, or correction, can be made.

2.3 Algorithms for Missing Value Imputation

The following section outlines the ML algorithms employed in this paper. The algorithms were selected to represent the diverse range of ML algorithms that literature has shown to perform well.

K-Nearest Neighbour

The non-parametric k-NN algorithm can be inexpensive to train, depending on the size of the training set, the distance metric used to measure similarity, and how the data is normalised. Outliers have a limited effect on model performance as the closest neighbours will be near the majority of the data points if a sufficiently large ‘k’ value is selected. The k-NN algorithm is known as a ‘lazy learner’ due to the algorithm only abstracting from the data when a prediction is required, increasing inference time in large datasets [17]. The learning strategy of k-NN is useful and widely applied for MVI, outperforming more complex ML algorithms [7, 15, 24].

Decision Tree

Decision trees are the most common information-based learning algorithm. Decision trees have the following advantages: they are easily interpretable and the algorithm easily handles categorical and continuous features, regardless of normalisation. This algorithm can be inexpensive to train, depending on the size of the dataset, and is efficient due to the tree traversal algorithm [16]. However, decision trees are prone to overfitting and creating leaf nodes with one or two instances [17]. Also, making decisions with a low sample size (the number of instances in the leaf node) results in unreliable predictions.

For error correction through MVI, the non-parametric nature of the decision tree algorithm can be exploited for a range of diverse features. As there is a limit on the size of a leaf node, the algorithm can provide unreliable values for features with a high cardinality [17]. The interpretability of the algorithm is valuable for understanding why certain missing values have been imputed, providing additional rules to domain experts [7].

Support Vector Machines

SVMs use a kernel-based approach and an error-driven learning algorithm. Initially, SVMs were used to carry out binary classification, but have been extended to handle multi-class classification and regression tasks [20]. SVMs perform well on high-dimensional data and on smaller datasets with minimal noise. The algorithm is relatively efficient in memory [20], but is a slow learner and often results in longer training times than the k-NN and decision tree algorithms [17].

Although SVMs cannot handle missing values, the MVI estimator can still be used to achieve comparable error correction and sometimes greater accuracy than other MVI algorithms. SVMs have been applied to missing value problems in the literature [20] and can be applied to impute values that are modelled by complex relationships between the target and descriptive features. SVMs are not easily interpretable and require more considerations during training to ensure the algorithm does not overfit due to the presence of noise [5, 17, 20].

Multi-layer Perceptron

The training of an MLP (and other neural networks) is intensive and requires careful data preparation and the selection of the number of hidden layer neurons, the parameters of the model, and the optimisation algorithm used to train the model weights. The algorithm is sensitive to missing values and the presence of noise, requiring careful selection of the training set [21].

MLPs and other neural networks are widely applied to the missing value problem. Similar to SVMs, the algorithm cannot handle missing values directly but can still provide reliable predictions. The ‘black box’ nature of MLPs does not allow for interpretability. The algorithm is not sensitive to the nature of features, nor the size of the dataset, and provides reliable and efficient predictions, often outperforming other ML algorithms [9, 19]. MLPs are often applied with other neural networks or ML algorithms to increase performance and form generative networks [12].

3 Description of Dataset

A battery passport is a document that describes the characteristics, battery type, and production information for an HVB. Some of the features typically included in a battery passport are described in Table 1.

An HVB consists of a variable number of modules, with each type of module having the same number of cells (18, 22, 40, or 50). The module information is largely the same for each module, with the major differences being the production dates, the material change indices, and the part descriptions. The material weights of the different modules vary significantly, indicating that modules do not have the same number of cells. However, this case study focuses on data from the higher-level battery packs and not on the module or cell component level.

A statistical summary of the data is presented in Table 2. In this table, the level of cardinality is considered to be low if there are fewer than 30 distinct values, high if there are more than 100 distinct values, or otherwise moderate.

4 Case Study Modelling

4.1 Data Preparation

The original dataset consists of 8049 HVBS, with their associated modules and cells. There are 6405 complete records in the HVB dataset with 6113 unique battery IDs, indicating the presence of duplicate records. The capacity, energy, and charges fields are

Table 1. HVB Feature Description.

Field	Description
HVB_UID	A unique HVB identifier allowing for the unambiguous identification of each individual battery and each corresponding battery passport
Material Part Number	The part number of the associated component. Not unique across HVB instances
Material Change Index	An index that distinguishes between different versions or states of the material
Material Part Description	A business logic-based description of the battery pack. Includes the model code, parts of the serial number and production date
Production Date	The manufacturing date of the HVB
Material Weight	The mass of the HVB (gram)
Model Code	A five-letter business code that describes the model/series of the HVB
Mean Aging Capacity	The mean decrease, over the time of usage, in the amount of charge (Ah) that the battery can deliver at the rated voltage, expressed as a percentage of the original rated capacity
Minimum Aging Capacity	The minimum decrease, over time of usage, in the amount of charge (Ah) that a battery can deliver at the rated voltage, expressed as a percentage of the original rated capacity
Mean Capacity	The mean value for the current battery capacity (Ah)
Minimum Capacity	The minimum value for the current battery capacity (Ah)
Energy Throughput	The energy throughput during the HVB life cycle (kWh)
Capacity Throughput	The capacity throughput during the HVB life cycle (Ah)
Charges	The number of times the HVB has been charged
Full Charges	The number of times the HVB has been fully charged
Voltage	The value for the current battery voltage (V)

empty for 1644 of the 8049 batteries (20.42%), which confirms that MVI is beneficial to the battery passport use case. Data instances with multiple missing battery characteristics cannot be accurately determined from the other fields. Associatively, determining the type of battery (Model Code) without the energy and capacity fields is nearly impossible. Outliers were identified in the ‘Material Weight’ feature, with 161 records having a material weight of 1g, which is not a realistic weight for an HVB.

After removing records with missing values, rows with duplicate battery HVB_UIDs, and rows with outliers in the ‘Material Weight’ feature, the dataset consisted of 6113 records with eight different model codes. Text-based features and primary keys were excluded from the dataset. Features with high correlations (>0.97) were also removed, leaving a single feature from each correlated pair. The ‘Production Date’ feature was pre-processed into ‘Production Date week’ and ‘Production Date year’ features. After

Table 2. Statistical Summary of the HVB Dataset.

Field	Data Type	Format / Range	Normalised Mean (Standard Deviation)	Cardinality
HVB_UID	String, Unique	31 characters	N/A	High
Material Part Number	Categorical Numerical	8 digits	1.0 (± 0.0005)	Moderate
Material Change Index	Categorical Numerical Feature	1 digit	1.0 (± 0.14)	Low
Material Part Description	Structured Text	28 characters	N/A	Moderate
Production Date	Date-time	YYYY-MM-DD	N/A	High
Material Weight	Categorical Numerical	6 digits	1.0 (± 0.35)	Moderate
Model Code	Categorical Text	5 characters	N/A	Low
Mean Aging Capacity	Continuous	decimal	1.0 (± 0.02)	High
Minimum Aging Capacity	Continuous	decimal	1.0 (± 0.02)	High
Mean Capacity	Continuous	decimal	1.0 (± 0.34)	High
Minimum Capacity	Continuous	decimal	1.0 (± 0.34)	High
Energy Throughput	Continuous	decimal	1.0 (± 1.2)	High
Capacity Throughput	Continuous	decimal	1.0 (± 1.2)	High
Charges	Continuous	number	1.0 (± 1.8)	High
Full Charges	Number	number	1.0 (± 3.9)	High
Voltage	Number	decimal	1.0 (± 0.10)	High

these cleaning steps, all numerical features were normalised, and missing values were randomly generated in 1% of the target feature. All algorithms were tested with the same pre-processed data, with missing values in the target feature.

4.2 Modelling

Categorical Features

The dataset includes four categorical features: ‘*Model Code*’, ‘*Material Weight*’, ‘*Material Part Number*’, and ‘*Material Change Index*’. ‘*Model Code*’ is treated as a multi-class classification problem. One-hot encoding of ‘*Material Weight*’ and ‘*Material Part Number*’ resulted in poor model performance due to “the curse of dimensionality”; therefore,

these features were normalised and treated as continuous. Predictions for the ‘Material Change Index’ were not conducted due to a strong majority class in the data.

Continuous Features

The dataset has seven continuous features and two high-cardinality categorical features treated as continuous features. Predictions focused on features with high cardinality and wide ranges, such as ‘Energy Throughput’ and ‘Capacity Throughput’. These dynamic features are correlated with temporal features like ‘Charges’ and ‘Full Charges’.

Temporal Features

MVI was not applied to temporal features such as ‘Production Date’, ‘Charges’ and ‘Full Charges’, due to the non-compliance with the MAR assumption and the potential for large prediction errors. These features depend on previously recorded dates and charge metrics.

4.3 Hyperparameter Tuning

A ten-fold cross-validation grid search was conducted for each algorithm-feature pair. The parameter grid was set wide enough to allow for the variations in each feature, as shown in Fig. 1. Grid searches were conducted using the complete, pre-processed data. Hyperparameters were determined in the best-case scenario, with no missing values or errors. The grid search results can be found at [25].

k-NN Imputer	Decision Tree
Neighbours: 3,5,7,9	Criterion: Squared/Absolute error,Poisson
Distance: Nan-Euclidean	Max Depth: 5,10,15,20,25
Weights: Uniform, Distance	Splitter: Best, Random
	Min Samples Leaf: 2,5,10
MLP	SVM
Hidden Layer Size: 40, 45, ..., 85	Kernel: Linear, Poly, RBF, Sigmoid
Activation: ReLU, tanh	C: 1,2,5,8,10
Learning Rate: 0.01, 0.05, 0.1	Epsilon: 0.01, 0.05, 0.1, 1.0
Solver: Adam, SGD	Degree: 1,2,3,4

Fig. 1. Hyperparameter grid boundaries.

5 Experiment Evaluation

The experimental setup, which aims to represent the steps that a data scientist would take to set up an ML pipeline, is outlined in Sect. 5.1. The evaluation of the results is presented in Sect. 5.2.

5.1 Experimental Setup

The data is prepared as described in Sect. 4.1.

The datasets are split into training (70%) and holdout test sets (30%). The training set is further split for training and testing in a training loop, while the holdout test set is used for ten-fold cross-validation, simulating experimentation, and deployment phases. Accuracy and error metrics are assigned as training and validation metrics.

Missing values are introduced into the target feature of the training and test sets at randomly selected indices. The training loop iterates 30 times, creating different training and test sets for each algorithm to avoid bias from a single missing value pattern. The prepared data, excluding the target feature, serves as input for the prediction algorithms (MLP, SVM, and decision tree), while the incomplete target feature is input for the imputation algorithms (k-NN, MICE). Predictive algorithms are trained on complete data and tested on incomplete data. All algorithms are validated using models from the training loop, and a Mann-Whitney U test is performed in the validation loop. Training and inference results are recorded.

Classification accuracy, which is the ratio of correct predictions to total predictions, evaluates categorical features. The classification accuracy is a commonly used metric for classification tasks and is also known as the percentage of correct predictions or the correct rate [6, 18]. For numerical features, the normalised mean-square error (NMSE) and root-mean-square error (RMSE), are often preferred due to their sensitivity to outliers [18]. However, in this study, the range-normalised RMSE (NRMSE) and the root-mean-square percentage error (RMSPE) are used to compare algorithm performance across different features. The NRMSE is relevant where the target feature's values are in a narrow range, while the RMSPE is more appropriate when the values span a wide range, and the percentage error is more important than the absolute value of the error.

A post-processing step reverts scaled features to their original format (not shown in this paper, for confidentiality reasons) and decodes class-encoded target features to calculate RMSPE accurately. The results are documented for record-keeping.

5.2 Experimental Results

The experiment was performed for 10 features and the detailed results can be found at [25]. The training results were comparable to the cross-validation results, showing an increased performance and lower error values in the validation loop, which has a smaller sample size. Figure 2 illustrates the error per feature and Table 3 summarises the classification performance by giving the average accuracy of each algorithm, the standard deviation, and the prediction times.

The particularly high performance of the algorithms indicates that, for classification tasks in this dataset, the algorithms are able to reliably distinguish different battery classes based on their characteristics (the observed features). The decision tree algorithm outperforms other algorithms with a near 100% accuracy and a low prediction time attributed to the efficiency of the tree traversal algorithm. Rule-based imputation could probably provide similar accuracies but would require significant time from a domain expert.

Table 3. Average classification accuracy for the ‘Model Code’ feature.

Algorithm	Average Accuracy (%)	Standard Deviation	Time
MLP	97.61	0.038	50 s
Decision Tree	100.00	0.00	1.0 s
SVM	98.26	3.31	2.9 s
SVM (One vs Rest)	99.24	2.31	11.7 s
MICE (Decision Tree)	99.41	2.22	21.8 s
kNN	98.61	3.11	1.0 s

The use of NRMSE and RMSPE as evaluation metrics proved successful for the regressed features. The normalised evaluation metric allows the algorithm performance to be compared across features, showing the highest error in the ‘Voltage’ feature and the lowest error in the ‘Material Weight’ feature. The highest RMSPE error is found in the ‘Capacity Throughput’ and ‘Energy Throughput’ features, with RMSPEs of 26% and 64% respectively, for the k-NN algorithm. These features are not reliably predicted by any algorithm.

Modelling the categorical ‘Material Part Number’ and ‘Material Weight’ features as continuous features proved effective, resulting in low RMSPE errors. The capacity-related features are predicted quite well, which can be attributed to their relatively low cardinality and range. The reliability of imputation methods is attributed to the nature of the imputed features, the observed features, and the selected MVI algorithms. The k-NN algorithm emerged as the best performing, demonstrating efficient learning and generalisation. Its performance was further examined using the MICE algorithm.

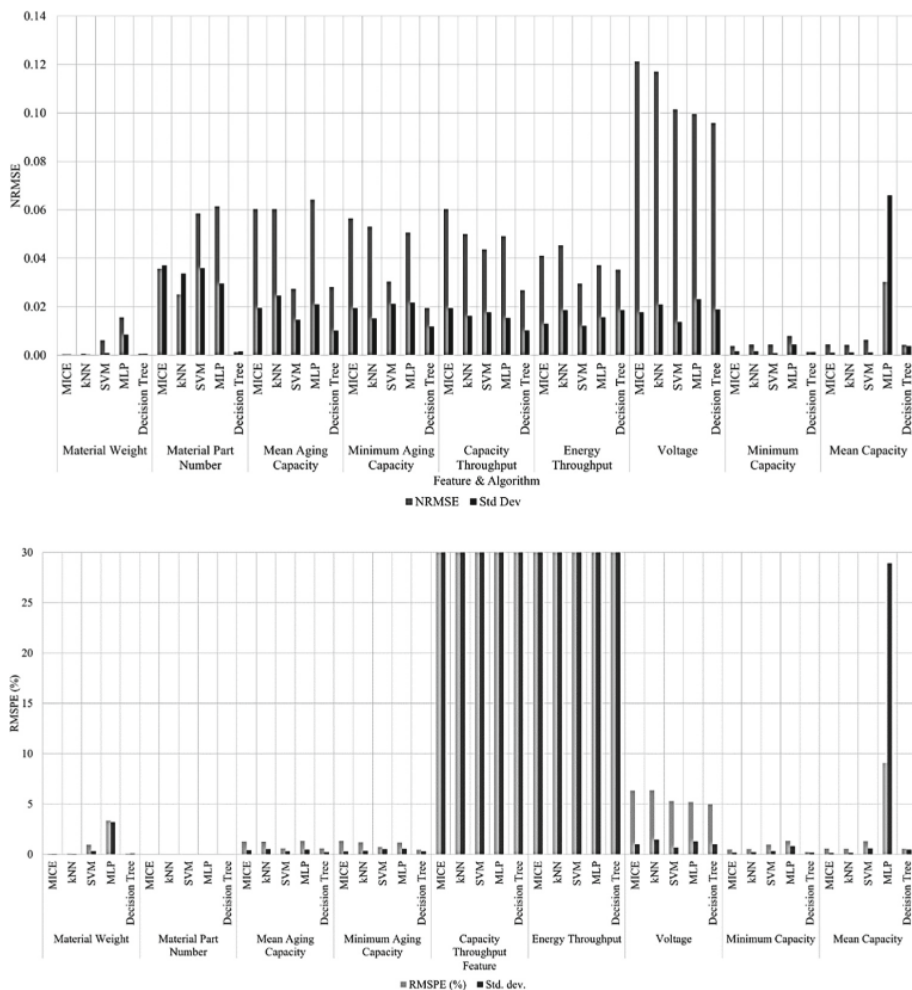


Fig. 2. NRMSE values (top) and RMSPE (bottom) values for the continuous features in the HVB dataset.

6 Conclusion

This paper explores the imputation of categorical and numerical errors in a real HVB dataset with 6113 records, towards a methodology for enhancing data quality in diverse datasets. Among the ML algorithms evaluated, k-NN performs best, followed by decision trees, while MLPs and SVMs perform relatively poorly in this experimental setup. MICE is effective as an MVI tool, offering results comparable to or better than the standard k-NN imputer. The best results are seen in features with low cardinality and limited range, while the ‘Voltage’, ‘Capacity Throughput’, and ‘Energy Throughput’ features present the highest error. An ensemble of ML algorithms and statistical techniques is necessary for reliable imputation during deployment [2].

The paper contributes a real-world assessment of popular MVI algorithms, including the strengths and weaknesses of a range of commonly used algorithms. This research also contributes to the automotive industry practice by aiding the selection of methods for error correction and data management. Thereby, the paper contributes to the broader field of data science by demonstrating the practical potential of AI and ML for data cleaning processes.

Future work should explore additional data pre-processing techniques such as binning continuous features and applying principal component analysis to address dimensionality issues. These techniques may enhance prediction accuracy and reliability, supporting the integration of ML models into automated pipelines for error correction in large datasets. Additionally, further research into the effect of missing rates and non-monotone missingness is required.

References

1. Abedjan, Z., et al.: Detecting data errors: where are we and what needs to be done? Proc. VLDB Endowment **9**(12), 993–1004 (2016). <https://doi.org/10.14778/2994509.2994518>
2. Alwan, A.A., Ciupala, M.A., Brimicombe, A.J., Ghorashi, S.A., Baravalle, A., Falcarin, P.: Data quality challenges in large-scale cyber-physical systems: a systematic review. Inf. Syst. **105**, 101951 (2022). <https://doi.org/10.1016/j.is.2021.101951>
3. De Bruyn, C.: BMW South Africa improves materials traceability through digital twinning. <https://www.engineeringnews.co.za/article/bmw-south-africa-improves-materials-traceability-through-digital-twinning-2023-11-06>. Accessed 3 June 2024
4. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection. ACM Comput. Surv. **41**(3), 1–58 (2009). <https://doi.org/10.1145/1541880.1541882>
5. Feng, H.H., Liao, M.Y., Chen, G.S., Yang, B.R., Chen, Y.M.: SVM and reduction-based two algorithms for examining and eliminating mistakes in inconsistent examples. In: Proceedings of 2004 International Conference on Machine Learning and Cybernetics, vol. 4, pp. 2189–2192 (2004). <https://doi.org/10.1109/ICMLC.2004.1382161>
6. Handelman, G.S., et al.: Peering into the black box of artificial intelligence: evaluation metrics of machine learning methods. Am. J. Roentgenol. **212**(1), 38–43 (2019). https://doi.org/10.2214/AJR.18.20224/ASSET/IMAGES/LARGE/01_18_20224_07B.JPEG
7. Jerez, J.M., et al.: Missing data imputation using statistical and machine learning methods in a real breast cancer problem. Artif. Intell. Med. **50**(2), 105–115 (2010). <https://doi.org/10.1016/J.ARTMED.2010.05.002>
8. Lin, W.C., Tsai, C.F.: Missing value imputation: a review and analysis of the literature (2006–2017). Artif. Intell. Rev. **53**(2), 1487–1509 (2020). <https://doi.org/10.1007/S10462-019-09709-4/FIGURES/8>
9. Lin, W.C., Tsai, C.F., Zhong, J.R.: Deep learning for missing value imputation of continuous data and the effect of data discretization. Knowl. Based Syst. **239**, 108079 (2022). <https://doi.org/10.1016/J.KNOSYS.2021.108079>
10. Little, R.J.A., Rubin, D.B.: Introduction. In: Statistical Analysis with Missing Data. Wiley (2002). <https://doi.org/10.1002/9781119013563.ch1>
11. Morris, J.: Your BMW Could Have a Digital Twin—Here’s How It Changes Everything. <https://www.forbes.com/sites/jamesmorris/2023/12/30/your-bmw-could-have-a-digital-twin-heres-how-it-changes-everything/?sh=cbf557553424>. Accessed 3 June 2024
12. Neves, D.T., Alves, J., Naik, M.G., Proença, A.J., Prasser, F.: From missing data imputation to data generation. J. Comput. Sci. **61**, 101640 (2022). <https://doi.org/10.1016/j.jocs.2022.101640>

13. Nguetilbaye, A., Wang, H., Mahamat, D.A., Elgendy, I.A., Junaidu, S.B.: Methods for detecting and correcting contextual data quality problems. *Intell. Data Anal.* **25**(4), 763–787 (2021). <https://doi.org/10.3233/IDA-205282>
14. Ou, H., Yao, Y., He, Y.: Missing data imputation method combining random forest and generative adversarial imputation network. *Sensors* **24**(4), 1112 (2024). <https://doi.org/10.3390/s24041112>
15. Pan, R., Yang, T., Cao, J., Lu, K., Zhang, Z.: Missing data imputation by K nearest neighbours based on grey relational structure and mutual information. *Appl. Intell.* **43**(3), 614–632 (2015). <https://doi.org/10.1007/s10489-015-0666-x>
16. Pantanowitz, A., Marwala, T.: Missing data imputation through the use of the random forest algorithm. In: Yu, W., Sanchez, E.N. (eds.) *Advances in Intelligent and Soft Computing*. AISC, vol. 116, pp. 53–62. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-03156-4_6
17. Ray, S.: A quick review of machine learning algorithms. In: *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, pp. 35–39 IEEE (2019). <https://doi.org/10.1109/COMITCon.2019.8862451>
18. Shahbazian, R., Greco, S.: Generative adversarial networks assist missing data imputation: a comprehensive survey and evaluation. *IEEE Access* **11**, 88908–88928 (2023). <https://doi.org/10.1109/ACCESS.2023.3306721>
19. Silva-Ramírez, E.L., Cabrera-Sánchez, J.F.: Co-active neuro-fuzzy inference system model as single imputation approach for non-monotone pattern of missing data. *Neural Comput. Appl.* **33**(15), 8981–9004 (2021). <https://doi.org/10.1007/s00521-020-05661-5>
20. Silva-Ramírez, E.-L., López-Coello, M., Pino-Mejías, R.: An application sample of machine learning tools, such as SVM and ANN, for data editing and imputation. *Stud. Fuzziness Soft Comput.* **358**, 259–298 (2018). https://doi.org/10.1007/978-3-319-62359-7_13
21. Silva-Ramírez, E.L., Pino-Mejías, R., López-Coello, M., Cubiles-de-la-Vega, M.D.: Missing value imputation on missing completely at random data using multilayer perceptrons. *Neural Netw.* **24**(1), 121–129 (2011). <https://doi.org/10.1016/J.NEUNET.2010.09.008>
22. Twala, B.: An empirical comparison of techniques for handling incomplete data using decision trees. *Appl. Artif. Intell.* **23**(5), 373–405 (2009). <https://doi.org/10.1080/08839510902872223>
23. Yoon, J., Jordon, J., van der Schaar, M.: GAIN: missing data imputation using generative adversarial nets. In: *35th International Conference on Machine Learning, ICML 2018*, vol. 13, pp. 9042–9051 (2018)
24. Zhang, S.: Nearest neighbor selection for iteratively kNN imputation. *J. Syst. Softw.* **85**(11), 2541–2552 (2012). <https://doi.org/10.1016/j.jss.2012.05.073>
25. Sheni, D.N., Basson, A.H., Grobler, J.: Appendix to evaluating algorithms for missing value imputation in real battery data (2024). <https://github.com/Shenid121/Academic-Papers-Appendices-Resul>



Using Pseudo Cases and Stratified Case-Based Reasoning to Generate and Evaluate Training Adjustments for Marathon Runners

Ciara Feely^(✉), Brian Caulfield, Aonghus Lawlor, and Barry Smyth

University College Dublin, Dublin, Ireland

{ciara.feely,brian.caulfield,aonghus.lawlor,barry.smyth}@ucd.ie

Abstract. Recommender systems have become a regular feature in our daily lives. They influence the books we read, the movies we watch and the content we consume on social media. There is opportunity to apply recommender systems to more complex domains, such as exercise, and in this paper we consider how such systems can play a role in supporting runners as they train for a marathon. However, making recommendations for more complex domains introduces additional challenges such as how to provide varied recommendations and how to evaluate these suggestions. In this work we address both of these issues using a stratified case-based recommendation approach and the use of so-called *pseudo-cases* for evaluation. The stratified approach allows for different recommendations to be generated for each runner based on whether they would like to continue along their current training trajectory, or target a more ambitious or a more conservative goal. We further describe how to evaluate these recommendations in terms of their *feasibility*, *plausibility*, *effectiveness* and *safety* using a large-scale, real-world dataset of more than 130,000 runners and their marathon training experiences.

Keywords: case-based reasoning · recommender systems · marathon running · training plan recommendation

1 Introduction

Preparing to run a marathon typically involves 12–16 weeks of training encompassing a variety of sessions to build the endurance, speed, and strength a runner needs to complete the 42.2 km (26 miles) distance. Many web-sites and books offer marathon training plans that are broadly catered towards a specific finish-time goal, for example 4.5 or 4 h marathon training plans. These plans typically provide a day-by-day, week-by-week description of how a runner should train to achieve this goal including distances and speeds for individual sessions, rest days etc. However, these plans are usually static in nature and do not adapt to the runners who follow them. Less experienced recreational runners, in particular, are likely to stray from these plans for a variety of reasons: they may have

over- or under-estimated their ability and fitness; they may become injured or experience other training disruptions; or their motivation may wane.

The widespread adoption of wearable sensors and training apps like Strava¹ and Runkeeper² mean that large amounts of activity data are now routinely captured about a runner's training activities. This activity data has created new opportunities to use ideas from machine learning and recommender systems to provide more personalised training programmes by tracking training progress and making interventions when runners appear to deviate from their target plan. Recently, there have been several efforts to use this type of activity data to provide more targeted training recommendations to achieve a target goal-time [6], or to improve race-day performance [10], or to reduce the risk of injury [8].

In this work we adopt similar ideas to generate training recommendations for runners. A key novelty in this work is that we generate these recommendations by explicitly targeting a range of performance goals, not just the runner's stated target but also more conservative and more ambitious targets that may be within reach. Similar ideas were explored in [11] to advise runners as they returned from injury, but here we consider training more broadly by providing training recommendations to runners as they train. To do this we use a *stratified* case-based reasoning approach to recommendation which is designed to generate training recommendations for runners based on the training of groups of similar runners with similar, less ambitious, or more ambitious performance goals. We do this to provide runners with options to adjust their training based on their progress.

A second contribution of this work concerns the evaluation of these recommendations. Similar efforts in the past have been limited to comparing a specific recommendation to the actions of similar runners. In this work, we use recommendations to generate so-called *pseudo cases* by combining a runner's training to date with a recommended training adjustment and thus mimic the future training of the target runner. We then analyse these pseudo cases according to:

1. *Feasibility* – Are the recommended training adjustments reasonable in terms of their impact on training load? For example, it would not be reasonable to expect a runner to significantly increase their weekly distance or improve their training pace, or both, in the short term.
2. *Plausibility* – Is there evidence that the combined training to date and the recommended future training occurs in practice? Do some runners with similar training histories to the target runner also follow similar training plans to the ones recommended?
3. *Effectiveness* – Is the recommended training adjustment likely to achieve the desired performance on race-day? For example, does the runner's predicted finish time, using their training history and recommended plan, match the target finish time?

¹ www.Strava.com.

² www.Runkeeper.com.

4. *Safety* – Is the training adjustment safe? For example, it is well known that runners need to carefully monitor their training load (overall weekly distance and pace) to avoid over-training and injury and the conventional wisdom cautions against training adjustments that increase weekly training load by more than 10%.

Some elements of this approach have been described and presented in [12] as part of a fruitful feasibility study. Here we present an extended treatment with a more detailed evaluation including a new feasibility analysis of the recommendations produced. This evaluation is based on a large-scale, real-world dataset containing the detailed training histories of more than 130,000 marathon runners.

2 Related Work

Training for a marathon typically requires 12–16 weeks of intense training composed of a carefully timed sequence of activities, including long runs (to build endurance), interval sessions (to build speed), hill sessions (for strength) and slow recovery runs (to aid recovery). Therefore, recommending training plans or specific training sessions or sequences of sessions represents a much more challenging recommendation task than more familiar recommendation tasks such as recommending books or music or movies [24].

Notwithstanding this additional complexity, the idea of a virtual coach capable of making such recommendations is compelling and has recently been explored in the literature [1, 4, 5, 18, 19, 22, 30, 31]. Several systems have been developed with runners in mind, for example, to help runners keep their heart rate in a certain range [21], or to generate optimal interval training sessions (in terms of maximal calorie burn) [29], or to provide in-race pacing advice [2] or post-race advice on how to improve performance [15]. In other sports, reinforcement learning [25] and particle swarm optimisation [14, 16, 17] have been used for related tasks.

Particularly relevant for this work is a growing body of work in the case-based reasoning (CBR) community to support marathoners [11, 26, 27] and ultra-marathoners [23] and endurance skaters [28]; CBR methods have proven to be effective because their nearest-neighbour approach facilitates a useful form of local reasoning that is readily explainable to an athlete. Previous work on case-based marathon training recommendations has looked at making short-term recommendations (next week of training) to help runners achieve a desired performance goal [6, 10], or to assess injury risk [8] or build fitness [3].

The work presented here aims to provide runners with a set of targeted training adjustments over the course of several weeks of training. The main technical contribution stems from a novel, stratified CBR approach to recommendation as well as the use of so-called pseudo-cases to evaluate these recommendations. While CBR techniques have been used in the past for similar tasks [6, 10, 26], they have typically been used to make a single recommendation or prediction,

based on the performance or training of a single set of similar runners. The stratified approach, adopted here, distinguishes between several groups of similar runners, with different performance/ability properties, in order to generate a set of training recommendations based on these different performance groupings; as mentioned above, similar ideas were presented in [11] to provide injured runners with training advice as they tentatively returned to training after a period of injury.

3 Stratified CBR for Training Recommendations

Our aim in this work is to provide marathon runners with recommendations about how they may wish to adjust their current training, depending on their progress so far and their current goals. It is not unusual for runners to wish to adjust their training. Perhaps it has been going well and they feel inclined to target a more ambitious goal, or maybe their current plan has been proving to be too challenging and they wish to target a less ambitious goal. Either way, the challenge is to make an adjustment that is right for their needs. If they wish to target a more ambitious goal then how should they increase their training load to achieve this, but in a safe way that avoids the risk of over-training. And, if they wish to pull back on training then how can they do this without compromising all of the hard work they have already invested in training to date. In this section, we describe the stratified case-based reasoning approach for generating these recommendations.

3.1 Representing Training Sets

Consider a runner r , w weeks before their target marathon m . Most smart-watches and training apps record a training session as various time series. For our purposes, a training session is made up of a time series that captures their time and distance at regular (100 m in this case) intervals during the activity. Each such time series can be used to determine the distance, mean pace etc. of the training session or activity. As in previous work [6, 7, 10], r 's training to date is represented by several types of weekly training features as follows:

1. Mean weekly pace (mins/km)
2. Fastest 10 km pace (mins/km)
3. Total weekly distance (km)
4. Longest run distance (km)
5. Number of rest days (unit)
6. Maximum consecutive days without training (unit).

Each of these six features types produce three different features, by calculating the cumulative average, maximum, and minimum values at a given point in training. For example, this means that there are three versions of *mean weekly distance*: (i) the average mean weekly pace across all training weeks up to and

including week w ; (ii) the minimum (fastest) mean weekly pace up to and including week w ; and (iii) the maximum (slowest) mean weekly pace up to and including week w . In this way, these features produce a training representation with 18 (6×3) individual features, representing r 's training in week w , which we refer to as $F_{pre}(r, w, m)$.

Next, we use sequential forward feature selection [20] to reduce these training features to a subset of six features which we use as our final representation of $F_{pre}(r, w, m)$; see also [10]:

1. Overall Fastest 10 km Pace (mins/km)
2. Average Fastest 10 km Pace (mins/km)
3. Fastest Weekly Pace (mins/km)
4. Slowest Weekly Pace (mins/km)
5. Average Weekly Pace (mins/km)
6. Average Weekly Distance (mins/km).

Using the same approach, we generate the feature set $F_{post}(r, m, w)$ to capture a runner's training *after* week w . In this way, a (historical) case for r , w weeks before marathon m contains a set of features describing their training up to and including week w , $F_{pre}(r, m, w)$, and a set of features after week w , $F_{post}(r, m, w)$, plus the future marathon time achieved by r in m , $MT(r, m)$. This serves as a complete case representation; see Eq. 1.

$$C(r, m, w) = \langle F_{pre}(r, m, w) \mid F_{post}(r, m, w), MT(r, m) \rangle \quad (1)$$

3.2 Recommending Training Adjustments

To generate training recommendations for a target runner r_t , w weeks before a future marathon m_u , we first find the $k = 30$ most similar cases using a standard Euclidean distance metric to compare the runner's recent training to date, $F_{pre}(r_t, m_u, w)$, with those of the case base ($F_{pre}(r_i, m_j, w)$). Then we divide these $k = 30$ similar cases into three groups – *fast*, *moderate*, and *slow* – using their future marathon finish times ($MT(r_i, m_j)$), thus forming a stratification to the nearest neighbours. We then generate a set of training adjustments for each stratified group based on the average future training completed, that is the mean of the training features, $F_{post}(r_i, m_j, w)$; we later refer to these recommendations as training *strategies* based on a faster, more moderate, or slower goal. Hence, we can present a runner r_t with options for their future training based on the previously mentioned pacing and distance features, that reflect training that similar runners, who have achieved slower (s_{slow}), intermediate/moderate (s_{mod}), or faster finish-times (s_{fast}), have completed for the weeks after week w . Using these insights the runner can choose to adjust their training corresponding goal time.

For example, if the moderate group is associated with a finish time of 245 min and the fast group is associated with a finish time of 225 min, and the runner can see that the runners achieving a finish of 225 min do so with a modest increase

in weekly distance and reduction weekly pace (training at a faster pace), then r_t can adjust their planned training activities to meet these new distance and pace targets if they wish to achieve a more ambitious finish time. Note that this study concerns making adjustments across the next several weeks of training as one overarching adjustment to training. However, in principle, it is possible to use similar to adjust a runner’s short-term training, for example the next week or two weeks of training by adjusting $F_{post}(r, m, w)$.

4 Evaluation

To explore the efficacy of the recommendations and how runners training for a marathon adopt and respond to the predicted finish-times and recommended advice, a live user study is required. This is beyond the scope of the present study and remains a goal for the future. Here, we offer a retrospective evaluation using a large-scale, real-world dataset of recreational marathon runners to evaluate different aspects of the recommendations produced.

4.1 Dataset and Methodology

The dataset in this evaluation is an anonymised dataset³ of 130,483 marathon training plans from 70,844 unique runners (of which 17% are female) and totalling 7,312,282 individual training activities captured in the 16 weeks of training before a marathon. We use this dataset to generate a marathon training case base, \mathcal{CB} in which each case corresponds to a complete (16 week) block of training and a marathon finish time. We also produce individual case bases cases for each week w of training, \mathcal{CB}_w , each comprising the pre and post training features for each runner r at week w , as described above.

We use a standard 5-fold cross validation approach whereby in each fold, 80% of the runners are used as training cases, and 20% are used as test cases, and this process is repeated such that each runner is used as a test case exactly once. For each target runner r_t and week w , we use \mathcal{CB}_w to generate three training adjustment strategies based on stratified approach described previously, s_{slow} , s_{mod} , and s_{fast} . Then, for each strategy s_G we add the recommended features, $Rec_{post}(s_G)$, to the target runner’s training features to date to produce a *pseudo-case* estimation, C_G , for r_t as per Eq. 2; where G refers to one of the slow, moderate, or fast groups, $Rec_{post}(s_G)$ simply refers to the recommended features and $MT(s_G)$ to the mean marathon time from the corresponding group of similar cases.

$$C_G(r, m, w, s_G) = \left\langle F_{pre}(r, m, w) \mid Rec_{post}(s_G), MT(s_G) \right\rangle \quad (2)$$

Here each C_G represents a hypothetical case of r_t which we then compare to the other cases in \mathcal{CB}_w to calculate the evaluation metrics that follow:

³ The authors access this dataset via a data sharing agreement with Strava Inc.

1. *Feasibility*: Is the recommended strategy achievable for the runner? We estimate this by comparing the training load of the recommendation with the training load of the runner’s actual training prior to recommendation in terms of training distance. Generally the increase/decrease in training load should be within $\pm 10\%$.
2. *Plausibility*: Is the recommended strategy plausible; is the pseudo-case, C_G , an example of a realistic marathon training experience? We estimate this based on the average distance between each C_G and the $k = 10$ most similar cases in \mathcal{CB}_w . The closer it is to its nearest cases the more plausible it is and therefore the more plausible the recommended strategy s_G will be.
3. *Effectiveness*: Does the recommended strategy, s_G , achieve the desired performance? We estimate this using the difference between the average finish-time of the runners generating the recommendation, $MT(s_G)$, and the average finish-times of $k = 10$ most similar cases to C_G in \mathcal{CB}_w .
4. *Safety*: Is the recommended strategy safe, in terms of the future risk of training injury/disruption for the runner? We estimate this using the proportion of the $k = 10$ most similar cases in \mathcal{CB}_w to the pseudo-case, that have a disruption (break) in training greater than 7 days 7 days is chose as a common threshold disruption duration that suggests injury [32].

The results for each of these metrics are discussed in the following sections. Figure 1 shows a the average values for each metric for males and females at different stages in training, while Fig. 2 shows a more fine-grained analysis of the values of each metric at a particular point ($w = 8$) in training.

4.2 Recommendation Feasibility

Figure 1 (a) and (b) show the feasibility of the recommendations for each strategy across the different weeks in training. Feasibility is calculated as the relative change in weekly distance (training load) required to implementing the recommended training: values above 0 indicate an increase in training load while values below 0 indicate a decrease in training load.

As might be expected the fast training strategy is associated with higher increase in training load; for males and females this strategy corresponds to a training load increase of just about 10%, which is on the boundary of what is normally considered to be safe in marathon training. The moderate training load, which corresponds to the runner continuing to train at their base rate, indicates a 5% increase in training load, which corresponds to natural increases in training load as the weeks progress. Finally, the slower training strategy is associated with little or no increase in training load for the weeks following recommendation.

These results are reasonably consistent across the different training weeks but with a modest reduction in training load increases as the runner approaches race-day. This makes sense. As training progresses runners will be accumulating increasing levels of training stress and their ability to up-modulate their training will reduce as race-day approaches. In terms of feasibility, we can say that they different recommendation strategies are broadly feasible, but with the fast group

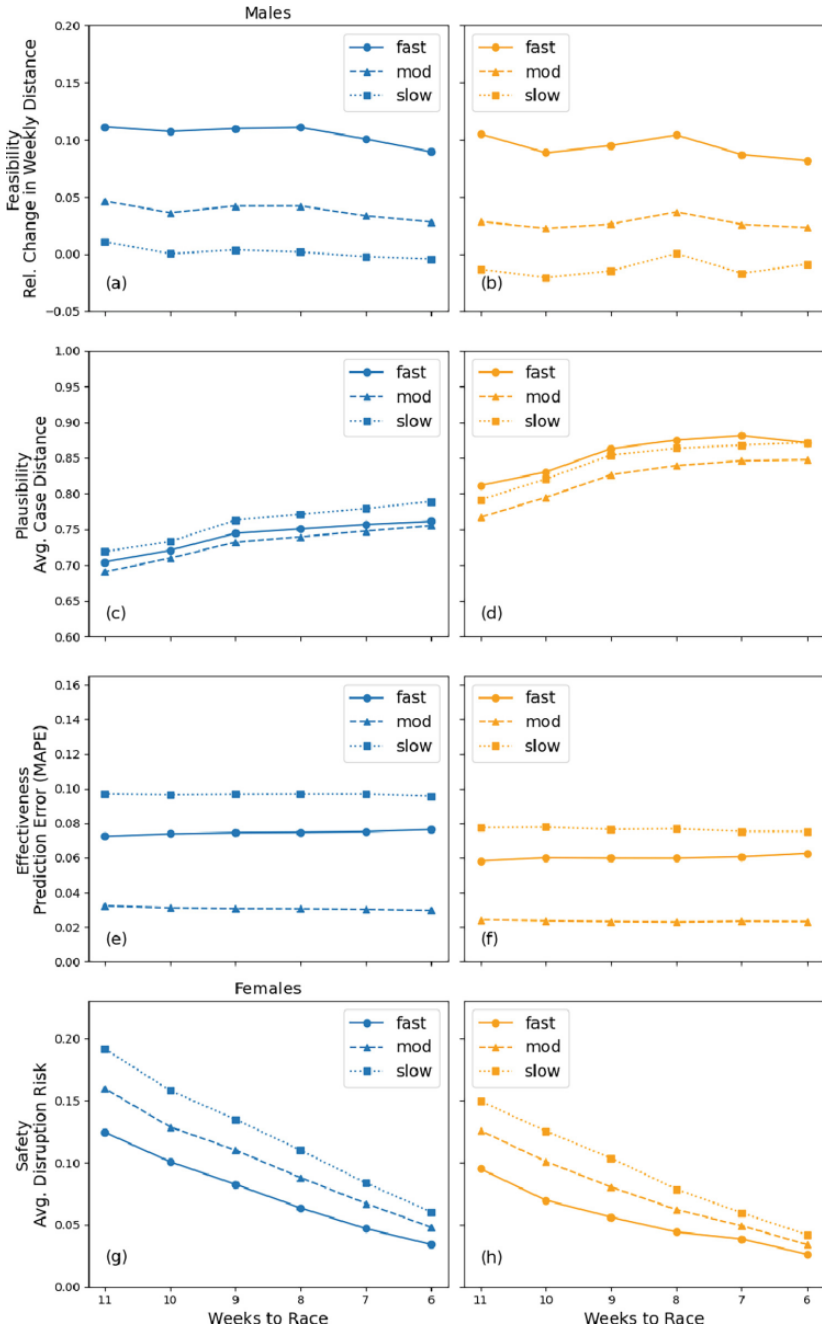


Fig. 1. An evaluation of the recommended training adjustments for the slow, moderate, and fast strategies for males and females in terms of: (i) feasibility, graphs (a) & (b); (ii) plausibility, graphs (c) and (d); and (iii) effectiveness, graphs (e) and (f); and (iv) safety, graphs (g) and (h).

on the edge of what would be considered to be feasible, at least from a safety and injury risk point of view.

When we look more closely at the results in Fig. 2(a) we see that while the median values of feasibility for the fast training strategy are higher than that of the moderate and slow training strategies, the range of values and the 25% and 75% quartile values are not very different. This means that for a portion of runners the faster training strategy would be deemed feasible while for another portion of runners the slow training strategy would require increases in training load that could be deemed infeasible. If we would ideally keep the relative increase in training load between -10 and $+10\%$ then it seems that each training strategy might only be feasible for about 25–50% of runners.

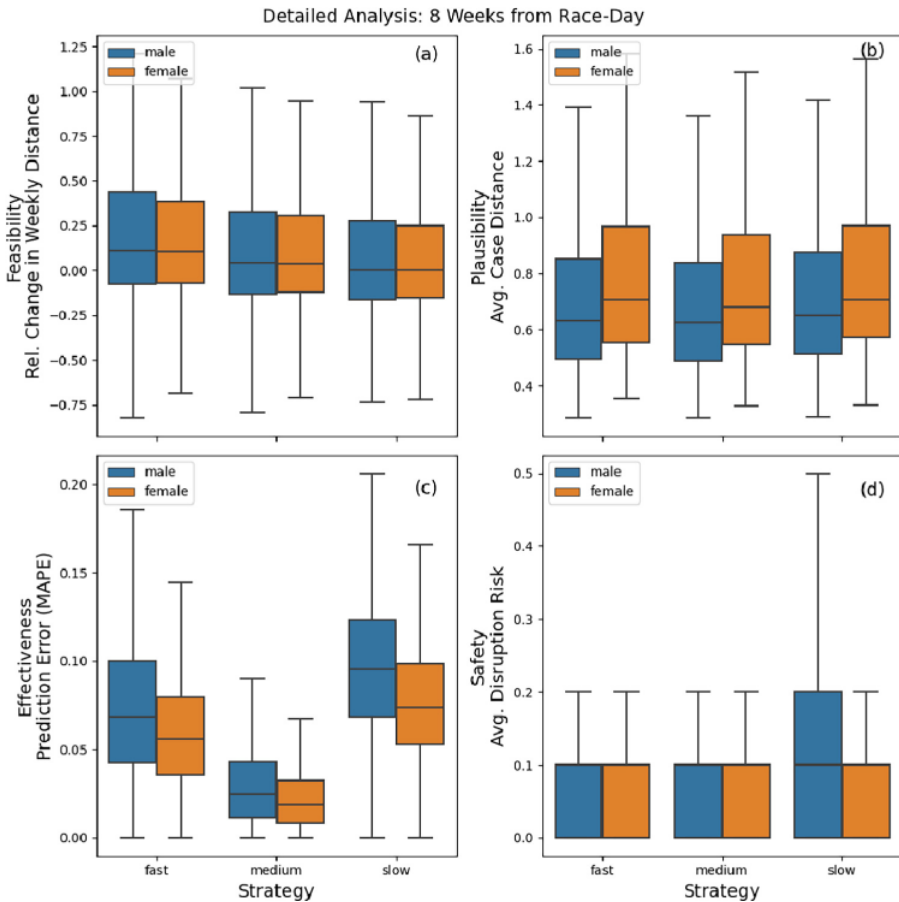


Fig. 2. A series of box-plots displaying the values for each of the following recommendation metrics: (a) feasibility, (b) plausibility, (c) effectiveness, and (d) safety, for each recommendation strategy (slow, medium and fast) and for males (blue/left) and females (orange/right). (Color figure online)

4.3 Recommendation Plausibility

Plausibility captures how likely we are to see the pseudo-case in the case base, that is how often runners, having completed their training to date carry out similar training to that which was recommended in practice. We use case distance as an (inverse) measure of plausibility; greater distance implies lower plausibility. Figure 1(c & d) show the recommendation plausibility for each strategy across different training weeks for male and female runners.

We see that generally the moderate recommendations are more plausible (lowest distance) than the fast and slow recommendations. Somewhat unexpectedly, the fast recommendations demonstrate higher plausibility than the slow recommendations, which may indicate that marathon runners in practice tend to push for faster times. Also note that the recommendations for male runners have higher plausibility (lower distance) than those generated for females, which may imply greater variation in the training patterns of female runners. Finally, it appears that plausibility tends to decrease (case distance increases) as runners approach race-day, suggesting that greater variation in training closer to race-day.

Similar to the previous section we see that in Fig. 2 there is a similar range and 25% and 75% quartile values for each of the training strategies. It is interesting to note that for male runners 50% have average case distance between 0.5–0.85 approximately while for female runners 50% of the average case distance is between 0.55–0.95 approximately. Through experimentation it could be possible to determine an acceptable value for this metric that could then decide whether or not a recommendation could be offered which would be inline with the common recommendation evaluation measure of case coverage.

4.4 Recommendation Effectiveness

The effectiveness of the recommended training plan measures whether or not the recommended plan achieves the performance result it had promised. Here we are not looking at which strategy achieves better performance, as the methodology ensures that the faster strategy will produce faster predicted performance. Rather, for a given runner, we calculate the error when comparing the average finish-time of the runners that generated the recommendation for each strategy, to the predicted finish-time of the pseudo-case generated from the runner’s training to date and the recommended training adjustment. A lower error indicates that the recommendation delivered the desired performance outcome.

Figure 1 (e) and (f) show the average error rate across at different training stages and recommendation strategies for males and females. Here there is a substantial difference in effectiveness across strategies, with a smaller error rate for the moderate training strategy close to 3% for males and 2% for females, which is lower than in previous work on performance prediction [2, 7, 9, 26, 27]. One interpretation of this is that, since the predicted performance is based on “filling in the blanks” in the runner’s training set with similar mid-range runners it follows that if runners were to train “as expected”, we could then predict their

performance with low error rates at any point in training. Also, unlike in the other results, the value for recommendation effectiveness is consistent across all weeks but lower error rates are associated with females regardless of strategy, which is consistent with prior work [7, 9].

Unlike the previous two metrics, when we compare the strategies in detail in Fig. 2 (c) we see a modest range of values for the moderate strategy – between 0–7.5% for females and between 0–9% for males, whereas for the fast and slow strategies 25% of males have performance error between 10–18.5% and 12.5–21%.

4.5 Recommendation Safety

Finally, it is important to ensure the training recommendations do not increase the risk of injury for runners. The data used in this work does not contain specific information about injuries, but as in previous work training disruptions (periods of 7 days or more without training) are used as a proxy for injury [11, 13]. Figure 1 (g) and (h) show the recommendation safety results for males and females. Somewhat unexpectedly, the slow strategy tends to be associated with a higher risk of disruption than either of the other strategies and overall the fast strategy presents with the lowest risk of disruption. While this is unexpected – we might expect that targeting a more ambitious goal increases the likelihood of injury – it is worth noting that this may be an artefact of the data if, for example, the slower runners train with more frequent disruptions as demonstrated in [13]. Injury risk also decreases closer to race-day which may be due to our dataset only including runners that made it to race-day and therefore those who become injured close to race-day and had to drop out of the race and are therefore not present in our dataset.

In the more detailed analysis in Fig. 2 the separation between fast and moderate training strategies becomes less clear while for male runners there is a slightly higher 75% quartile value and a substantially higher upper limit value.

5 Conclusion

This paper presents an extended evaluation of a novel stratified CBR approach to marathon training recommendations using pseudo-cases. The stratified CBR approach makes it possible to recommend different training adjustments to runners based on sets of runners with similar training histories but different marathon finish-times. In this way we can make future training suggestions (adjustments) to a target runner based on runners who have trained in a similar fashion but achieved faster or slower finish-times.

The evaluation analysed pseudo-cases representing hypothetical training scenarios in which the runner completed each recommended training strategy. These were formed by combining the runner’s training to date with the recommended future training for a given strategy. We conducted an offline evaluation of these pseudo-cases to compare the recommended training strategies based on several evaluation criteria (feasibility, plausibility, effectiveness, and safety). Initial

results were positive. The faster training strategy was less feasible than others, as it required a greater increase in training load. There was minimal differences in plausibility across the training strategies. The slower training strategy appears to be associated with higher injury risk, and the moderate strategy is most likely to achieve the predicted finish-time. An interesting finding was that by predicting the future training a runner completes, we can predict an accurate marathon finish-time (2% error rate) when the runner adopts the moderate training strategy (which corresponds to their training as planned). This error rate is lower than previous finish-time prediction models using training data alone [7].

Future work will seek to conduct a live user study to fully evaluate this work and determine whether runners find the training recommendations useful in practice and their impact on runner's performance and injury risk when adopted.

Acknowledgements. Supported by Science Foundation Ireland through the Insight Centre for Data Analytics (12/RC/2289_P2).

References

1. Berndsen, J., Lawlor, A., Smyth, B.: Running with recommendation. In: HealthRecSys@ RecSys, pp. 18–21 (2017)
2. Berndsen, J., Smyth, B., Lawlor, A.: Pace My race: recommendations for marathon running. In: Proceedings of the 13th ACM Conference on Recommender Systems, pp. 246–250. ACM (2019)
3. Berndsen, J., Smyth, B., Lawlor, A.: Fit to run: personalised recommendations for marathon training. In: Fourteenth ACM Conference on Recommender Systems, pp. 480–485 (2020)
4. Buttussi, F., Chittaro, L.: MOPET: a context-aware and user-adaptive wearable system for fitness training. *Artif. Intell. Med.* **42**, 153–163 (2008). <https://doi.org/10.1016/j.artmed.2007.11.004>
5. Buttussi, F., Chittaro, L., Nadalutti, D.: Bringing mobile guides and fitness activities together: a solution based on an embodied virtual trainer. In: Proceedings of the 8th Conference on Human-Computer Interaction with Mobile Devices and Services, MobileHCI 2006, pp. 29–36. ACM, New York, NY, USA (2006). <https://doi.org/10.1145/1152215.1152222>
6. Feely, C., Caulfield, B., Lawlor, A., Smyth, B.: Providing explainable race-time predictions and training plan recommendations to marathon runners. In: Fourteenth ACM Conference on Recommender Systems, RecSys 2020, pp. 539–544. Association for Computing Machinery, New York, NY, USA (2020). <https://doi.org/10.1145/3383313.3412220>
7. Feely, C., Caulfield, B., Lawlor, A., Smyth, B.: Using case-based reasoning to predict marathon performance and recommend tailored training plans. In: Watson, I., Weber, R. (eds.) ICCBR 2020. LNCS (LNAI), vol. 12311, pp. 67–81. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58342-2_5
8. Feely, C., Caulfield, B., Lawlor, A., Smyth, B.: A case-based reasoning approach to predicting and explaining running related injuries. In: Sánchez-Ruiz, A.A., Floyd, M.W. (eds.) ICCBR 2021. LNCS (LNAI), vol. 12877, pp. 79–93. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-86957-1_6

9. Feely, C., Caulfield, B., Lawlor, A., Smyth, B.: An extended case-based approach to race-time prediction for recreational marathon runners. In: Keane, M.T., Wiratunga, N. (eds.) ICCBR 2022. LNCS, vol. 13405, pp. 335–349. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-14923-8_22
10. Feely, C., Caulfield, B., Lawlor, A., Smyth, B.: Modelling the training practices of recreational marathon runners to make personalised training recommendations. In: Proceedings of the 31st ACM Conference on User Modeling, Adaptation and Personalization, pp. 183–193 (2023)
11. Feely, C., Caulfield, B., Lawlor, A., Smyth, B.: A case-based reasoning approach to post-injury training recommendations for marathon runners. In: Recio-Garcia, J.A., Orozco-del-Castillo, M.G., Bridge, D. (eds.) ICCBR 2024. LNCS, vol. 14775, pp. 338–353. Springer, Cham (2024). https://doi.org/10.1007/978-3-031-63646-2_22
12. Feely, C., Caulfield, B., Lawlor, A., Smyth, B.: Recommending personalised targeted training adjustments for marathon runners. In: Proceedings of the 18th ACM Conference on Recommender Systems, RecSys 2024, Bari, Italy, 14–19 October 2024 (2024)
13. Feely, C., Smyth, B., Caulfield, B., Lawlor, A.: Estimating the cost of training disruptions on marathon performance. *Front. Sports Active Living* **4**, 507 (2022)
14. Fister, I., Brest, J., Iglesias, A., Fister, I.: Framework for planning the training sessions in triathlon. In: Proceedings of the Genetic and Evolutionary Computation Conference Companion, GECCO 2018, pp. 1829–1834. Association for Computing Machinery, New York, NY, USA (2018). <https://doi.org/10.1145/3205651.3208242>
15. Fister, I., Fister, D., Deb, S., Mlakar, U., Brest, J., Fister Jr., I.: Post-hoc analysis of sport performance with differential evolution. *Neural Comput. Appl.* **32** (2020). <https://doi.org/10.1007/s00521-018-3395-3>
16. Fister, I., Rauter, S., Yang, X.S., Ljubič, K., Fister, I.: Planning the sports training sessions with the bat algorithm. *Neurocomput.* **149**(PB), 993–1002 (2015). <https://doi.org/10.1016/j.neucom.2014.07.034>
17. Fister Jr., I., Fister, I.: Generating the training plans based on existing sports activities using swarm intelligence. In: Patnaik, S., Yang, X.-S., Nakamatsu, K. (eds.) *Nature-Inspired Computing and Optimization*. MOST, vol. 10, pp. 79–94. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-50920-4_4
18. Henriët, J.: Artificial intelligence-virtual trainer: an educative system based on artificial intelligence and designed to produce varied and consistent training lessons. *Proc. Inst. Mech. Eng. Part P J. Sports Eng. Technol.* (2016). <https://doi.org/10.1177/1754337116651013>. <https://hal.inria.fr/hal-01385534>
19. Hülsmann, F., Göpfert, J.P., Hammer, B., Kopp, S., Botsch, M.: Classification of motor errors to provide real-time feedback for sports coaching in virtual reality - a case study in squats and Tai Chi pushes. *Comput. Graph.* **76**, 47–59 (2018). <https://doi.org/10.1016/j.cag.2018.08.003>. <http://www.sciencedirect.com/science/article/pii/S0097849318301304>
20. Jović, A., Brkić, K., Bogunović, N.: A review of feature selection methods with applications. In: 2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), pp. 1200–1205. IEEE (2015)
21. López-Matencio, P., Alonso, J.V., González-Castano, F., Sieiro, J., Alcaraz, J.: Ambient intelligence assistant for running sports based on K-NN classifiers. In: 3rd International Conference on Human System Interaction, pp. 605–611. IEEE (2010)

22. Mandot, C., Chawla, R.: Artificial intelligence based integrated cricket coach. In: Unnikrishnan, S., Surve, S., Bhoir, D. (eds.) ICAC3 2013. CCIS, vol. 361, pp. 227–236. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36321-4_21
23. McConnell, C., Smyth, B.: Going further with cases: using case-based reasoning to recommend pacing strategies for ultra-marathon runners. In: Bach, K., Marling, C. (eds.) ICCBR 2019. LNCS (LNAI), vol. 11680, pp. 358–372. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-29249-2_24
24. Ricci, F., Rokach, L., Shapira, B. (eds.): Recommender Systems Handbook. Springer, Cham (2015). <https://doi.org/10.1007/978-1-4899-7637-6>
25. Silacci, A., Khaled, O.A., Mugellini, E., Caon, M.: Designing an e-coach to tailor training plans for road cyclists. In: Ahram, T., Karwowski, W., Pickl, S., Taiar, R. (eds.) IHSED 2019. AISC, vol. 1026, pp. 671–677. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-27928-8_102
26. Smyth, B., Cunningham, P.: Running with cases: a CBR approach to running your best marathon. In: Aha, D.W., Lieber, J. (eds.) ICCBR 2017. LNCS (LNAI), vol. 10339, pp. 360–374. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-61030-6_25
27. Smyth, B., Cunningham, P.: An analysis of case representations for marathon race prediction and planning. In: Cox, M.T., Funk, P., Begum, S. (eds.) ICCBR 2018. LNCS (LNAI), vol. 11156, pp. 369–384. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01081-2_25
28. Smyth, B., Willemsen, M.C.: Predicting the personal-best times of speed skaters using case-based reasoning. In: Watson, I., Weber, R. (eds.) ICCBR 2020. LNCS (LNAI), vol. 12311, pp. 112–126. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58342-2_8
29. Suh, M.K., Nahapetian, A., Woodbridge, J., Rofouei, M., Sarrafzadeh, M.: Machine learning-based adaptive wireless interval training guidance system. *Mob. Netw. Appl.* **17**(2), 163–177 (2012). <https://doi.org/10.1007/s11036-011-0331-5>
30. Trejo, E.W., Yuan, P.: Recognition of yoga poses through an interactive system with Kinect based on confidence value. In: 2018 3rd International Conference on Advanced Robotics and Mechatronics (ICARM), pp. 606–611 (2018). <https://doi.org/10.1109/ICARM.2018.8610726>
31. Waßmann, I., Graf von Malotky, N.T., Martens, A.: Train4U - mobile sport diagnostic expert system for user-adaptive training. In: Lames, M., Danilov, A., Timme, E., Vassilevski, Y. (eds.) IACSS 2019. AISC, vol. 1028, pp. 77–85. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-35048-2_10
32. Yamato, T.P., Saragiotto, B.T., Lopes, A.D.: A consensus definition of running-related injury in recreational runners: a modified Delphi approach. *J. Orthop. Sports Phys. Therapy* **45**(5), 375–380 (2015). <https://doi.org/10.2519/jospt.2015.5741>

Applications of Machine Learning



Emotion Detection in Hindi Language Using GPT and BERT

Ritika Agarwal^(✉)  and Noorhan Abbas

University of Leeds, Leeds, UK

ritika.agarwal24@gmail.com, n.h.abbas@leeds.ac.uk

Abstract. Emotion detection in textual data plays an important role in various NLP applications and is an important branch of sentiment analysis. There is a dearth of applications of emotion detection in low resource languages such as Hindi. There are a number of challenges in the available open-source resources in Hindi language such as imbalanced datasets and lack of understanding of textual nuances specific to Hindi. This paper presents a comprehensive study on emotion detection in Hindi text using Large Language Models (LLMs). We will be leveraging GPT-3.5-turbo API for dataset augmentation using few-shot learning and zero-shot learning techniques and further fine-tuning pretrained BERT and mBERT models. This study's key findings include generating two balanced datasets to establish baselines, fine-tuning models for emotion detection and evaluating their performance. There is a significant improvement in Accuracy and F1-score, demonstrating the effectiveness of the approach employed. We further discuss the future directions for research including multimodal emotion detection and cross lingual applications along with ethical considerations in deploying emotion detection systems. This research contributes to advancing the understanding and applications of emotion detection in Hindi text such as customer feedback assessment, social media sentiment analysis, mental health evaluation and human-computer interaction (HCI) via virtual assistants like Alexa and Siri.

Keywords: Fine-tuning mBERT · Hindi Emotion detection · Sentiment analysis · Data augmentation · Synthetic data generation

1 Introduction

With 690.5 million speakers Hindi is the third largest language spoken globally [4]. Hindi is the official language of India, and it is also spoken prominently in eight other countries including Fiji, UAE and Singapore [25]. The linguistic landscape of Hindi is enriched by its association with several other languages and dialects prevalent in the Indo Gangetic region e.g. Awadhi and Bhojpuri [22].

The digital age has seen a significant increase in Hindi's influence. There are approximately 692 million Internet users in India as of 2023 out of which 467 million are active social media users [10]. These active social media users spend an average of 141.6 min, i.e. roughly two and a half hours per day on social media platforms [24]. All the major social media platforms like Instagram, Facebook, X, LinkedIn and WhatsApp have Hindi

as a supported language. This shows the growing digital footprint of the language as well as its integration into the online discourse.

With the advent of deep learning, NLP has experienced a rapid increase in advancements. There has been an increase in understanding and analyzing human emotions conveyed through text data. Emotion detection is an important sub-field of NLP. It has a critical role across various applications like sentiment analysis, customer feedback assessment, mental health evaluation and human-computer interaction via virtual assistants like Alexa and Siri. The primary objective of emotion detection system is to recognize and categorize fine-grained emotions such as Happiness, Anger, Sadness, Surprise, Fear and Disgust, if present within text inputs.

There are considerable research efforts dedicated to emotion detection in English. However, there is a dearth of resources such as annotated datasets and research studies for languages other than English. This scarcity is particularly noticeable in the availability of comprehensive datasets tailored for emotion detection in Hindi. The existing datasets, which are few, are often characterized by their limited size, sparse records, and imbalanced class distributions which result in significant challenges for training robust and accurate emotion detection models.

The scarcity of resources in Hindi language poses a significant barrier to advancing research in emotion detection for the language. Due to the unique linguistic characteristics and cultural nuances present in Hindi texts there is an urgent need to develop specialized approaches and datasets to facilitate effective emotion detection and analysis. To cater to the needs of diverse linguistic communities worldwide, addressing this gap in resources and knowledge is paramount to unlocking the full potential of emotion detection in Hindi.

Hence, this study aims to address these critical gaps in emotion detection research for Hindi by generating two balanced datasets in Hindi using LLMs and existing Hindi datasets by using few-shot and zero-shot learning techniques. Additionally, we will fine-tune two classifiers using BERT and mBERT for enhanced emotion detection accuracy.

2 Related Work

In this section, we will look at the relevant work already done in the following areas viz. (1) Datasets with emotion classification (2) Techniques for emotion detection in text (3) Capability of LLMs (4) Few-Shot learning.

2.1 Datasets with Emotion Classification

The availability of annotated datasets is crucial for training emotion detection models. Kumar et.al [12] has done a detailed survey on the datasets available in Hindi and English. In addition to this there are several other datasets present in Hindi [1, 5–7, 11, 16, 19], other Indian regional languages [16], and Hindi-English code-mixed text [18] for emotion classification, the details of which are present in Table 1.

Two datasets are employed in this study: Bhaav [11] and Emo-Dis-Hi [1].

Bhaav [11]. Kumar et al. [11] mentions that Bhaav is the first and largest Hindi text corpus for emotion analysis. It consists of 20,304 sentences from 230 short stories

Table 1. Summary of various datasets for emotion detection

Dataset	Content and Language	Feature	Emotion Model
ISEAR	7665 labeled English sentence	7 emotion labels	Discrete
SemEval	1250 Arabic and English texts from Tweets and newspapers	6 emotion labels	Discrete
EMOBANK	10000 labeled English sentence, spanning over a wider domain	6 emotion labels	Dimensional
WASSA-2017 Emolnt	Annotated tweets in English	4 emotion labels	Discrete
Daily Dialog	13118 labeled sentences in English	7 emotion labels	Discrete
Emotion-Stimulus	1594 labeled sentence in English	7 emotion labels	Discrete
Smile dataset	3085 tweets in English	5 emotion labels	Discrete
BHAAV dataset	20304 labeled Hindi sentence	5 emotion labels	Discrete
EmoInHindi	1,814 dialogues with 44,247 utterances in Hindi labeled	16 emotion labels	Dimensional
Emo-Dis-HI	2667 sentences in Hindi labeled	9 emotion labels	Discrete
Hindi EmotionNet	4430 sentences in Hindi labeled	3 emotion labels	Discrete
Children Story	780 sentences in Hindi labeled	5 emotion labels	Discrete
HindiMD	1818 tweets in Hindi labeled	3 emotion labels	Discrete
SAIL Dataset	1052 Bengali, 1278 Hindi, 1103 Tamil tweets labeled	3 emotion labels	Discrete
Social media Text	12000 Hindi English code-mixed social media text	3 emotion labels	Discrete

spanning across 18 genres. It aims to analyze the emotions expressed by characters in Hindi stories and provide annotations for five emotion categories- anger, joy, sad, suspense and neutral. Kumar et al. [11] emphasizes the significance of Bhaav as a resource for studying emotions in Hindi text. The dataset addresses the lack of emotion analysis resources in low-resource languages. The study highlights the potential applications of the Bhaav dataset, discusses challenges in annotating emotions in Hindi, and presents baseline classifier performances. The main limitation of this dataset is the imbalance towards neutral class.

Emo-Dis-Hi [1]. It is a newly created dataset for emotion recognition in Hindi, developed from scratch due to the absence of existing resources in this domain. The dataset focuses on disaster-related news documents obtained by crawling popular Hindi news websites. Ahmad et al. [1] explains that the sentence-level annotation was performed using Plutchik’s wheel of emotions. Each sentence was labelled by the annotators with one of the nine emotion categories, including sadness, sympathy/pensiveness, fear/anxiety, optimism, joy, disgust, anger, and surprise. The sentences that did not evoke

any emotion were labeled as no-emotion. Due to the focus on disaster-related content, the dataset exhibits a skew towards negative sentiment.

2.2 Techniques for Emotion Detection in Text

Kumar et al. [12] focuses on Hindi text emotion analysis using the Bhaav dataset. Bhaav dataset is imbalanced in nature, hence, random under sampling was performed first to reduce the number of records in the majority class and then oversampling was done by using SMOTE to oversample the minority class in order to achieve a more balanced distribution. The study includes training on various machine learning and deep learning techniques including mBERT to predict emotions in Hindi sentences. The study further shows that Word2Vec models used in traditional machine learning algorithms generate context independent embeddings while mBERT generates context dependent embeddings and thus, achieves the best performance. The study acknowledges the challenge of limited availability of emotion recognition models in languages other than English, emphasizing the need for more research in regional and local languages.

Ahmad et al. [1] also addresses the challenge of emotion detection in Hindi by creating a new dataset for emotion detection in disaster domain called Emo-Dis-Hi. The dataset has nine emotion classes identified. The study shows a deep learning framework that leverages information from a resource rich language like English to enhance emotion detection in Hindi. CNN and Bi-LSTM are used as base learning models to achieve this. A cross lingual word embedding representation of words is generated in the shared embedding space. Neural networks are trained on existing datasets and then transfer learning strategies are employed for emotion classification in Hindi. The study concludes that knowledge from a resource-rich language like English can be effectively transferred across languages and domains for improved emotion detection in Hindi.

Nandwani et al. [13] reviews various techniques for sentiment analysis and emotion detection on datasets of various domains mainly in English. Some of the methods reviewed are lexicon based (dictionary based, corpus based), machine learning based (naive bayes, SVM), deep learning based (CNN, LSTM) and hybrid approaches. Transfer learning is also reviewed. Dictionary based approaches offer adaptability whereas corpus-based approaches offer increased accuracy within a domain. The performance of machine learning algorithms depends upon the size of dataset and pre-processing performed. Deep learning algorithms like LSTM and RNN with attention usually outperform traditional methods in large datasets. BERT models also achieved high accuracy.

2.3 Capability of LLMs

Venkatakrishnan et al. [20] investigates the use of pre trained transformer-based models like GPT 3.5 and RoBERTa for emotion detection in NLP. The main focus of the study is to examine the potential of these models as automatic label generators in order to improve accuracy. The dataset was collected using SNScrapper for the tweets pertaining to two events - the murder of Zhina (Mahsa) Amini in Iran and earthquake in Turkey and Syria. The tweets were collected without applying any language filters and contain Persian and Turkish tweets as well. A sample of 10000 tweets was collected for the

purpose of the study. Both the models show promising results in detection. GPT 3.5 with its generative nature demonstrates a strong understanding of language context and captures complex emotional nuances but struggles with fine-grained emotions. However, for a limited set of labels, RoBERTa because of its fine-tuning abilities and extensive pre training for emotions provides more accurate predictions. The choice between GPT and RoBERTa depends on the specific requirements of downstream ML tasks.

Sahu et al. [17] proposes a prompting-based approach to generate labeled training data for intent classification using off the shelf large language models like GPT 3. The study evaluates the proposed approach using few-shot learning technique. A simple prompt structure based on seed intent and available examples is given to GPT-3 to generate diverse training data. The study concludes that in tasks with distinct intents the GPT generated data significantly boosts the performance of intent classifiers. Sahu et al. [17] suggests that GPT could be used as a classifier to filter out unfaithful examples and enhance data quality. Relabeling GPT-generated data by humans further improves intent classification performance.

2.4 Few-Shot Learning

Wang et al. [21] surveys few-shot learning (FSL) technique in machine learning. FSL leverages prior knowledge to rapidly adapt to new tasks with minimal supervised examples. The study provides a formal definition of FSL and discusses the similarities and differences of FSL with related machine learning problems like transfer learning, imbalanced learning etc. A deep dive is performed into core issue with FSL and different FSL methods based on utilization of prior knowledge in data, model and algorithm are discussed. Pros and cons of each category are then discussed. Potential applications of few-shot learning including robotics, NLP are discussed.

3 Design and Methodology

The research study comprises of several key steps as outlined in Fig. 1. The two primary datasets Bhaav [11] and Emo-Dis-Hi [1] serve as the foundation for analysis. By leveraging the capabilities of GPT-3.5-turbo API for text generation the study employs advanced techniques such as prompt engineering, zero-shot learning and few-shot learning to augment the dataset and enhance its quality. For emotion detection or classification two state of the art models - BERT and mBERT are utilized. Baseline performances are evaluated as well as fine-tuning is performed to address the nuances of emotion detection in Hindi text. These steps form the methodology framework for the research study leading to an exhaustive analysis of emotion detection in Hindi text using the latest NLP techniques.

3.1 Datasets

In this study, we selected Bhaav and Emo-Dis-Hi datasets based on several critical factors that align with the research objectives. Bhaav and Emo-Dis-Hi datasets were chosen primarily because they are among the most comprehensive resources available for Hindi

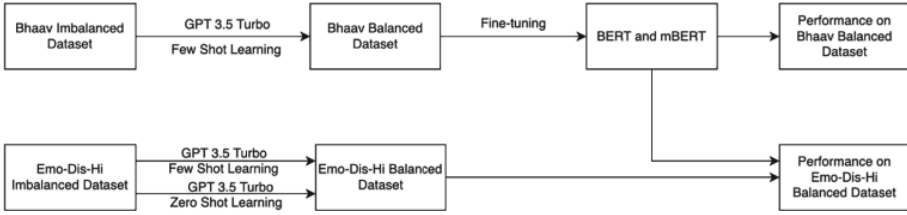


Fig. 1. Methodology for the research study

emotion detection, covering a wide range of emotions and contexts. Bhaav dataset's literary focus allows for the exploration of complex emotional expressions in narrative form, while Emo-Dis-Hi dataset's emphasis on disaster-related content provides a more intense emotional spectrum. This combination of datasets is essential for testing the versatility and robustness of our models across different types of textual content. The basic information about the datasets is already discussed in Sect. 2.1.

Bhaav [11]. The annotation was performed by three native Hindi speaking volunteers with at least ten years of formal education in Hindi. The dataset however exhibits a significant imbalance with most instances classified as neutral (57.6%). The remaining emotion categories constitute a smaller proportion of the overall population. e.g. anger (7.2%), suspense (7.44%), joy (12.13%), and sadness (15.6%).

Emo-Dis-Hi [1]. The annotation was performed by three native Hindi speaking volunteers with post-graduate in linguistics and Hindi as their primary language. The dataset however exhibits a significant imbalance with most instances classified as sadness (55.4%) followed by sympathy/pensiveness (13.8%), no emotion (8.2%), optimism (6.6%), fear/anxiety (5.4%), joy (5.3%), disgust (2.3%), anger (1.7%) and surprise (1.2%).

The class imbalance necessitates pre-processing to create a balanced dataset, which is crucial for enabling the model to effectively learn and differentiate between all emotion classes.

3.2 Models

GPT-3.5-turbo API [14]. This API is used in this study for text generation. This model has been developed by OpenAI and it represents a state-of-the-art LLM (large language model) known for its ability to generate human like text across different tasks like text completion, summarization, translation and question and answering. GPT-3.5 has a complete understanding of language patterns, semantics and context as it has been trained on a vast dataset consisting of literature, articles, websites and other textual sources. It has also been extensively trained in various languages including Hindi ensuring a strong understanding of the language's syntax, semantics, and context. The key features of this API include its support for prompt engineering, enabling users to provide specific instructions to the model in order to guide the text generation process towards desired outcomes in terms of domain, format etc. In addition to this, the API also facilitates advanced techniques like zero-shot learning and few-shot learning. These techniques help the model to perform tasks such as generating text on the basis of examples provided.

In this research, the GPT-3.5 turbo API serves as a powerful tool for dataset augmentation, enabling the generation of high-quality text data to create balanced datasets from existing Bhaav and Emo-Dis-Hi datasets using few-shot and zero-shot learning approaches.

BERT [2]. The Bidirectional Encoder Representations from Transformers (BERT) model is used in this study for text classification or emotion detection tasks. This model has been developed by Google. BERT is known for its bidirectional contextual understanding using the transformer architecture to capture semantic meaning from text data. With its multiple layers of self-attention mechanisms, BERT can effectively model contextual dependencies in both directions of a sequence.

In this research, the pre-trained BERT model is fine-tuned specifically for the task of emotion detection from text. Fine-tuning involves adapting the parameters of the pre-trained BERT model to the target classification task by using a labeled dataset. By fine-tuning on task-specific data, BERT is expected to achieve improved performance. BERT was selected to show the differences between the two models (mBERT and BERT), one specifically trained on Hindi text and the other not.

mBERT [3]. The multilingual BERT (mBERT) model is used in this study for text classification or emotion detection tasks. It is an extension of the BERT model and is designed to handle multilingual text data effectively. mBERT was included in the methodology as it offers several advantages such as ability to handle multilingual text data, capturing cross-lingual semantic relationships and providing robust performance across diverse languages. By leveraging mBERT for emotion detection in Hindi text, we aim to enhance the accuracy and effectiveness of our classification model for capturing nuanced emotional states expressed in textual data. mBERT was selected for this research considering its accessibility and baseline performance on Bhaav dataset [12].

3.3 Hyperparameter Selection

GPT-3.5-turbo API. The parameters below were set up during text generation tasks.

- Temperature - This parameter controls the randomness of the generated text. Temperature is set as 0.8 to get some degree of creativity in outputs.
- Max tokens - This parameter limits the maximum number of tokens (words or sub-words) in the generated text. No limit has been set for this since we are generating sentences and we dont want to limit their length.

BERT and mBERT. The parameters below were setup during emotion detection tasks.

- Learning Rate - Hyper Parameter Training was done to select the best learning rate from [2e-5, 3e-5, 5e-5].
- Epochs - Hyper Parameter Tuning was done to select the best number of epochs from [5, 10].
- Optimizer - BERT models are typically trained using gradient-based optimization algorithms such as Adam or stochastic gradient descent (SGD). AdamW has been used as an optimizer.
- Maximum Sequence Length - This has been set as 70.
- Batch Size - This has been set as 32.

3.4 Performance Metrics

Accuracy measures the proportion of correctly classified instances out of the total instances in the dataset. While accuracy is a straightforward metric, it might not be suitable for imbalanced datasets as it can be misleading when one class dominates the dataset. Since the resultant dataset is balanced it is a good metric for evaluation.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Additionally, F1 score, which is the harmonic mean of precision and recall, can provide a balanced assessment of the model's performance, especially in situations where class imbalance exists. It combines both precision and recall into a single metric, providing a more comprehensive evaluation of the model's ability to correctly classify instances across all classes.

$$F1\ score = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

4 Experimental Results and Discussion

The experiments were conducted on a Spyder IDE running on a MacBook Pro with 8 GB of RAM and 2.7 GHz Dual-Core Intel Core i5 Processor. The steps below were performed and results were obtained.

4.1 Generation of a New Balanced Dataset Using Existing Bhaav Dataset with Few-Shot Learning and Prompt Engineering Using GPT-3.5-turbo API

A new balanced dataset is generated using advanced few-shot learning techniques with the GPT-3.5-turbo API. The few-shot learning approach involved providing 10 carefully selected sentences from the original dataset across five emotion categories: four from neutral, two each from anger and joy and one from sadness and suspense. These examples were chosen based on their clear expression of emotion, appropriate length and alignment with the overall tone of the dataset.

Using these examples, we applied prompt engineering to generate additional records for the underrepresented categories. While we aimed to generate 1,000 records (~200 per category), API token limitations led to generating 979 sentences. Multiple API calls were made to achieve the desired output, followed by post-processing to ensure that the generated sentences were contextually consistent with the original dataset. This new dataset balanced the previously skewed distribution of emotions, with roughly equal representation across all five categories. (Table 2).

To ensure the generated dataset was representative of the original Bhaav dataset, several steps were taken to validate the quality and consistency of the generated data. We evaluated the semantic and syntactic consistency between the original and generated sentences. Sentences that did not meet the required standards were either revised or discarded, ensuring that only high-quality data was added to the dataset. Checks were

Table 2. Emotion Categories in GPT generated dataset from Bhaav Dataset

Sentiment	Count
Neutral	205
Suspense	196
Anger	194
Sad	193
Joy	191

performed and it was noted that the generated sentences adhered to typical Hindi linguistic patterns, including the exclusive use of Hindi characters and sentence lengths ranging between 3 to 22 words. Additionally, a statistical comparison was performed between the accuracy and F1 scores of models trained on the original and generated datasets. The models exhibited comparable performance on both datasets, further validating the representativeness and effectiveness of the generated data in capturing the emotional nuances of the original dataset. To further validate the quality of the generated labels, a manual annotation check was conducted on a sample of 100 records. Two native Hindi speakers independently annotated these records and compared their labels with those generated by the model. The results showed that the model’s labels were consistent with human annotations, reinforcing the reliability of the generated dataset for emotion detection tasks.

4.2 Fine-Tuning BERT and mBERT on Balanced Dataset Generated Using GPT-3.5-turbo

Fine-tuning refers to the process of adjusting the pre-trained parameters of the models to better suit the specific task of emotion detection in Hindi text. By leveraging the high-quality data generated in Sect. 4.1, the BERT and mBERT models were trained to capture the nuances of emotional expression in Hindi text. Fine-tuning these models allows them to learn from the newly acquired dataset and adapt their parameters to effectively classify text inputs into the predefined emotional categories. This step enhances the model’s ability to understand and interpret emotional nuances in the Hindi language, thereby improving their performance in emotion detection tasks. 70% of the data was used for training and 30% for testing.

These models were implemented using PyTorch [15] and the HuggingFace transformers library [23]. The training was done in a PyTorch environment with no GPU support. Based on hyperparameter tuning, the learning rate of $2e-5$ and number of epochs as 10 has been selected as best parameters for both BERT and mBERT. ‘bert-base-uncased’ [9] and ‘bert-base-multilingual-cased’ [8] are the models used from the Hugging Face transformers library for BERT and mBERT respectively. The other hyperparameters have been discussed previously in Sect. 3.3. BertTokenizer.from_pretrained() method is used for tokenizing the text and BertForSequenceClassification.from_pretrained() is used for classification. The number of labels parameter has been set as 5 to identify the 5 emotions – Anger, Joy, Sad, Suspense and Neutral.

4.3 Evaluating the Performance of Balanced Dataset on Fine-Tuned BERT and mBERT

Table 3 depicts the results for the GPT generated balanced dataset on fine-tuned BERT and mBERT. For the BERT model, training significantly improves performance compared to baseline model, with accuracy increasing from 0.255 to 0.779. However, training further enhances mBERT’s performance, with accuracy reaching 0.85 from 0.197 in baseline model. mBERT exhibits higher accuracy, with training compared to BERT. F1 score also follow a similar trend as accuracy. Overall, fine-tuning significantly improves the performance of both BERT and mBERT models for emotion detection on the GPT-generated dataset. Since mBERT is trained on Hindi, it is showing us improved accuracy and F1 score.

Table 3. Results of fine-tuning BERT and mBERT

Model	Training Type	Accuracy	F1 Score
BERT	Baseline	0.255	0.149
BERT	Fine-Tuned	0.779	0.782
mBERT	Baseline	0.197	0.119
mBERT	Fine-Tuned	0.850	0.857

4.4 Generation of Balanced Dataset Using Emo-Dis-Hi Dataset with Few-Shot Learning and Zero-Shot Learning Using GPT-3.5-turbo API

Furthermore, we used the fine-tuned BERT and mBERT models on a new Hindi dataset Emo-Dis-Hi. This dataset has 9 emotion categories sadness, sympathy/pensiveness, optimism, fear/anxiety, joy, disgust, anger, surprise, and no emotion. However, BERT and mBERT have been pre-trained to classify 5 emotions – joy, sad, anger, suspense and no emotion. Hence the Emo-Dis-Hi dataset was filtered to contain only these 5 emotion categories. There were 2360 records spanning these categories (Table 4). Table 5 highlights the results of the evaluation.

From the low F1 score and dataset being highly imbalanced, we can note that it is just learning the dominant class. Therefore, there is a need to balance the dataset. We will use GPT-3.5-turbo API to generate the dataset using few-shot learning for 4 emotion categories – joy, sad, anger and no emotion and zero shot learning for suspense. 10 examples spanning across 4 categories were given for few-shot learning. 941 records were generated across 5 categories (Table 4).

Table 4. Emotion Categories in Emo-Dis-Hi dataset

Sentiment	Count in original dataset	Count in GPT generated dataset
Neutral	273	197
Suspense	0	180
Anger	58	195
Sad	1849	193
Joy	180	176

Table 5. Results of evaluating Emo-Dis-Hi on fine-tuned BERT

Model	Training Type	Accuracy	F1 Score
Fine-tuned BERT on Bhaav balanced dataset	Baseline	0.079	0.155
Fine-tuned BERT on Bhaav balanced dataset	Few-shot learning (10% Emo-Dis-Hi dataset)	0.784	0.322

4.5 Evaluation of Balanced Emo-Dis-Hi Dataset on Fine-Tuned BERT and mBERT

Table 6 highlights the results for the GPT generated balanced dataset using Emo-Dis-Hi on fine-tuned BERT and mBERT. Steps similar to Bhaav dataset as highlighted in Sect. 4.1 were taken to validate the quality and consistency of the generated data.

Through this step we wanted to show that BERT and mBERT which have been fine-tuned on another dataset can perform better than the baseline BERT and mBERT models. The models fine-tuned on another dataset show much better performance than baseline and promising results with few-shot and further fine-tuning.

We can see an increase of 12.8 percentage points in accuracy between BERT baseline and BERT trained on another dataset. There is a further increase of 26 percentage points with few-shot learning and increase of 59.2 percentage points with fine-tuning when compared with BERT trained on another dataset. Similar trends are observed for F1 score as well. mBERT also shows similar results to BERT in both F1 score and accuracy.

The best performance for Emo-Dis-Hi GPT generated balanced dataset is by further fine-tuning the fine-tuned BERT. The model achieves an accuracy of 90.5% and F1 score of 90.7%.

Table 6. Results of evaluating GPT generated Emo-Dis-Hi on BERT and mBERT

Model	Training Type	Accuracy	F1 Score
BERT	Baseline	0.185	0.062
BERT	Fine-tuned BERT on Bhaav balanced dataset	0.313	0.285
Fine-tuned BERT on Bhaav balanced dataset	Few-shot learning (10% Emo-Dis-Hi balanced dataset)	0.573	0.579
Fine-tuned BERT on Bhaav balanced dataset	Fine-tuning (80% Emo-Dis-Hi balanced dataset)	0.905	0.907
mBERT	Baseline	0.217	0.126
mBERT	Fine-tuned mBERT on Bhaav balanced dataset	0.290	0.281
Fine-tuned mBERT on Bhaav balanced dataset	Few-shot learning (10% Emo-Dis-Hi balanced dataset)	0.612	0.650
Fine-tuned mBERT on Bhaav balanced dataset	Fine-tuning (80% Emo-Dis-Hi balanced dataset)	0.900	0.904

5 Conclusion, Future Work and Ethical Issues

In this research study we have successfully created two novel balanced datasets for the Bhaav and Emo-Dis-Hi datasets for emotion detection in Hindi. We further demonstrated that by fine-tuning the BERT and mBERT models on these datasets there is an increase from 25.5% to 77.9% in accuracy for BERT and from 19.7% to 85% in accuracy for mBERT. F1 score also followed a similar trend.

The research was further extended to evaluate the performance of the fine-tuned models on an additional Hindi dataset generated from GPT 3.5 using existing dataset. We saw some interesting results, i.e. the models fine-tuned on another dataset show much better performance than baseline. Further few-shot learning and fine-tuning was performed on fine-tuned models and an increase in accuracy to 90% was observed for both BERT and mBERT models.

Overall, the findings emphasize the potential of integrating state-of-the-art large language models (LLM) with dataset augmentation and fine-tuning strategies to advance the field of emotion detection in Hindi text. By addressing the key challenges like dataset imbalance and linguistic nuances the research contributes valuable insights and methodologies towards the development of robust emotion detection systems tailored to the Hindi language domain.

Future work for this research can be expanded in multiple areas like multimodal emotion detection by combining visual and textual cues, domain specific emotion detection for areas like customer service, education and healthcare. We can also extend this research to study cross-lingual emotion detection to create more generalizable emotion detection systems.

Ethical considerations should ensure mitigating biases in models, and transparently disclosing the limitations and potential societal impacts of the research findings. According to research study by Yoo et al. [26] LLMs are prone to generate toxic content and exhibit social biases. Therefore, the records generated using prompting based approach need to be considered carefully. These ethical concerns can be addressed by human inspection of generated results or debiasing the language model before using it for generation.

References

1. Ahmad, Z., Jindal, R., Ekbal, A., Bhattacharyya, P.: Borrow from rich cousin: transfer learning for emotion detection using cross lingual embedding. *Expert Syst. Appl.* **139**, 112851 (2020)
2. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: pre-training of deep bidirectional transformers for language understanding (2018). arXiv preprint [arXiv:1810.04805](https://arxiv.org/abs/1810.04805)
3. Devlin, J.: Bert/multilingual at google-research/bert. GitHub (2019). <https://github.com/google-research/bert/blob/master/multilingual.md>. Accessed 1 Mar 2024
4. Dyvik, E.H. 2023. The most spoken languages worldwide 2023. Statista. [Online]. Available at: <https://www.statista.com/statistics/266808/the-most-spoken-languages-worldwide/>. Accessed 1 Mar 2024
5. Ekbal, A., Bhattacharyya, P., Saha, T., Kumar, A., Srivastava, S.: HindiMD: a multi-domain corpora for low-resource sentiment analysis. In: *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pp. 7061–7070 (2022)
6. Garg, K., Lobiyal, D.K.: Hindi EmotionNet: a scalable emotion lexicon for sentiment classification of Hindi text. *ACM Trans. Asian Low-Resource Lang. Inf. Process. (TALLIP)* **19**(4), 1–35 (2020)
7. Harikrishna, D.M., Rao, K.S.: Emotion-specific features for classifying emotions in story text. In: *2016 Twenty Second National Conference on Communication (NCC)*, pp. 1–4. IEEE (2016)
8. Hugging Face. 2024. google-bert/bert-base-multilingual-cased. <https://huggingface.co/google-bert/bert-base-multilingual-cased>. Accessed 1 Mar 2024
9. Hugging Face. 2024. google-bert/bert-base-uncased. <https://huggingface.co/google-bert/bert-base-uncased>. Accessed 1 Mar 2024
10. Kemp, S.: Digital 2023: India (2023). <https://datareportal.com/reports/digital-2023-india>. Accessed 1 Mar 2024
11. Kumar, Y., Mahata, D., Aggarwal, S., Chugh, A., Maheshwari, R., Shah, R.R.: Bhaav-a text corpus for emotion analysis from Hindi stories (2019). arXiv preprint [arXiv:1910.04073](https://arxiv.org/abs/1910.04073)
12. Kumar, T., Mahrishi, M., Sharma, G.: Emotion recognition in Hindi text using multilingual BERT transformer. *Multimed. Tools Appl.* 1–22 (2023)
13. Nandwani, P., Verma, R.: A review on sentiment analysis and emotion detection from text. *Soc. Netw. Anal. Min.* **11**(1), 81 (2021)
14. OpenAI. 2024. Text Generation. OpenAI Platform Documentation. <https://platform.openai.com/docs/guides/text-generation>. Accessed 1 Mar 2024
15. Paszke, A., et al.: Pytorch: an imperative style, high-performance deep learning library. *Adv. Neural Inf. Process. Syst.* **32** (2019)
16. Phani, S., Lahiri, S., Biswas, A.: Sentiment analysis of tweets in three Indian languages. In: *Proceedings of the 6th Workshop on South and Southeast Asian Natural Language Processing (WSSANLP2016)*, pp. 93–102 (2016)

17. Sahu, G., Rodriguez, P., Laradji, I.H., Atighehchian, P., Vazquez, D., Bahdanau, D.: Data augmentation for intent classification with off-the-shelf large language models, 2022. arXiv preprint [arXiv:2204.01959](https://arxiv.org/abs/2204.01959)
18. Sasidhar, T.T., Premjith, B., Soman, K.P.: Emotion detection in Hinglish (Hindi+ English) code-mixed social media text. *Procedia Comput. Sci.* **171**, 1346–1352 (2020)
19. Singh, G.V., Priya, P., Firdaus, M., Ekbal, A., Bhattacharyya, P.: EmoInHindi: a multi-label emotion and intensity annotated dataset in Hindi for emotion recognition in dialogues (2022). arXiv preprint [arXiv:2205.13908](https://arxiv.org/abs/2205.13908)
20. Venkatakrishnan, R., Goodarzi, M., Canbaz, M.A.: Exploring large language models' emotion detection abilities: use cases from the middle east. In: 2023 IEEE Conference on Artificial Intelligence (CAI), pp. 241–24. IEEE (2023)
21. Wang, Y., Yao, Q., Kwok, J.T., Ni, L.M.: Generalizing from a few examples: a survey on few-shot learning. *ACM Comput. Surv. (CSUR)* **53**(3), 1–34 (2020)
22. Wikipedia contributors. 2024. Hindi. Wikipedia. <https://en.wikipedia.org/wiki/Hindi>. Accessed 1 Mar 2024
23. Wolf, T., et al.: Huggingface's transformers: State-of-the-art natural language processing, 2019. arXiv preprint [arXiv:1910.03771](https://arxiv.org/abs/1910.03771)
24. Wong, B.: Top Social Media Statistics And Trends. *Forbes* (2024). <https://www.forbes.com/advisor/in/business/social-media-statistics/>. Accessed 1 Mar 2024
25. Worlddata.info. 2024. Hindi speaking countries. <https://www.worlddata.info/languages/hindi.php>. Accessed 1 Mar 2024
26. Yoo, K.M., Park, D., Kang, J., Lee, S.W., Park, W.: GPT3Mix: leveraging large-scale language models for text augmentation (2021). arXiv preprint [arXiv:2104.08826](https://arxiv.org/abs/2104.08826)



Classification and Recommendation of Mental Health Assistance Events Using an RNN-LSTM, Fast-And-Frugal Trees and Weighted Sum System

Nathan R. Dickson  and Nicholas H. M. Caldwell  

University of Suffolk, Ipswich, Suffolk IP4 1QJ, UK
n.caldwell@uos.ac.uk

Abstract. A common challenge for organizations providing vast ranges of services to large customer pools is linking customers only to relevant services, due to the sheer amount of service and customer information available. Using AI techniques, this project provided a linkage between the emotional needs of potential end users and events available through the education and community support charity Suffolk Libraries. The effective data classification and the implementation of a personalized recommendation algorithm allowed the project to connect the events and services offered to those members of the community who would benefit most from them.

Keywords: Recommendation Engine · Social Prescribing · Classifier

1 Introduction

The wider economic costs of mental illness in the UK have been estimated at £117.9 billion per year, or approximately 5% of UK GDP [1]. One in six adults in England have a common mental health disorder, 4.5 million were in contact with mental health services in 2021–22, 1.2 million were estimated to be on waiting lists for community-based mental health services at end of June 2022, 8 million people with mental health needs were estimated not in contact with mental health services as of 2021 [2]. Social prescribing connects people to non-medical sources of support or resources in the community that can assist with both mental health and well-being, typically through one or more of: social relationships, physical activity, awareness, learning, and giving / volunteering [3]. Social engagement in any group activity, whether formal or informal learning, leisure or social, has been shown to correlate with social capital as well as physical and mental health [4, 5]. As neutral community spaces, public libraries offer a range of social and health-related activities, promoting and enhancing the well-being of individuals in their community [6].

Commercial applications to support various elements of social prescribing exist – these can directly link to an individual’s medical record and provide recommendations

from a highly curated set of services and information sources in a specific local area, see Harrington and coworkers [7] for an overview of healthcare apps aimed at children and young people. Patel and colleagues undertook a questionnaire study of experts and members of the public into the opportunities and challenges of digital tools for social prescribing, with predicted advantages of increased access to services, but likewise cautions on data protection, confidentiality, ensuring genuine accessibility and avoidance of unintended consequences [8]. These issues have been considered in the development of this project.

The UK-based Suffolk Libraries project Discover More hopes to connect the libraries' services with community members who require additional support highlighted by the Emotional Needs Audit, developed by the mental health charity, Suffolk Mind. The Emotional Needs Audit is a short text-based questionnaire which identifies whether the respondent may have "emotional needs" such as the need for privacy, or achievement, or meaning and purpose, or attention, etc. The project can currently be considered as informal social prescribing where individuals, who may have undiagnosed or unaddressed milder mental health needs, can help themselves to improve their mental health and mental well-being by connecting to appropriate library services. Part of the University of Suffolk's contribution to the project was to develop a prototype system – and this paper covers the creation of the necessary algorithms and supporting code infrastructure. Suffolk Libraries needed a Natural Language Processor (NLP) to interpret human data, a mapping algorithm to tag the human-produced event descriptions, and a recommendation engine to connect users to events of potential interest.

A dataset of 14,000 events already existed in the Suffolk Libraries database which were stored in a Markdown format. Some events had been manually tagged at an earlier stage, although this mapping included only general types of events, activities, and services. It did not include tagging of the dataset with one or more of the twelve "emotional needs" used in the Suffolk Mind Emotional Needs Audit. The Suffolk Libraries and Suffolk Mind teams took a subset of the event dataset and manually tagged this subset ($n = 1322$) against the twelve emotional needs. This human-labelled data became the ground truth for prototype development. The tagged set was further divided into a training dataset and a test dataset, with a runtime 80:20 split ($n = 1058$ training, $n = 264$ test).

The team had three tasks. Firstly, the mapping of different types of activities needed to be refined and to be mapped to Suffolk Mind's Emotional Needs Audit's twelve emotional need categories. Secondly, NLP was to be used in the development of an automatic mapping program, also indicating levels of correlation (high, medium, low) to show the extent to which activities meet the various emotional needs. Thirdly, a recommendation engine evaluating users' unmet needs from the Emotional Needs Audit and recommending events with high correlation levels was required.

To fulfil Suffolk Libraries' needs for a robust demonstrator prototype that could be used as a sound foundation for future work and as a proof point to secure substantial external funding, mapping and recommendation algorithms were implemented and contextualized within a web-based prototype, where all the elements were hosted on a commercial cloud provider, and interaction with the system was through a secure custom Application Programmer's Interface (API) for algorithmic testing and a web interface

for human user testing. The related software development choices and approaches for the prototype will be described in Sect. 3.1. This represents the system context in which the different techniques would be compared against each other for accuracy as well as the development and deployment context for the final prototype.

2 Related Works

2.1 Natural Language Processors

Natural Language Processors (NLP) trade-off between varying levels of depth and complexity [9]. More fine-grain analysis methods can pick up on key individual details, while analyses of larger structures, such as sentences, can often pick up on texts' more pragmatic meaning. Although technically possible to produce an NLP format that considers text at a range of scales, this increases the compute power required to perform the training of the data [10, 11].

Neural network approaches are most suitable when large datasets and long training times are available, whereas probabilistic methods (such as Naïve Bayes [NB]) produce acceptable results faster and with smaller training datasets, though with a lesser degree of accuracy [11]. Both neural networks and NB were considered, as their key characteristics and advantages align with the project's scope.

NB is a text classification method that uses the concepts in Bayesian probability to evaluate the probability of strings of texts matching specific categories compared to a predefined corpus of training text (see [12] for a detailed explanation). NB is a highly effective method of text classification given its ability to deal with a relatively small training corpus and low CPU usage intensity benefits [13, 14] It does assume (unrealistically) that features are independent in each of their respective classes. NB's results' rely on prior probabilities within the dataset and so will produce almost completely random results when presented with novel data [15]. Because of these limitations, practical uses of NB are limited, but when suitable it remains a highly efficient method for getting accurate classification predictions [16].

A wide range of neural network (NN) designs and methods can be applied to build text classifiers. Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) are deemed particularly effective (RNN) with the different cell types in the RNN models allowing for increased accuracy [17].

Comparisons of CNNs and RNNs for a text classifier trained on prelabelled text-based radiology reports suggested feasibility of both, however RNNs were found to perform moderately better [18]. This conclusion is also reflected in various other studies comparing CNNs and RNNs, e.g. [19, 20].

RNN models are a type of artificial neural networks which feeds back past information into current iterations. This provides significant advantages for information where there are consequential connections between past and future events. For example, in English adjectives describe nouns, thus providing a link between words.

Bi-directional variations of cells consistently outperform their unidirectional counterparts [21]. That study also showed that increasing the number of hidden layers (up to a limit of 4 additional layers) within their models increased the accuracy of their results.

Further additional layers either did not substantially improve accuracy or, in some scenarios, the accuracy decreased. Similar results have been reported elsewhere e.g. [22, 23].

Multi-label classification (MLC) presents additional challenges over single-label classification, such as label interdependency, as highlighted by [24, 25], where the presence of specific labels affects the probability of other labels. Furthermore, MLC can be significantly affected by imbalances in the training data, which is often an inherent characteristic of many multi-label datasets [25, 26]. The imbalance ratio per label can be calculated through several methods, such as the Maximum Imbalance Ratio (MaxIR), the ratio of the most common label against the rarest one [26].

Several methods of label classification, such as Exact Match Ratio, can be adapted with difficulty for MLC by introducing concepts such as partial correctness [27]. Some methods exist for calculating the accuracy of multi-label classification, including ranking-based methods such as those highlighted by [27, 28]. Each method's usefulness depends on the application it is being used for, and how the testing data has been labelled. For instance, One-Error measures how often the top-ranked predicted label is outside the set of true labels. It does not require the test set to be labelled with rankings [28]. In contrast, for some applications, [27] highlights it is essential that all true labels be predicted even at the cost of a higher rate of false positives (e.g. disease detection). The Coverage method is an accuracy metric that evaluates how far, on average, a learning algorithm needs to go down in the ordered list of predictions to cover all the true labels [27]. For the Coverage method, the training data must contain labels in ranked order.

2.2 Recommendation Algorithms

A common method for making recommendations to users is Collaborative Filtering (CF), which has been successful in various industries, such as e-commerce [29] and streaming. CF functions by compiling datasets of different users' interests. When a new user joins, their interest is matched with the dataset of users with similar interests, so-called "neighbors", to produce a list of recommendations. CF is, however, not appropriate for this project for two reasons. Firstly, attendance data for the 14,000 existing events was not tracked, thus there was no pool of existing event and preference data with which a new user could be aligned. Secondly, the expected end users have diverse individual needs, including varying degrees of mental ill-health. A CF approach could make recommendations suitable for one user but not another.

An alternative is the Weighted Sum approach, which can be more programmatically defined and tailored to Suffolk Libraries' needs. It can be effective when greater control is needed for the algorithm and a selection of recommendations is needed [30]. The major downside is that it is slower than the CF approach.

As the recommendation algorithm relies on real-time recommendations, speed may become important. Fast and Frugal Heuristics Trees (FFHT) [31] allow for quick decision-making and can be applied to computational models, trading accuracy for speed. It can also be combined with the Weighted Sum approach giving greater control of Weighted Sum at a more efficient speed.

3 Development

3.1 Architectural Design and Development

Agile development was used to build a functional robust prototype, making use of early and continuous delivery of software to identify and resolve issues, and Continuous Integration and Continuous Delivery (CI/CD) pipelines used to deploy the software to the hosting servers. The architecture was chosen to reflect Suffolk Libraries' existing infrastructure, and follows high availability (HA), utilising a HA web server pair and a ternary setup for database availability. Figure 1 shows the architecture, including the web prototype, the mapping algorithm, and the recommendation engine.

Data was in JSON format, making it particularly suited to a NoSQL structure. For the database solution, the Atlas service by MongoDB was used, yielding significant security benefits (such as encryption at rest and in transit) and conformity to compliance standards such as GDPR. The cloud servers (on Digital Ocean) were fully hardened, following strong security principles.

To mitigate malicious third-party access to the system through GUI interface or API call, all requests required the user to authenticate using an individually generated API key. All requests into the application from the GUI or via API call were then subjected to validation checks to prevent injection attacks and return an error if they received an invalid response. The program checked both the validity of the JSON and the validity of the data type.

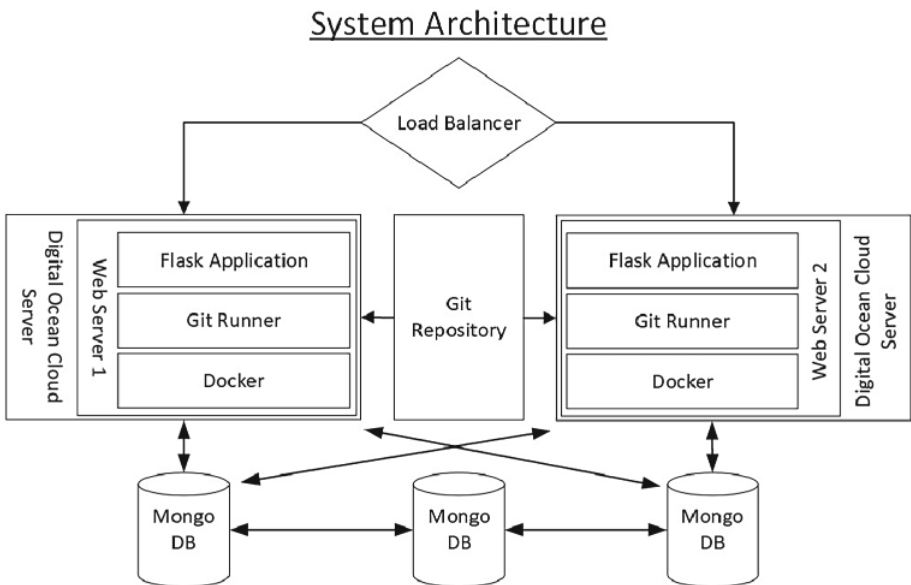


Fig. 1. System Architecture Diagram.

Flask was used due to its microservice architecture focusing on producing web-applications. Flask allowed for easy integration of API interfaces and enabled URL

routing which allowed differentiation of GUI and API services and for other systems to interact with the system smoothly. For the front-end, the system served Jinja2 HTML templates using SCSS compiled to CSS for styling and delivered responsive web pages suitable for a wide range of end user devices. The system was packaged into a Docker container, allowing the project to be quickly and efficiently deployed to various platforms and operating systems.

Comprehensive testing of the system was undertaken. This included accessibility testing to ensure that Suffolk Libraries maintained its commitment to inclusive services. Browser compatibility testing was conducted to ensure service accessibility on a large range of user platforms. Exploratory testing highlighted errors and bugs previously missed. Unit testing was applied throughout the code base to ensure the code worked as expected throughout – this included edge-case testing and false positive testing.

Beta tests were conducted by Suffolk Libraries throughout the development process. This rectified errors in the capturing of requirements. The library team assessed the proposed recommendations in terms of appropriateness for meeting the emotional needs of imagined test personae. Acceptance testing ensured that the completed program met Suffolk Libraries' expectations.

3.2 NLP Classifier Dataset

A human-labelled dataset of 1322 labelled events were used across all NLP methods to build the classifier. This bootstrap corpus data was divided into two sections via an 80:20 ($n = 1058:264$) runtime shuffled split: the "Training Data" and the "Testing Data".

Example labelled text data:

Event Title: "Open Space - wellbeing drop-in group
(Felixstowe group)"

Event Description: "Open Space drop-ins are informal meet-ups for anybody interested in their mental health and wellbeing, as well as their families and carers. Come along for support with mental health and well-being and to socialise with others in your local community."

Human Labels:

["EmotionalConnection", "Respect", "Community", "MeaningAndPurpose"]

A factor in the human labelling of events was the subjective and interdependent nature of the label categories corresponding to the twelve emotional needs. In general, labels that indicated how activities were being run (e.g., "Movement") were easier to assign consistently than labels that focused on the desired effect that activities were due to have on their participants (e.g., "EmotionalConnection"). Labelers' personal understanding of the meaning of a label thus influenced their categorization choices: The label of "Community" may for example be assigned to an event aimed at creating a sense of togetherness between participants, or alternatively it may involve activities that are set to benefit the general public (e.g. a litter-picking event). Within the total set of labelled training data, there was a natural imbalance in label frequency because this data originally

came from a preexisting repository of past events which did not have an even distribution of all event types (and consequently of labels). To counteract the potential bias this could introduce to the mapping algorithm, the algorithm was trained on a subset of this event data which sought to even out this imbalance by excluding some of those events which were tagged with the most frequently used labels. The Maximum Imbalance Ratio (MaxIR) of the final training data set was 417:241. It was not feasible to improve the Maximum Imbalance Ratio further as that would have reduced the total available training data subset to a size that would have been too small.

3.3 Traditional NLP Architecture

At the data cleaning process's first stage, all bootstrap corpus data was tokenized to produce smaller tokens of data. Due to the focus on texts' overall sentiment, the bootstrap corpus data was tokenized by full words [32]. The tokenized words were lemmatized, converting all words to their simple lemma. This process helped to remove much of the time-based context, for example, verbs' tenses that might have been captured in the corpus, as this data was not relevant to the events' categorization [33]. Many tokenized words were common words and unlikely to carry a large amount of contextual data about the events. These common "stop words" were removed from the dataset to focus the classifiers' weighting on less common words that contain more context [34]. One major factor affecting the Naïve Bayes approach in correctly identifying the probability of text being classified to a particular tag is the frequency of that word within the given text. Therefore, it was important not to overly prune the data to avoid skewing the proportions of keyword appearances. Although the word density of keywords was increased through pruning, the proportions of keywords within the text were intended to remain the same.

3.4 Neural Network NLP Architecture

The project used RNNs. The number of vectors in the proposed architecture were in a one-to-many relationship, with the input layer consisting of the event text and the output layer representing the twelve emotional needs. Three RNN cell types were considered: standard simple RNN cell (RNN), gated recurrent unit (GRU), and Long short-term memory unit (LSTM). Uni-directional and bi-directional versions of the three cell types were produced to compare the results. Similar to the process for the NB approach, the NN version's focus remained on the text's overall context in the corpus and therefore the text was tokenized by full words.

The RNN build process consumed a corpus of labelled past events run by Suffolk Libraries and trained the model on these records. The RNN took in the pre-trained model combined with the new event's text. It then produced predicted results in the form of three lists, indicating a high, medium, or low correlation with the emotional needs. Figure 2 illustrates the recommendation engine's process, including a Fast and Frugal Heuristic Tree (FFHT) and a Weighted Sum algorithm.

The FFHT was used instead of a database query because the emphasis in this particular use case was on never returning a "NULL" or "No results found" response to vulnerable users. Such a response could indicate that no suitable event or activity is

available, and thus give the impression to a vulnerable user that no help is available, which could exacerbate or even cause negative feelings in the user.

Whilst an initial set of weights was used during the test phases of this project, this was updated following a survey of users having trialed the prototype application. However, the Weighted Sum algorithm was built to be programmable and so the final weights in the live version will be determined by Suffolk Libraries and Suffolk Mind.

Recommendation Engine Flowchart

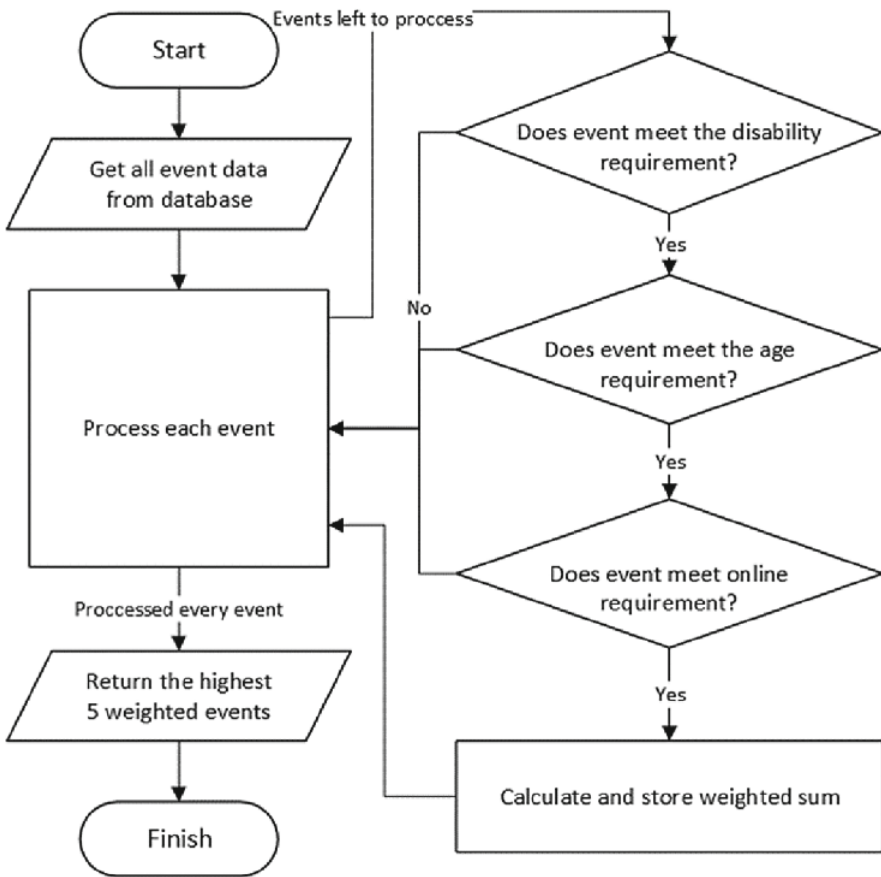


Fig. 2. Recommendation Engine Flowchart (Source: Personal Collection).

4 Results and Discussion

This project's results were collected from the NLP algorithms using the same corpus of bootstrap data provided by the Suffolk Libraries team. The evaluation metric for classification accuracy was calculated using the One-Error method converted to a percentage. Each algorithm was run for a series of iterations to show the algorithm's performance with increasing iterations on the training data. For the NNs, these iterations were in the form of standard epochs while for the NB method, multiple duplicates of the training data were run. Running NB through multiple "iterations" is unusual but was undertaken as an experiment to determine if it could improve the classification accuracy.

This process of testing the NLP algorithms was performed ten times for each algorithm. The results of this process shown below were then rounded to the nearest whole integer, reflecting both the variations between runs often being greater than a whole per cent, and to account for inaccuracies in the testing process (Table 1).

Table 1. Results (% accuracy and representative absolute numbers accurate to nearest whole number) from NLP Classification. Test set n = 264.

ITERATIONS	300	600	1000	10000
NB	75% (198)	81% (214)	82% (217)	82% (217)
RNN (RNN-Cell)	42% (111)	63% (167)	70% (185)	71% (188)
Bi-RNN (RNN-Cell)	43% (113)	65% (172)	72% (191)	72% (191)
RNN (GRU)	52% (138)	73% (193)	80% (212)	81% (214)
Bi-RNN (GRU)	56% (148)	75% (198)	84% (222)	83% (220)
RNN (LSTM Cell)	55% (145)	75% (198)	83% (220)	84% (222)
Bi-RNN (LSTM Cell)	57% (150)	78% (206)	85% (225)	87% (230)

Due to the nature of both NB and NN, some runs were expected to return below-average results, which could not accurately reflect the algorithm's performance. There were 2 runs out of 280 were discounted for being more than 2 standard deviations from the mean, representing 0.71% of the runs, namely Naïve Bayes iteration set 1000 run 7 (result 61%, 161) and RNN (LSTM Cell) iteration set 300, run 2 (result 20%, 53).

The first striking difference between NB and the NNs is their performance at low iteration counts, where NB greatly outperformed all NN algorithms. As iterations increase, however, NB was eventually overtaken in accuracy by some – but not all – NN algorithms. This is largely due to the yield curve for the NB method which shows that it does not benefit greatly from more iterations. While the NN algorithms benefitted more from increasing iterations, they also suffered from diminishing returns with higher counts. For example, moving from 300 to 600 iterations produced a mean yield of 21% increase in accuracy for the NN algorithms, while moving from 1,000 to 10,000 iterations only yielded a 1% increase.

When comparing NN algorithms, it is clear that in this project, bi-directional NNs outperformed uni-directional NNs by a mean average of 2%. One notable exception to

this was the transition on the Bi-RNN (GRU) from 1,000 to 10,000 iterations where the model decreased in accuracy. This is due to the model being trained to overfit the data [35]. This result is supported by other literature in this field that suggests that bi-directional NNs are likely to perform better due to meaning in human speech being conveyed through clusters of words, rather than by words in isolation from one another [36]. Therefore, algorithms that analyze a word in the context of the words that precede and follow it are better suited for determining the text's tone and meaning.

Among the variations in the RNN cells in this project, the LSTM cell outperformed the GRU and standard RNN cells. The difference between the standard cell compared to the LSTM and GRU cells was more pronounced while the LSTM and GRU cells performed more similarly. It is noticeable that the standard uni- and bi-directional RNN cells did not outperform the Naïve Bayes algorithm even at the highest tested iterations. The literature suggests that the simpler GRU cells have a computational speed advantage over LSTM cells and are perhaps better suited to lower complexity data sequences than LSTM (see e.g. [37, 38]. For the Suffolk Libraries dataset, the automated tagging of new events is undertaken separately to recommendations to end users, so the speed saving is less important than the slightly more accurate results available via LSTM.

Based on the experiments undertaken for this dataset and this problem, Naïve Bayes and bi-directional RNN (LSTM) are the best candidate solutions. The Naïve Bayes approach provides a solid solution that can be retrained more readily than the neural network, so gave the project a fallback implementation for handling unexpected dataset evolution. The higher accuracy of the bidirectional RNN (LSTM) made it the approach of choice as providing better matches of events to end users is important – while an event may be free financially (in terms of admission), there are still opportunity costs and logistical costs (travel time, travel cost, etc.) that are much more meaningful than say an inaccurate recommendation of a programme on a streaming service (where a programme can be prematurely exited at trailer or in first few minutes of viewing). Given that some of the eventual end users will have lower mental well-being or mental health disorders, a useful recommendation is more likely to maintain their engagement with the system and the process of finding and participating in appropriate events.

5 Conclusions and Future Work

This project successfully delivered a prototype encompassing NLP and mapping algorithms for labelling events organized by Suffolk Libraries, as well as a recommendation engine linking users to events based on each user's individual emotional needs. The system has been designed to support recommendation of activities from a dynamically changing set of activities rather than a tightly curated set of services.

The prototype demonstrates a novel contribution through its combination of robust techniques – an established classification algorithm applied to a bespoke dataset and data structure, and the blend of fast-and-frugal-heuristic trees with a weighted sum approach in the recommendation algorithm. Although the RNN model has high accuracy, it requires a potentially long rebuild phase on different training data or changed parameters relative to NB and similar techniques. The key limitations here concern the emotional needs, as any changes there will necessitate a complete model rebuild, and the event data.

Significantly different new types of event data will require retraining of the model. The historical event data was also problematic as there was no standardized format in either space or time – different libraries within the Suffolk Libraries network recorded events in different ways and the event descriptions also evolved over time. The recommendation algorithm can provide more personalized results, which is important for prospective end users, than techniques such as collaborative filtering. Unlike collaborative filtering, the recommendation algorithm cannot learn from the user base – however personalization of results is arguably more appropriate to addressing individual needs and more mindful of user privacy and mental well-being.

Different performance-influencing approaches could be examined to improve event classification accuracy. For example, this project centered around NN prototypes with similar structures and a set number of layers. Varying the number of layers could potentially yield better results and would be worth testing. Furthermore, this project did not conduct any runtime analysis for the time different NLP algorithm models took to train and then to produce results. This analysis will become relevant when differentiating between methods which produced similar results in terms of classification accuracy, as this may reduce the overall compute power required.

Ultimately, this project delivered a fit-for purpose prototype which can help Suffolk Libraries to connect their community members requiring additional support to events meeting their emotional needs. The project has now been taken forward commercially and the final system will be used by members of the public on a large scale across the full set of 45 libraries in the county of Suffolk. Formal evaluation will determine whether library activities recommended to actual end users deliver statistically significant improvements in mental well-being. If so, this will position the system for formal social prescribing by clinicians, and much wider use.

Acknowledgments. This project was funded by Suffolk Libraries (HMRC charity number XT34476. Registered company number IP031542) and the European Regional Development Fund as a KEEP+ Research and Innovation Collaboration (project number 539).

Disclosure of Interests. Nicholas HM Caldwell declares no conflicts of interest. Nathan Dickson received a part-time salary for this work from Suffolk Libraries.

References

1. McDaid, D., et al.: The economic case for investing in the prevention of mental health conditions in the UK. London School of Economics and Mental Health (2022). <https://www.mentalhealth.org.uk/explore-mental-health/publications/economic-case-investing-prevention-mental-health-conditions-UK>
2. Hyde, J., et al.: Progress in improving mental health services in England. National Audit Office UK (2023). ISBN: 978-1-78604-472-3, <https://www.nao.org.uk/reports/progress-in-improving-mental-health-services-in-england/>
3. Teuton, J.: Social prescribing for mental health: background paper. NHS Health Scotland (2015). <https://www.healthscotland.scot/media/2067/social-prescribing-for-mental-health-background-paper.pdf>

4. Aguilar, J.P., Sen, S.: Comparing conceptualizations of social capital. *J. Community Pract.* **17**(4), 424–443 (2009)
5. Herzog, A.R., Ofstedal, M.B., Wheeler, L.M.: Social engagement and its relationship to health. *Clin. Geriatr. Med.* **18**, 595–609 (2002)
6. Kelley, A., Riggelman, K., Clara, I., Navarro, A.E.: Determining the need for social work practice in a public library. *J. Community Pract.* **25**(1), 112–125 (2017)
7. Harrington, R.A., Gray, M., Jani, A.: Digitally enabled social prescriptions: adaptive interventions to promote health in children and young people. *J. R. Soc. Med.* **113**(7), 270–273 (2020). <https://doi.org/10.1177/0141076819890548>
8. Patel S., Craigen, G., Pinto da Costa, M., Inkster, B.: Opportunities and challenges for digital social prescribing in mental health: questionnaire study. *J. Med. Internet Res.* **23**(3), e17438 (2021). <https://doi.org/10.2196/17438>
9. Reshamwala, A., Mishra, D., Pawar, P.: Review on natural language processing. *IRACST – Eng. Sci. Technol. Int. J. (ESTIJ)* **3**(1), 113–6 (2013)
10. Chowdhury, G.G.: Natural language processing. *Annu. Rev. Inf. Sci.* **37**(1), 51–89 (2005). <https://doi.org/10.1002/aris.1440370103>
11. Indurkha, N., Damerau, F.J.: *Handbook of Natural Language Processing*. CRC Press, Taylor & Francis Group, Boca Raton, FL (2010)
12. Marquez, L.: Machine Learning and Natural Language Processing. In: *Curso de Industrias de la Lengua: La ingeniería Lingüística en la Sociedad de la Información: 1–53*. Universitat Politècnica de Catalunya, Barcelona (2000). <https://upcommons.upc.edu/handle/2117/96428>. Accessed 15 May 2014
13. Mooney, R.J.: Comparative experiments on disambiguating word senses: an illustration of the role of bias in machine learning. In: *Conference on Empirical Methods in Natural Language Processing*, pp. 17–18. University of Pennsylvania, Philadelphia (1996). <https://aclanthology.org/W96-0208/>
14. Rish, I.: An empirical study of the naive Bayes classifier. In: *IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence*, pp. 41–6. IBM, New York (2001)
15. Islam, M.J., Wu, Q.M.J., Ahmadi, M., Sid-Ahmed, M.A.: Investigating the performance of I-Bayes classifiers and K-nearest neighbor classifiers. In: *2007 International Conference on Convergence Information Technology (ICCIT 2007)*, pp. 1541–46. IEEE (2007). <https://doi.org/10.1109/ICCIT.2007.148>
16. Granik, M., Mesyura, V.: Fake news detection using naive Bayes classifier. In: *2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON)*, pp. 900–03. IEEE (2017). <https://doi.org/10.1109/UKRCON.2017.8100379>
17. Minaee, S., et al.: Deep learning–based text classification. *ACM Comput. Surv.* **54**(3), 1–40 (2021). <https://doi.org/10.1145/3439726>
18. Banerjee, I., et al.: Comparative effectiveness of convolutional neural network (CNN) and recurrent neural network (RNN) architectures for radiology text report classification. *Artif. Intell. Med.* **97**, 79–88 (2019). <https://doi.org/10.1016/j.artmed.2018.11.004>
19. Lee, J.Y., Deroncourt, F.: Sequential short-text classification with recurrent and convolutional neural network. In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 515–20. Association for Computational Linguistics. (2016). <https://doi.org/10.18653/v1/N16-1062>
20. Yin, W., Kann, K., Yu, M., Schütze, H.: Comparative study of CNN and RNN for natural language processing. (2017). <https://arxiv.org/abs/1702.01923>
21. Ragheb, A.N., Gody, A., Said, T.: Comparative study of different types of RNN in speech classification. *Egypt. J. Lang. Eng.* **8**(1), 1–16 (2021). <https://doi.org/10.21608/ejle.2021.45203.1014>

22. Surkan, A.J., Singleton, J.C.: Neural networks for bond rating improved by multiple hidden layers. In: 1990 IJCNN International Joint Conference on Neural Networks (Volume 2), pp. 157–62. IEEE (1990). <https://doi.org/10.1109/IJCNN.1990.137709>
23. Allen-Zhu, Z., Li, Y., Liang, Y.: Learning and generalization in overparameterized neural networks, going beyond two layers. In: 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), pp. 1–12 (2019). <https://papers.nips.cc/paper/2019/file/62dad6e273d32235ae02b7d321578ee8-Paper.pdf>
24. Chekina, L., Gutfreund, D., Kontorovich, A., Rokach, L., Shapira, B.: Exploiting label dependencies for improved sample complexity. *Mach. Learn.* **91**, 1–42 (2013). <https://doi.org/10.1007/s10994-012-5312-9>
25. Liu, W., Wang, H., Shen, X., Tsang, I.W.: The emerging trends of multi-label learning. *IEEE Trans. Pattern Anal.* **44**, 7955–7974 (2022). <https://doi.org/10.1109/TPAMI.2021.3119334>
26. Tarekegn, A.N., Giacobini, M., Michalak, K.: A review of methods for imbalanced multi-label classification. *Pattern Recogn.* **118**, 107965 (2021). <https://doi.org/10.1016/j.patcog.2021.107965>
27. Sorower, M.: A literature survey on algorithms for multi-label learning. Oregon State University, Corvallis, Oregon, Technical Report (2010). https://www.researchgate.net/publication/266888594_A_Literature_Survey_on_Algorithms_for_Multi-label_Learning
28. Nasierding, G., Kouzani, A.Z.: Comparative evaluation of multi-label classification methods. In: 2012 9th International Conference on Fuzzy Systems and Knowledge Discovery, pp. 679–683. IEEE (2012). <https://doi.org/10.1109/FSKD.2012.6234347>
29. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Analysis of recommendation algorithms for e-commerce. In: EC '00: Proceedings of the 2nd ACM Conference on Electronic Commerce, pp. 158–67. ACM Press, New York (2000). <https://doi.org/10.1145/352871.352887>
30. Tang, J., Wang, K.: Personalized Top-N sequential recommendation via convolutional sequence embedding. In: WSDM '18: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, pp. 656–73. ACM Press, New York (2018). <https://doi.org/10.1145/3159652.3159656>
31. Gigerenzer, G., Todd, P.M.: Fast and frugal heuristics: the adaptive toolbox. In: Gigerenzer, G., Todd, P.M., ABC Research Group (eds.), *Simple Heuristics That Make Us Smart*, pp. 3–34. Oxford University Press, Oxford (1999)
32. Grefenstette, G.: Tokenization. In: van Halteren, H. (ed.) *Syntactic Worldclass Tagging: Text, Speech and Language Technology*, pp. 117–33. Springer, Dordrecht (1999). https://doi.org/10.1007/978-94-015-9273-4_9
33. Plisson, J., Lavrač, N., Mladenčić, D.: A Rule based approach to word lemmatization. In: Proceedings of the 2004 International Symposium on Information and Communication Technologies, pp. 83–86 (2004). <https://www.semanticscholar.org/paper/A-Rule-based-Approach-to-Word-Lemmatization-Plisson-Lavrac/5319539616e81b02637b1bf90fb667ca2066cf14>
34. Chandra, N., Khatri, S.K., Som, S.: Natural language processing approach to identify analogous data in offline data repository. In: Kapur, P., Klochkov, Y., Verma, A., Singh, G. (eds.) *System Performance and Management Analytics*. Asset Analytics, pp. 65–76. Springer, Singapore (2019). https://doi.org/10.1007/978-981-10-7323-6_6
35. Srivastava, N., et al.: Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **15**(1):1929–58 (2014). <https://jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf>
36. Song, M., Zhao, X., Liu, Y., Zhao, Z.: Text sentiment analysis based on convolutional neural network and bidirectional LSTM model. In: International Conference of Pioneering Computer Scientists, Engineers and Educators, pp. 55–68 (2018). https://doi.org/10.1007/978-981-13-2206-8_6

37. Zarzycki, K., Ławryńczuk, M.: Advanced predictive control for GRU and LSTM networks. *Inf. Sci.* **616**, 229–254 (2022). <https://doi.org/10.1016/j.ins.2022.10.078>
38. Cahuantzi, R., Chen, X., Güttel, S.: A Comparison of LSTM and GRU networks for learning symbolic sequences. In: Arai, K. (ed.) *Intelligent Computing. SAI 2023. LNNS*, vol. 739, pp. 771–785. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-37963-5_53



Digit Detection: Localizing and Convoluting

Tyrell Martens and John Z. Zhang^(✉)

Department of Math and Computer Science, University of Lethbridge,
Lethbridge, AB T1K 3M4, Canada
{tyrell.martens, john.zhang}@uleth.ca

Abstract. We explore the development of a set of algorithms for accurately localizing and classifying handwritten digits, addressing the challenges posed by variations in individual writing styles, digit sizes, and the presence of multiple digits in an image. Our work is structured into three tasks, each building upon the previous ones. *Task A*, which is trivial, focuses on classifying individual handwritten digits with 28×28 pixel resolution using a *convolutional neural network*. *Task B* extends the CNN in Task A, as a black box, by developing algorithms to recognize multiple digits with identical dimension sizes, placed on a large image. *Task C* further complicates Task B by considering digits with varying dimension sizes in a large image. Our proposed approach involves the use of convolution operations for digit localization, and takes advantage of inspirations from existing algorithms for handling varying digit dimension sizes. Experimental results demonstrate the effectiveness of our approach in achieving high accuracy rates. Our work contributes to the advancement of robust algorithms capable of accurately classifying handwritten digits in different scenarios and is expected to be applicable to classifying other handwritten objects, such as handwritten letters.

Keywords: Digit detection and classification · Convolutional neural networks · Detecting digits under different circumstances

1 Introduction

The concept of mimicking human cognition to solve a particular problem has been studied since the 1950s [3]. *Artificial intelligence (AI)* has seen constant evolution over the last several decades. With its rapid progress, a complex, more specialized field of artificial intelligence came into existence: *machine learning* [21]. It is a process in which a computer has the ability to solve a variety of problems from the observations and manipulations of data [21]. Machine learning has been the most common area in AI for many years [21], with one exciting methodology that has emerged: *deep-learning artificial neural networks*. Briefly, a neural network is composed of an input layer, on which there is a set of *input units*, each connecting to a *feature*, a set of middle hidden layers, each of which is

composed of a set of *hidden units*, and an output layer, again consisting of a set of *output units*. On the connection between two neurons on adjacent layers, there is a weight that represents how significant the connection is, therefore impacting the next layer. Generally speaking, a neural network computes a function given the input features, using the weights as intermediate parameters, by *forward propagating* the computed values from the input units to the units in the first hidden layer, to the units in the next hidden layer, and eventually to the units in the output layer. Learning occurs by adjusting the weights attached to the connections between two units on adjacent layers. The goal is to minimize the difference between the calculated output and the expected output of the function. The adjustments of the weights are done through *backward propagation*, from the output units through the hidden layers to the input units with complex mathematical algorithms [1]. A simple neural network is shown in Fig. 1.

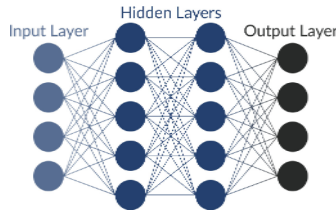


Fig. 1. A simple neural network.

One common prediction task of neural networks is classification, which involves predicting the correct label for a given piece of input data. Such an example is document processing, where handwritten digits and characters must be classified [25]. The problem of classifying handwritten digit has been traditionally and incredibly popular when it comes to introducing and discussing various classification tasks. Large strides have been made when deep learning is utilized, more specifically when a *convolutional neural network* is made use of [12]. In this work, we explore how deep learning with CNN, in abreast with the adaptation of some conventional algorithms, can be used to conduct digit detection.

2 Related Works

For centuries the convolution operation has been essential for mathematicians, physicists, and engineers. It has been applied to many aspects of our world with image convolution being extremely popular for image-processing algorithms. Image convolution is equivalent to computing the dot-wise product of a matrix (known as the filter or *kernel*) with a selected area of pixel values from an image. This filter is applied from left-to-right, top-to-bottom across all pixels to produce a new convolved image which can be analyzed to determine key features

and patterns within the original image [7, 15]. The *stride* of a convolution operation is the distance that the kernel “jumps” when being applied left-to-right and top-to-bottom [7]. The dimensions of the convolved image will be not reduced if the input image is *padded* with values of zero. *Convolutional neural networks* (CNN) are a specialized form of neural networks that conduct *convolutional operations* [6]. CNNs contain more layers than the other neural networks, often changing between *convolution layers*, *ReLU layers* and *pooling layers*, with *fully connected layers* at the end of the network [6]. ReLU layers apply a ReLU activation function across all units, mapping negative neurons to zero and maintaining positive neurons at their current values [6]. Pooling layers compute the average or select the maximum value for a certain section of units [6]. CNNs work exceptionally well for grid-structured inputs, such as images.

The classification of handwritten digits using CNN has been explored [12] as they perform exceptionally well on the spatial structure of digits. In [19], Patrice *et al.* establish that CNNs often achieve the greatest performance compared to, for example, *support vector machines*, in handwriting recognition tasks [19]. Han *et al.* [24] explore the performance of different network architectures, concluding that traditional neural networks result in better network performance than *radial basis function networks* and *counterpropagation networks* [24]. Traditional neural networks encompass various forms, from which Xiaofeng and Yan [9] analyze a variety of architectures and claim that CNNs consistently outperform other architectures. The challenge of detecting handwritten digits has progressed to identifying multiple digits, where a mere CNN is no longer sufficient. The use of image-processing algorithms or the expansion of the neural networks to include boundary-box analysis is required. Recently, with the increase in computational power, the expansion of a neural network to include digit localization has seen a great amount of success. In [4], Muhammad *et al.* [4] use a unified multi-digit recognition approach that utilizes a single CNN to conduct localization and classification of up to 18 digits in an image. In [22], Ruzhang employs and visualizes the localization and classification of multiple handwritten digits with a CNN. There have been advancements with localization using image-processing algorithms, with some allowing for accurate text and digit localization that can then be used for classification [11, 15]. Nurul *et al.* [11] use Local Binary Pattern for extraction and K-Nearest Neighbor for classification [11]. In [15], a multiscale edge-based text extraction algorithm is proposed that is flexible to font size, style, color, orientation, and text alignment.

In our work, we make use of image convolutions and *Divide and Conquer* [20], with density-based clustering (a variation of *DBSCAN* [2]), to localize the handwritten digits, and a CNN is then used for identifying the digits. For the sake of space, we do not discuss their algorithmic details in this extended abstract.

3 Detecting Digits Through Localizing and Convoluting

3.1 Task A: Digit Detection Through CNN

The process of identifying handwritten digits from an individual is challenging for computers to achieve. Every individual has subtle changes in digits they write.

Our first task is to construct a convolutional neural network to predict what number a handwritten digit is. The result of this task will be a black box that can identify handwritten digits that have never been previously observed. For background knowledge, all the digits in this task are obtained from the MNIST dataset [13], with each consisting of 28×28 pixels with no noise. An example is shown in on the left in Fig. 2. It should be noted that, with the classification power of CNNs, classifying a handwritten digit is an easy task. An excellent implementation, among others, can be found in [5].

For Task A, a convolutional neural network is designed, implemented and trained on 60000 digits from the MNIST. After each epoch of training, the network is tested on 10000 sample MNIST digits [13]. The testing is conducted after each epoch to see the progression of accuracy for digits the model is never trained on. The architecture of the model contains two convolutional layers with a kernel size of 5, a stride of 1, a padding of 2, two ReLU layers, and two max-pooling layers with a kernel size of 2, as shown in Fig. 2. The model is trained using cross-entropy loss [17] and the Adam optimizer [17] for 15 epochs, with a batch size of 128 and a learning rate of 0.01.

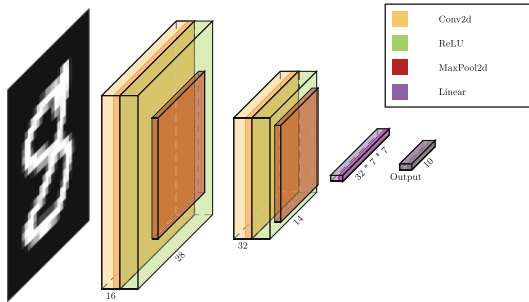


Fig. 2. Classification Model Architecture.

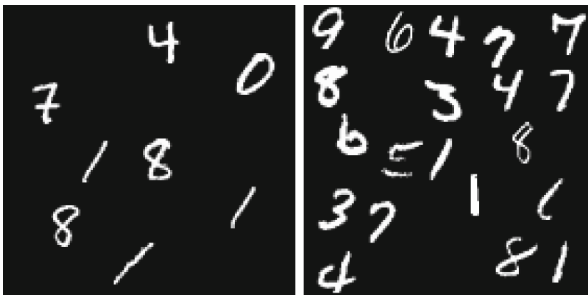


Fig. 3. Multiple digits with the same dimension sizes a large image.

3.2 Task B: Multiple-Digit Detection Through Identical Dimension Boundary Box Analysis

Employing our implementation of Task A as a black box, we then explore how to recognize multiple digits placed on a single large image. The large image will consist of $n \times n$ pixels, where n can be, for example, 512 or 1024, with each of the digits being of their original 28×28 pixels in size. The digits are placed randomly on the image *with no overlap*. Such an example is shown in Fig. 3.

The operation of detecting multiple digits on a single large image begins with the localization of each digit's 28×28 pixel boundary box. Once the boundary box of a digit is determined, the image inside the boundary box is input into Task A to classify it.

Task B: Version 1. The critical operation for this task is to obtain the boundary boxes of digits in a large image. With the use of an image convolution operation [7], the location of the digits can be identified. Image convolution is used to compute the dot-wise product of a 28×28 matrix (kernel, as mentioned in Sect. 2) with a selected 28×28 section of pixel values from the large image. If the calculation is done over a black (a value of 0) section, then the calculation will be 0, while rectangles that contain brighter pixels (a value of 1) will result in a larger computed value. We label the coordinate of the top left of the section with the value computed. The dot-product of every corresponding coordinate is mapped, resulting in some sections with larger values. The pixels that correspond to the sections with larger values represent the potential center location of a digit on the convolution image. The convolution operation is conducted using the `convolve2d` method in the Python SciPy library¹ with padding, as shown in Fig. 4. The coordinates of the larger value pixels are translated 21 pixels left and 21 pixels up, from the center location. At these coordinates, images of 28×28 pixels are extracted and then fed into Task A.

Task B: Version 2. For Task B, we also propose another variation. The original MNIST digits contain a border of multiple pixels that results in large spacing between digits. In order to better handle the challenges presented when classifying multiple digits, we preprocess the original MNIST digits and crop them to include only a one-pixel black border. We then randomly select multiple digits and put them on a large image. It can be seen that those digits get closer. This new challenge would render a convolution operation with a 28×28 pixel kernel no longer effective. In our experiment, a new kernel dimension size of 20×20 pixels is chosen as this is one pixel larger than the maximum MNIST digit size in pixels, as shown by the statistics in Table 1. This, however, does not provide easily detectable sections with larger values as seen in Task B Version 1. The convolution image produced with a 20×20 pixel kernel have peaks of increased values where digits are located and may also have had peaks in between digits, since some digits can get really close.

¹ <https://docs.scipy.org/doc/scipy/index.html>.

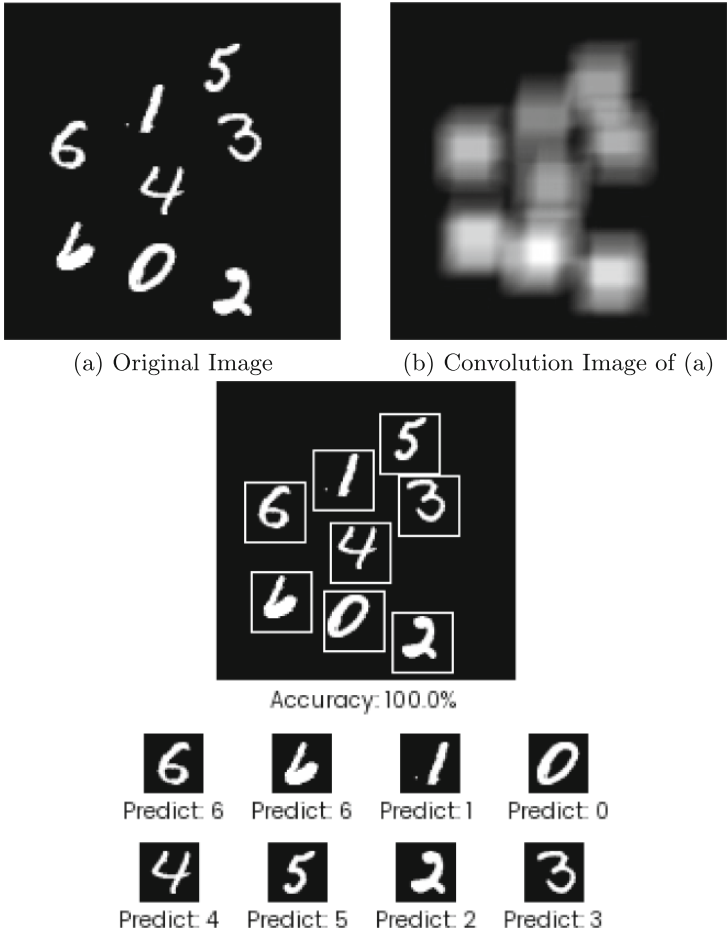


Fig. 4. 28×28 pixel digits localization and classification.

Table 1. Size Information for Digits in MNIST Dataset.

Measurement	Size (pixels)
Average Width	14.5
Average Height	18.8
Max Width	19
Max Height	19
Min Width	2
Min Height	9

Algorithm 1. Cluster Dimensions Extraction

```

Input:  $I$  (image, represented as 2D pixel data)
Output: width, height of each cluster present in image
1: procedure DIMENSIONS( $I$ )
2:    $(x, y) \leftarrow$  list of coordinates of all pixels greater than 0.1 in image
3:   x-index  $\leftarrow$  list of all x coordinates sorted
4:   y-index  $\leftarrow$  list of all y coordinates sorted
5:   Initialize an empty list  $X$ 
6:   Initialize an empty list  $Y$ 
7:   for each  $v_i, v_{i+1}$  in x-index and y-index do
8:     if  $v_{i+1} - v_i$  in x-index greater than 1 then
9:       save  $v_i + 1$  into  $X$ 
10:    end if
11:    if  $v_{i+1} - v_i$  in y-index greater than 1 then
12:      save  $v_i + 1$  into  $Y$ 
13:    end if
14:  end for
15:  if  $X$  and  $Y$  are empty then
16:     $width \leftarrow \max(x\text{-index}) - \min(x\text{-index}) + 1$ 
17:     $height \leftarrow \max(y\text{-index}) - \min(y\text{-index}) + 1$ 
18:    return  $width, height$ 
19:  end if
20:  for each  $(x_i, y_i)$  in  $X$  and  $Y$  do
21:     $S \cup$  slice  $I$  into smaller images at  $(x_i, y_i)$ 
22:  end for
23:  Remove all black images from  $S$ 
24:  for each  $s_i$  in  $S$  do
25:     $width, height \leftarrow$  Call DIMENSIONS( $s_i$ )
26:    Append  $width, height$  into  $D$ 
27:  end for
28:  return  $D$ 
29: end procedure

```

Our approach is presented in Algorithm 1, where the data structure D is a global variable. The algorithm begins by acquiring the coordinates of all pixels above 0.1, as this removes pixels that are extremely close to 0. It then horizontally and vertically checks where gaps are detected among the coordinates. If there are no gaps then there is one cluster and the dimensions of the cluster are returned. This can be seen in lines 15–19 in the algorithm. If a gap is detected between two pixels the lower pixel coordinate plus one ($v_i + 1$) is saved as this acquires the value where the gap starts and not the end of the first cluster, as seen in lines 7–14. The image is then sliced at every gap detected and the sliced images are recursively passed into the algorithm to extract all cluster dimensions. This can be found in lines 20–27 in Algorithm 1.

3.3 Task C: Multiple-Digit Detection Through Varying Dimension Boundary Box Analysis

The process of identifying and detecting the multiple digits becomes increasingly more difficult when the digit’s size (in pixels) varies. An example is shown in Fig. 5. A full image will be $n \times n$ pixels with each digit being between 12 pixels to $\frac{n}{3}$ pixels in dimension size ($n = 148$ in our experiments). The digits are placed randomly on the image with no overlap. Since the dimension size of each image is no longer known, the convolution-based method proposed for Task B is no longer effective.

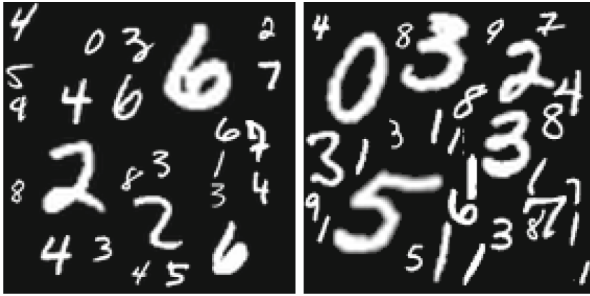
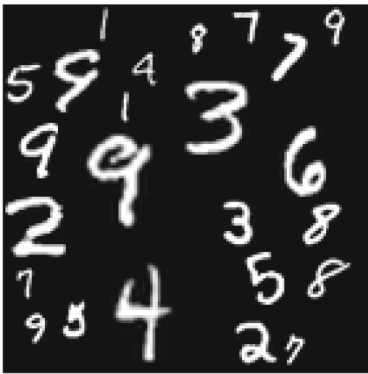
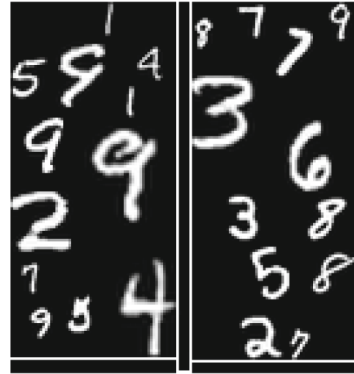


Fig. 5. Multiple digits with varying pixel sizes in a large image.



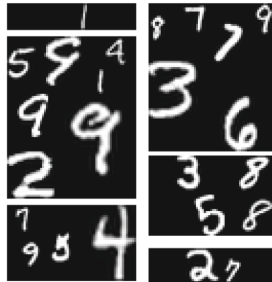
(a) Original Image



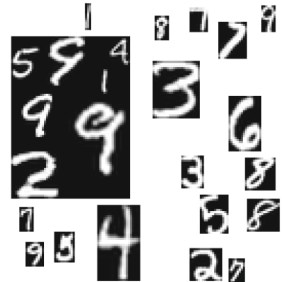
(b) Sliced Image of (a)



(c) Black Sliced Images Removed



(d) Recursively call (b) and (c)



(e) No Black Rows or Columns

Fig. 6. Divide-and-Conquer Digit Localization.

Task C: Version 1. We have devised a new algorithm inspired by *Divide and Conquer* [20] and *DBSCAN* [2]. The general notion is that we slice a large image recursively where black rows or columns are detected (as shown in Figs. 6 (a)–(d)). *DBSCAN* is used to detect multiple digits (as shown in Fig. 6 (e) and Fig. 7). Each of the detected clusters of digit pixels is then fed into Task A for

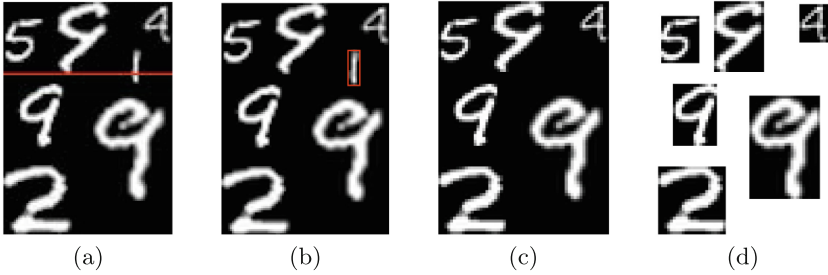


Fig. 7. Density-Based Clustering.

classification. For the sake of space, we omit the details of our algorithm in this extended abstract.

Task C: Version 2. In Version 2, we try to deal with a more complex situation, as shown in Fig. 8. The algorithm proposed for Task C: Version 1 would only have to be modified slightly to detect and classify multiple digits in an image. We omit the technical details. One such as example, after we apply our algorithm of Version 2 to it, is shown in Fig. 9 with the individual digits localized and classified.



Fig. 8. A more complex situation between two digits in Task C.

4 Experiment Results and Discussions

The data for all experiments to verify the effectiveness of the approaches are acquired from the *MNIST* (Modified Nist) database². It is a database that contains a total of 70000 handwritten digit images. All images are 28×28 pixels, with a total of 784 pixels each and with a brightness value from 0.0–1.0. All implementation is written in Python 3.9.5. Image processing algorithms are

² <https://pytorch.org/vision/stable/generated/torchvision.datasets.MNIST.html>.

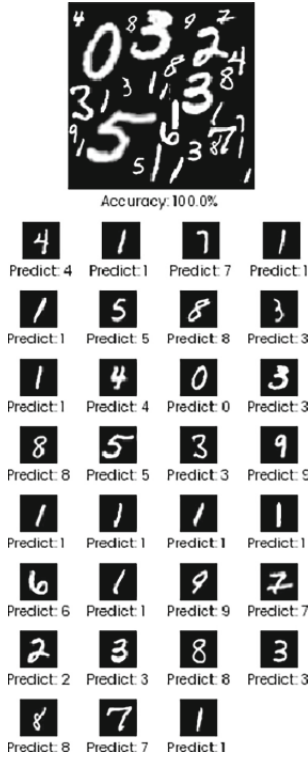


Fig. 9. Task C: Version 2. $n \times n$ pixel digit localization and classification.

created using the libraries Numpy³, Pillow⁴, SciPy⁵, and OpenCV⁶ with the neural network being constructed and trained using PyTorch⁷. Our experiments are conducted on a 2019 MacBook Pro containing a 2.6 GHz 6-Core Intel Core i7 CPU, an AMD Radeon Pro 5300M 4 GB GPU, and 16 GB 2667 MHz DDR4 RAM.

4.1 Results

For Task A, a trivial problem, our proposed CNN model is trained and validated on 60000 MNIST digits and tested on 10000 MNIST digits. After 15 epochs the training and testing accuracy is 99.29% and 98.59% respectively. It can be seen that the classification of a single digit for our CNN model is comparable to other classification methods and other neural network architectures [9, 14, 19, 24].

³ <https://numpy.org/>.

⁴ <https://pypi.org/project/pillow/>.

⁵ <https://docs.scipy.org/doc/scipy/index.html>.

⁶ <https://pypi.org/project/opencv-python/>.

⁷ <https://pytorch.org/>.

Table 2. Task B and Task C: Results Over 100 Images.

Task	Digits per Image	Average Localization Accuracy (%)	Average Accuracy (%)
B V.1	5–10	98.19	94.56
B V.2	10–20	97.43	93.66
C V.2	17–25	99.69	94.34
C V.2	23–43	98.27	94.29

For Tasks B and C, the final accuracy results of our proposed different algorithms proposed are shown in Table 2. In our evaluations, the accuracy of the algorithms is broken into two measurements, total accuracy and localization accuracy. The total accuracy is calculated by analyzing whether the predicted digits are present in the input image. Each digit present both in the predictions and the input image is removed from the input image and the accuracy is increased by 1. After all the predicted digits have been iterated through, the accuracy is then divided by the number of digits in the image. If extra digits are predicted, this method could report false positives [16]. The localization accuracy is computed by comparing the number of predicted digits and the number of digits present in the image. Any differences in the sizes would be considered as a decrease in accuracy. The average total accuracy and average localization accuracy are computed after all images are processed. This way, it allows for reliable validation even among the arbitrary digit locations, digit size dimensions, and order of the digits extracted. The separation between boundary box analysis accuracy and total accuracy allows for an easy comparison for algorithms that are proposed for detecting multiple digits in images in the literature.

4.2 Discussions

The challenge of character and digit recognition has been an incredibly popular topic for some time. The field has seen advancements in detecting multiple characters and digits in different complex real-world images with image processing algorithms [8, 15], as well with the use of deep learning neural networks [4, 18, 22]. There have been successful results with the use of edge-detection algorithms and neural networks [23]. However, some edge-detection algorithms struggle with expensive computation costs and with images containing a large number of dense clusters with low pixel separation [10].

The algorithms proposed and implemented in this work provide an efficient, robust, and adaptive solution, independent of digit sizes, the number of digits, and the size of the image. However, our proposed approaches are not applicable to the situation where heavy noises and the intersections of the digits are considered, as found in many complex real-world images. The intention of the study is to provide a possible approach to the challenge of classifying handwritten digits. We believe that our approach could be applied to various similar object detection tasks, such as letter detection, document processing, bank check processing, mail sorting, the entry of data, etc.

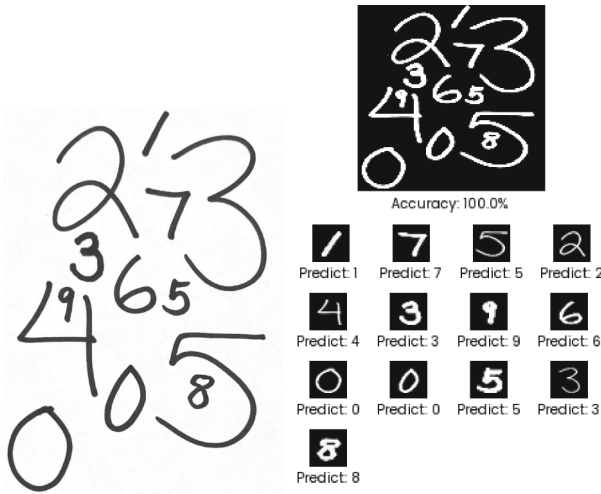


Fig. 10. An example of localization and classification of scanned digits.

The identification of multiple handwritten digits of the same dimension sizes is accomplished with the use of image convolutions with high accuracy. But this is limited by the requirement that the digit dimensions be known and used as the kernel dimension for the image convolution. This is highly specific and only maintains high accuracy in this niche problem. But our approach demonstrates the use of image convolutions in digit recognition establishes a viable option.

In order to classify the digits written by any individual, the size of the digits must remain arbitrary. An approach inspired by *Divide and Conquer* [20] and *DBSCAN* [2] is proposed and implemented that allows for the flexibility and adaptability aimed for. The algorithm mainly employs the faster divide-and-conquer algorithm over the slower density-based clustering algorithm to produce an efficient algorithm for identifying all handwritten digits in an image. However, when considering large images that have a higher digit density, the algorithm must resort to extracting many digits using the slower density-based algorithm.

It should be also noted that the accuracy of Task C is tested for 50 different background image sizes. The density of digits is relatively constant for each image size chosen, with per image containing 18–41 digits. One observation is that when scaling down images, the digits in the images are also scaled down. In order to determine the lower limit for scaling digits down, our algorithms are examined with all digit sizes at 10, 11, 12, and 13 pixels. We observe that, for digits whose size is below 12 pixels, the accuracy drastically decreases. So in our experiments, digits are scaled to a minimum of 12 pixels and a maximum of 50 pixels. The maximum value can be increased to any value in our approach. However, in our experiments 50 pixels is chosen as this allows for other digits to be added to the 140×140 pixel background image. The size of the background image may also be altered to any value. Also note that the size 140 pixels is arbitrarily selected, only for the demonstration of our experiments.

In addition to the existing digits images from MNIST, we also include scanned images in our experiments. The digits are written in a variety of styles by individuals using two colors and three different pen thicknesses. Each scanned image is scaled to 140×140 pixels, converted to gray scale. The colors are inverted and the contrast is increased. Experiments with 20 images containing 7–17 digits per image have 93.33% with an average localization accuracy of 97.84%. An example is shown in Fig. 10 for examination.

We have conducted empirical analysis of execution times of our proposed algorithms in our tasks. Due to the space limit, we omit it in this abstract.

5 Conclusion

In this study, a comprehensive approach to the classification of handwritten digits is designed, implemented, and discussed through the development and evaluation of a hybrid of CNN and some classic algorithms. We have showcased the effectiveness of our proposed approach through a series of experiments. We highlight the specialization of the proposed algorithms, with each tailored to tackling distinct challenges in handwritten digit recognition. We believe that our approach provides a viable option for various applications of object detection, including letter detection, document processing, bank check processing, mail sorting, data entry, and etc. As for our future work, our approach faces challenges in scenarios involving heavy noise and intersections of digits, a characteristic of complex real-world images. At this moment, we are considering whether we can introduce noises into the digit images in MNIST and use them to train our CNN model. We also plan to handle the situation where a digit's size in an image is too small as well as a digit may rotate by some degree.

References

1. Aggarwal, C.C.: Neural Networks and Deep Learning, vol. 10, p. 493. Springer, Cham (2018). <https://doi.org/10.1007/978-3-031-29642-0>
2. Ali, T., Asghar, S., Sajid, N.A.: Critical analysis of DBSCAN variations. In: Proceedings of the International Conference on Information and Emerging Technologies, pp. 1–6 (2010)
3. Anyoha, R.: The history of artificial intelligence (2020). <https://sitn.hms.harvard.edu/flash/2017/history-artificial-intelligence>
4. Asif, M., Bin Ahmad, M., Mushtaq, S., Masood, K., Mahmood, T., Ali Nagra, A.: Long multi-digit number recognition from images empowered by deep convolutional neural networks. *Comput. J.* **65**(10), 2815–2827 (2022)
5. Chollet, F.: Deep learning with python. Manning (2021)
6. Chychkarov, Y., Serhienko, A., Syrmamiikh, I., Kargin, A.: Handwritten digits recognition using SVM, KNN, RF and deep learning (2021). <https://ceur-ws.org/Vol-2864/paper44.pdf>
7. Dominguez, A.: A history of the convolution operation (2015). <https://www.embs.org/pulse/articles/history-convolution-operation/>

8. Goodfellow, I.J., Bulatov, Y., Ibarz, J., Arnoud, S., Shet, V.: Multi-digit number recognition from street view imagery using deep convolutional neural networks. arXiv preprint [arXiv:1312.6082](https://arxiv.org/abs/1312.6082) (2013)
9. Han, X., Li, Y.: The application of convolution neural networks in handwritten numeral recognition. *Int. J. Database Theory Appl.* **8**(3), 367–376 (2015)
10. Igbinoso, I.E.: Comparison of edge detection technique in image processing techniques. *Int. J. Inf. Technol. Electr. Eng.* **2**(1), 25–29 (2013)
11. Ilmi, N., Budi, W.T.A., Nur, R.K.: Handwriting digit recognition using local binary pattern variance and k-nearest neighbor classification. In: *Proceedings of the 4th International Conference on Information and Communication Technology (ICoICT)*, pp. 1–5 (2016)
12. Kumar, B.: Convolutional neural networks, a brief history of their evolution. <https://medium.com/appyhigh-technology-blog/convolutional-neural-networks-a-brief-history-of-their-evolution-ee3405568597>
13. LeCun, Y., Cortes, C., Burges, C.: MNIST handwritten digit database. ATT Labs (2010). <http://yann.lecun.com/exdb/mnist>
14. Liu, C.L., Nakashima, K., Sako, H., Fujisawa, H.: Handwritten digit recognition: benchmarking of state-of-the-art techniques. *Pattern Recogn.* **36**(10), 2271–2285 (2003)
15. Liu, X., Samarabandu, J.: Multiscale edge-based text extraction from complex images. In: *Proceedings of the IEEE International Conference on Multimedia and Expo*, pp. 1721–1724 (2006)
16. Manoa, H.: False positives and false negatives. <https://manoa.hawaii.edu/exploringourfluidearth/chemical/matter/properties-matter/practices-science-false-positives-and-false-negatives>
17. Rajpal, G.: Optimizer and loss functions in neural network (2020). <https://medium.com/analytics-vidhya/optimizer-loss-functions-in-neural-network-2520c244cc22>
18. Sermanet, P., Chintala, S., LeCun, Y.: Convolutional neural networks applied to house numbers digit classification. In: *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, pp. 3288–3291 (2012)
19. Simard, P.Y., Steinkraus, D., Platt, J.C., et al.: Best practices for convolutional neural networks applied to visual document analysis. In: *ICDAR*, p. 6 (2003)
20. Smith, Douglas R.: *Science of Computer Programming* 5, pp. 37–51, 54–58. Elsevier Science Publishers B.V. (1985)
21. Staff, C.: Machine learning vs. AI: differences, uses, and benefits (2023). <https://www.coursera.org/articles/machine-learning-vs-ai>
22. Yang, R.: Classifying hand written digits with deep learning. *Intell. Inf. Manag.* **69** (2018)
23. Yang, X., Pu, J.: Multi-digit recognition using convolutional neural network on mobile. In: *Proceedings to Yang2015 MDigMR*, pp. 1–10 (2015)
24. Yu, H., Xie, T., Hamilton, M., Wilamowski, B.: Comparison of different neural network architectures for digit image recognition. In: *Proceedings of the 4th International Conference on Human System Interactions, HSI 2011*, pp. 98–103 (2011)
25. Zoumana, K.: *Classification in machine learning: a guide for beginners* (2022). <https://www.datacamp.com/blog/classification-machine-learning>



Djinn—Data Journalism Interface for Newsgathering and Notifications

Sara Elo Dean¹, Lars Adrian Giske²(✉), Herman Jangsett Mostein³,
Silvia Podestà⁴, Halvor Helland Barndon⁴, Sara Stegane³,
and Henrik Nordberg³

¹ IBM, Helsinki, Finland
sara.elo.dean1@fi.ibm.com

² iTromsø, Tromsø, Norway
giske@itromso.no

³ Visito, Bergen, Norway
{herman,sara,henrik}@visito.no

⁴ IBM, Copenhagen, Denmark
{silvia.podesta,halvor}@ibm.com

Abstract. Journalists often face the daunting task of manually sifting through vast amounts of documents to uncover newsworthy story ideas. The Djinn platform, or “Data Journalism Interface for Newsgathering and Notifications”, developed by iTromsø, Visito, and IBM, addresses this challenge by leveraging generative AI, supervised machine learning, and rule-based algorithms. Djinn processes municipal documents from Norwegian archives, ranks them by newsworthiness, and generates efficient summaries and visual thumbnails. Its architecture includes a central document ranker alongside local rankers tailored to individual newsroom preferences, enhancing the identification of relevant content. Djinn fosters user trust through explainable AI components and feedback mechanisms. Business results from March–April 2023 and 2024 indicate significant increases in story production and reader engagement for newsrooms using Djinn. This paper discusses Djinn’s design and architecture, the challenges encountered during development—including data availability and privacy laws—and its contributions to computational journalism by illustrating the practical application of AI technologies in investigative journalism.

Keywords: Datadriven journalism · AI · Machine Learning

1 Introduction

Investigative journalism within the urban planning, housing, and development domain is crucial for uncovering stories of societal significance. However, manually reviewing large volumes of municipal documents is labor-intensive, hindering journalists’ ability to focus on fact-checking and interviews.

S. Elo Dean, L. A. Giske, H. J. Mostein and S. Podestà—Equal Contribution.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
M. Bramer and F. Stahl (Eds.): SGAI 2024, LNAI 15447, pp. 147–161, 2025.
https://doi.org/10.1007/978-3-031-77918-3_11

The Djinn project aimed to explore how generative artificial intelligence (AI) and classical machine learning (ML) could meet this challenge, and enhance newsroom value capture and build trust in AI within media.

Developed by iTromsø in collaboration with Visito and IBM, the Djinn platform automates the extraction and ranking of relevant information from documents, helping journalists uncover impactful stories that promote public understanding and participation in local democracy. The platform was designed to empower journalists by replacing inefficient municipal web tools with a more effective solution, improving journalistic productivity and the quality of information available to readers.

This paper presents the design and architecture of the Djinn platform, detailing its combination of AI approaches to support various search paradigms and enrich unstructured text. The platform includes functionalities such as document ranking, data labeling, summarization, and named entity recognition. Additionally, it discusses challenges faced during development and scaling, including data availability and risks associated with generative AI.

2 Related Work

The challenge of classifying documents as newsworthy has been explored by Spangher et al. [26]. In their case, data is labelled based on whether it resulted in a front-page story and utilize a BERT model for classification, similar to Djinn. Liu et al. [16] adopt an unsupervised approach to identify newsworthy stories using Twitter streams, scoring them by tweet significance. Their method enhances context awareness through clustering to filter out noise and chit-chat, while also summarizing and extracting named entities from documents, akin to Djinn.

3 Design of the Djinn Solution

The Djinn platform originated as an experimental project leveraging generative AI and has evolved into a comprehensive end-to-end journalist’s tool. This platform utilizes the capabilities of generative large language models (LLMs), which are paradigm-shifting innovations significantly impacting the media and publishing industries. News outlets, regardless of size, are increasingly exploring AI-powered technologies to revitalize their business models and create new efficiencies. Common AI applications in journalism include monitoring breaking news, alerting journalists to new information, extracting and converting data, generating text, performing automatic spell checks, and detecting fake news [1].

3.1 Catering to Different Search Paradigms

Djinn supports three primary use cases, each addressing specific informational and navigational user intents, as defined by Broder [4].

- **Insight Extraction:** Allows users to uncover key entities and topics within documents, facilitating serendipitous discovery without needing a specific query.
- **Recommendation Systems:** The platform predicts the “newsworthiness” of content, a critical measure in journalism that indicates the potential for information to lead to hit stories. This helps the users prioritize their investigative efforts. The prediction is made by classifiers trained on labeled examples provided by journalists at iTromsø and other Polaris Media newsrooms.
- **Information Retrieval:** Provides targeted search results for users with specific information needs, enhancing the efficiency of their search experience.

Compared to traditional keyword-oriented search tools, the platform stands out by simplifying information evaluation and shifting agency from the user to the AI system. In a keyword search, users must explicitly decide what to search for and manually assess the value of results. In contrast, an AI-driven experience delegates this value determination to a system that can interpret content and predict its relevance for journalists.

3.2 Power of Complementary Techniques

For Djinn to function effectively, it must manage unstructured text, diverse terminology, and various file formats. To address this variability, Djinn employs multiple parallel approaches, including generative models, classifiers, entity extraction, and rule-based logic. Relying on a single method may not adequately cover the range of anomalies; thus, complementary techniques are more likely to provide useful insights for journalists determining document relevance for their investigation. For example, a brief email might not yield a helpful summary, but a certain person’s name within it could prompt further reading. Conversely, a 30-page building plan may rank high due to mentions of protected buildings in the text, despite its generic title. While a blueprint of a construction site may lack insights, a visual thumbnail can help users quickly decide whether to open the document. Where one technique falls short of providing valuable information, another may reveal critical points of interest.

4 Solution Architecture

This all comes together in the Djinn platform, which is composed of several integrated technology components. The municipal data is processed and presented to the user as shown in Fig 1.

1. The content originates in **data sources** provided by the municipalities.
2. The **data pipeline** downloads the content, normalizes the format and meta-data, and stores each document for downstream processing.
3. The **document store** contains the text of each document and the **meta-data store** manages the meta-data associated with each document.

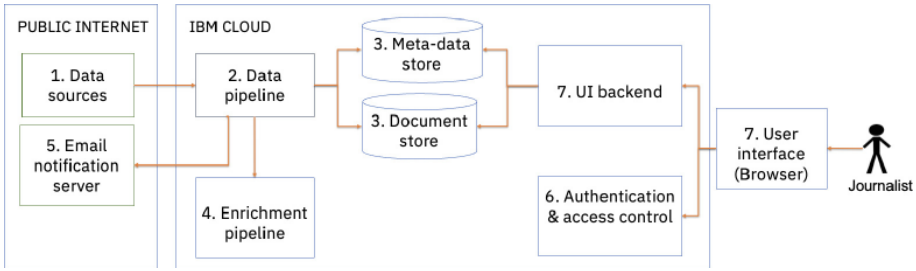


Fig. 1. Data processing and user interface components.

4. The **enrichment pipeline** generates a visual thumbnail and a summary for each document, extracts named entities, and predicts the relevance of each document, adding new meta-data to the **meta-data store**.
5. The **email notification server** informs journalists of daily highlights via email.
6. The **authorization and access control** service identifies users and provides journalists access to Djinn.
7. The **user interface** presents a dashboard where journalists can explore the ranked documents, filter them by time period and municipality, and search for relevant information across all documents or a subset thereof.

Training of the relevance prediction model is conducted using the Djinn components depicted in Fig. 2.

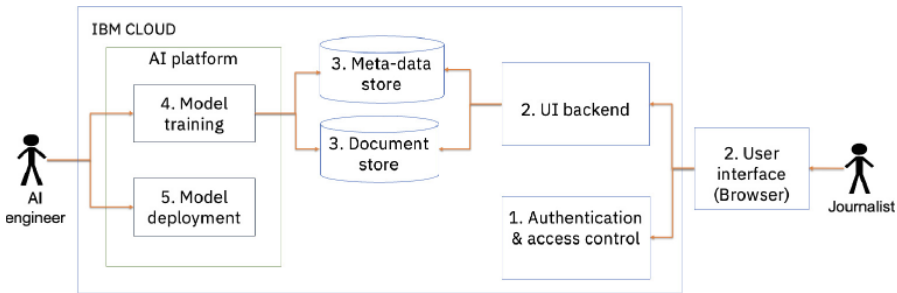


Fig. 2. Djinn data labeling and model training components.

1. The **authorization and access control** service identifies users, granting them access to the Djinn dashboard for the appropriate municipalities.
2. The **user interface** allows the journalist to label documents as examples of “newsworthy” and “not newsworthy” for ranker model training.
3. The labeled examples are stored in the **document store** and **meta-data store**.

4. The **model training** environment in the AI platform allows the AI engineer to conduct ranker training using the labeled data.
5. The AI engineer deploys the trained ranker model to the **model deployment** environment, making it available for predictions in the **enrichment pipeline** during data processing.

The following subsections describe the technical details of the Djinn components.

4.1 Technology Stack

Djinn is hosted in IBM Cloud [9], IBM’s Enterprise Public Cloud platform and runs on top of watsonx.ai [14], IBM AI platform supporting generative AI and machine learning development. Djinn data pipeline and user interface software images are deployed in IBM Code Engine [10], a fully managed serverless platform that runs containerized workloads. Djinn uses as the document store IBM Cloud Databases for PostgreSQL [13], an object-relational database system, and as the meta-data store IBM Cloud Object Storage [11], a scalable and resilient managed data service. IBM App ID [12] is used to manage authentication via Single sign-on as well as access control. Email notifications are sent using an SMTP server.

4.2 Modelling Newsworthiness

Djinn is designed to accommodate various newsrooms with distinct preferences and data sources. This structure necessitates the distribution of document ranking capabilities, allowing different newsrooms to receive tailored recommendations on similar documents. Because journalists label documents for ranker training while identifying potential news stories, generating training data is a gradual process. Therefore, a single model for each newsroom would not efficiently leverage all available information for training. Despite differing preferences, common attributes exist in documents of interest across newsrooms.

Central Model Architecture. To balance the exploitation of information and preserve newsroom-specific preferences, a **Central Model Architecture (CMA)** is proposed. The CMA consists of two types of document ranker models, the central model and the local model. The central model is a single large binary classification model that trains on labeled feedback from all newsrooms. The intent behind this document ranker is to recognize universally interesting documents. The base model of the central ranker, which has been fine-tuned for the classification task, is NorBERT, a Bidirectional Encoder Representations from Transformers (BERT) model trained from scratch on Norwegian language data. This makes it particularly suitable for analyzing the often complex, jargon-filled and bureaucratic Norwegian language used in the documents being processed [24].

In addition to the central model, each newsroom is equipped with a local document ranker model. This model, a binary classifier trained solely on labeled data from its corresponding newsroom, learns patterns specific to that newsroom. The local models are trained with a Support Vector Machine (SVM) algorithm with fewer parameters than the central model, optimizing for computational efficiency and cost.

Inference. During inference, a document is classified by both the central model and the local model associated with the newsroom. Both models return a confidence score that indicates how likely the document belongs to the “newsworthy” or “not newsworthy” class. The two confidence scores are aggregated by an aggregation function.

The central confidence R_c and the local confidence R_l are aggregated through the following function.

$$R = R_l R_c + R_l^2(1 - R_c) + R_c^2(1 - R_l) \quad (1)$$

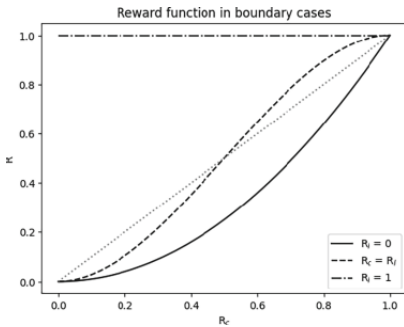


Fig. 3. The dashed line represents the final rank when R_c and R_l are the same. The solid line represents the final rank when one of the ranks is 0. The mixed line represents the final rank if one of the ranks is 1. The dotted line is a reference.

gives a zero output, or close to zero, the aggregation function squares the other model’s confidence, to decrease the final rank, invalidating the confidence of the non-zero ranker. If both the models predict the same, that is, $R_c = R_l$, their combined signal is enhanced through the aggregate function. Note that the function is completely symmetrical, which entails that the local and central models have the same influence on the final output.

4.3 Data Labeling

The rankers in the Djinn platform are trained on datasets labeled by journalists. After extensive user testing, a simple thumbs-up or down feedback feature

This function generates the final rank shown to the user by combining the central and local ranks. The function ensures that if one of the central or local models ranks a document as newsworthy with high confidence, the final rank will classify the document as newsworthy, even if the other one disagrees. This is to strongly prevent false negatives in the classification, as a false negative is worse than a false positive, as it could lead to important documents being missed by the journalists.

In addition to preventing false negatives, this function also enhances the signals given from the local and central models under certain conditions (Fig 3). If one of the models

was integrated into the web interface for each document. During onboarding, journalists are encouraged to provide balanced feedback on the newsworthiness of documents, focusing particularly on outliers that may be incorrectly ranked. This corrective feedback helps improve the models' generalization abilities. Following the rollout of Djinn to 30 new newsrooms in February 2024, journalists labeled 2,000 documents within a week, demonstrating the effectiveness of the design in encouraging feedback.

Training and retraining local models on this feedback allows them to reflect individual newsroom preferences. While newsworthy construction documents often share common language patterns, the trained models can generalize and rank previously unseen documents. However, ensuring data quality and volume poses challenges, as journalists may use different criteria for evaluating newsworthiness, leading to potential inconsistencies in labeling. To address this, users receive labeling guidelines during onboarding, and the purpose of feedback is clearly explained to promote consistency [3].

4.4 Summarization

Even with a ranked list of documents, a journalist still has to open documents to assess the content, which is a time-consuming task when each municipality publishes anywhere between 50 and 500 potentially newsworthy documents every day. To further enhance the user's ability to make quick decisions, every document in Djinn is summarized by a generative large language model (LLM). This gives the user insight into the contents fast, allowing them to evaluate the relevance of the document, and if needed, to give corrective feedback on the newsworthiness of the document.

The summary is generated by the Llama-2-70b-chat model, hosted in the IBM watsonx.ai platform. While the model is only pre-trained on 0.03% Norwegian language data, the journalists still find its language capabilities to be satisfactory for their needs [27].

4.5 Named Entity Recognition

Named Entity Recognition (NER) enhances document assessment by extracting key entities such as person names, company names, locations, and job titles. This information is presented to users alongside rankings and summaries, helping journalists gauge relevance. For example, a mention of a prime minister in a zoning dispute may indicate newsworthiness. In Djinn, named entities are extracted using the pre-trained encoder-only LLM IBM Slate [15]. The model was trained by IBM on multiple languages simultaneously, including Norwegian Bokmål and Nynorsk, and is able to decode these languages with better accuracy than models trained only on one language [20]. In Djinn, Extracted entities are re-ranked based on their TF-IDF weight, which measures the significance of a term within a document relative to a larger corpus. Higher TF-IDF weights prioritize entities for display, ensuring users see the most relevant information for efficient news gathering.

5 Designing for Trust

Studies have long identified how a lack of user trust is a key factor in enterprise AI implementation failures [2]. Therefore designing for trust is an essential aspect in the development of AI-driven search experiences, such as Djinn. Effective search design has traditionally relied on cognitive processes like information foraging, utilizing UI techniques such as descriptive titles and clear labeling [23]. However, the introduction of AI brings fresh trust-related challenges. Societal trust can hinder technology adoption [17] and therefore the pursuit of responsible, trustworthy, and explainable AI (XAI) is now a critical industry concern [5, 19]. Yet the interplay between user interactions, system explainability, and trust remains underexplored [7], with the literature mainly focusing on algorithmic transparency, accountability, explainability, and privacy considerations [19]. With Djinn, we examined the relationship between AI user experience (AIUX) and trust-building along two dimensions: 1) user flow and 2) interface layout.

- 1) The user feedback mechanism in Djinn, featuring thumbs-up and down options, allows users to engage with the system’s decision rules, which research shows positively correlates with user trust [7].
- 2) Initially, a table layout was used to help journalists transition from traditional search experiences, providing a comprehensive overview of results for exploration at their own pace. In contrast, the card-based layout in the production platform presents AI-curated results, shifting more decision-making power to the system and requiring greater user trust for adoption (Fig. 4). A transitional UI approach or explainability mechanisms, such as source referencing seen in some conversational search engines, could help address user acceptance challenges.

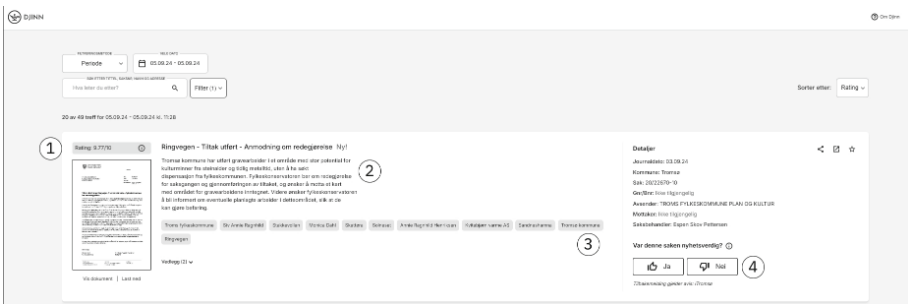


Fig. 4. The main components of Djinn interface: 1) Ranking score 2) AI-generated summary 3) Extracted entities 4) Feedback option.

5.1 Interface Components for AI-Based Search

The Djinn interface enhances search experiences by incorporating design elements that elevate perceived information value for users [25] and guide users to

the most pertinent results [30]. Unlike traditional keyword-based search engines, Djinn leverages generative AI outputs, which are inherently probabilistic and variable, necessitating flexible layout designs and specific components to ensure coherence and intuitiveness [15].

Content Cards. Djinn moves beyond conventional list displays, such as Google’s Search Engine Results Pages (SERP), by utilizing card components. These cards effectively address the design challenges posed by the variability of AI-generated outcomes. Their flexible size accommodates multimedia content and encourages user interaction, resembling playing cards that prompt actions like clicking or tapping [18].

Document Summaries. Variability in summary length is managed through a “Load more” feature. This promotes visual cohesion and reduces cognitive load, by giving users the option to visualize more content on request also helps reduce users’ cognitive load.

Extracted Entities Tags. Tagging with major keywords improves document readability and navigation among related content.

Content Relevance Score. This score is included in the card design, enhancing explainability and user trust in AI retrieval.

User Feedback. This feature allows users to contribute feedback, which helps retrain the ranker models and fosters a sense of agency, control and trust in the AI outputs.

5.2 Defining User Interactions with AI

The design of Djinn adheres to IBM Design’s UX guidelines, emphasizing AI’s role in journalists’ workflows and its interaction with human users. This context shapes the user’s mental model, with Djinn embodying the ‘Coach’ role identified by Papachristos [22], which helps users achieve goals and sets clear expectations using relevance scores. By highlighting the probabilistic nature of outputs, Djinn encourages critical thinking and fact-checking among journalists, aligning with media literacy principles that advocate for the critical evaluation of information sources [28].

Recognizing the risk of user over-reliance on AI, Djinn’s design incorporates mechanisms to mitigate this issue, ensuring that journalists remain engaged in their decision-making processes. However, the interplay between trust mechanisms and decision-making in AI-driven search experiences needs further research. Best practices suggest using disclaimers for AI-generated content; IBM employs an AI slug and specific modes in its Carbon for AI extension to indicate when AI is involved in processes [8]. More focused research could clarify how trust influences journalists’ reliance on AI-generated information.

6 Challenges

Development of the Djinn platform began in late April 2023. The pilot phase concluded successfully by mid-June 2023, where the platform was tailored to the needs of the iTromsø newsroom, focusing solely on documents from the Tromsø municipality. By March 2024, the platform was productionized and expanded to 35 newsrooms across Norway. Scaling the platform introduced new challenges. While data availability in Tromsø was excellent, the same was not the case for all municipalities.

6.1 Data Availability

Norway’s government encourages public participation in planning processes [21], making zoning and development an ideal testing ground for AI-driven solutions. This shared interest between the press and authorities eased data access during the project.

Norway’s transparency laws guarantee access to public management and policymaking information, balanced by privacy and business regulations from Norway and the EU, allowing exemptions for sensitive personal or business information [6]. The press enjoys GDPR exemptions, allowing Djinn to process the relevant data and store it for editorial use.

Currently, 55% of the 110 municipalities covered by Polaris Media openly publish planning and development documents online, while the remaining municipalities either face challenges in doing so due technical limitations or lack of resources or do not consider it necessary.¹

Three of the municipalities that did not initially provide downloadable data—Senja, Harstad, and Alta—made the necessary data available, or are in the process of doing so, as a result of the Djinn project. Additionally, Midt-Telemark lifted their ban on foreign IP addresses, enabling Djinn and other cloud-based platforms to access their data.

6.2 Transparency

While over half of the municipalities covered by Polaris Media publish full-text documents on their web portals, the level of transparency varies. Some municipalities publish all planning, zoning, and development documents, while others exempt certain categories or specific documents. Access to this information often requires manual FOIA requests.

Djinn does not automate the FOIA request process due to the risk of generating an overwhelming number of requests. Norwegian law requires FOIA requests to be of a “reasonable” scope [6] (§24). Automated requests for all non-public documents would likely be denied due to resource constraints.

¹ We charted and evaluated data availability before scaling up the Djinn solution from iTromsø, which covers the Tromsø municipality in Northern Norway, to 35 Polaris Media newsrooms across Norway. This data can be made available upon request.

Targeted automated requests for potentially newsworthy documents could address this issue but remain outside the current scope of Djinn.

Municipal websites vary in their data hosting solutions, ranging from structured APIs to web pages with embedded JavaScript dynamically generating document links. Due to this variability, Djinn uses configuration templates to specify how data from each municipality is processed. Web scraping poses challenges as changes to HTML templates can cause failures.

6.3 Risks of Generative AI

Hallucination in generative AI refers to the generation of nonsensical, inaccurate, or detached text. Hallucinations pose a potential risk for organizations adopting LLMs. In the case of Djinn, this risk is mitigated by design. The summary generated by Djinn is part of a “human-in-the-loop” editorial process; the AI-generated summary is not presented to readers as is. The journalists use the summary to decide where to focus their time. The investigation and story-writing is still conducted by the journalist.

The Llama 2 70B model used by Djinn has an estimated accuracy of 94.9% and a hallucination rate of 5.1% [29]. With Djinn processing around 12000 documents monthly, this translates to approximately 612 summaries containing errors.

To mitigate this, journalists and editors have been made aware of the risk of relying solely on summaries. Document ranking and entity extraction complement the summarization feature and allow the journalist to consider more than just the summary to determine document relevancy.

7 Business Results

Business data related to the news category “housing and property” shows a significant increase in both story production and reader engagement in March and April 2024, after the rollout of Djinn, compared to the same months in 2023, before its implementation. This trend is evident across several newsrooms, indicating improved performance in published stories and reader interaction.

7.1 Increased Traffic and Production

The positive effects of Djinn on reader engagement are evident in the total traffic share changes from 2023 to 2024. Traffic share describes how much of the total page traffic of a newsroom is tied to a specific news category, in this case “housing and property”. A majority of newsrooms saw substantial increases in traffic share, with one newsroom experiencing a 1316 percent increase, indicating improved reader attraction and retention (Fig. 5). Several newsrooms also experienced significant increases in their production (Fig. 5). This indicates a meaningful boost in the number of articles produced, reflecting a positive shift in editorial productivity within the category. While some newsrooms experienced

declines, such as one with a 41% decrease in production, these cases provide valuable insights for further optimization. Overall, the data highlights the potential of Djinn to enhance newsroom productivity and underscores the importance of tailored strategies for different editorial preferences.

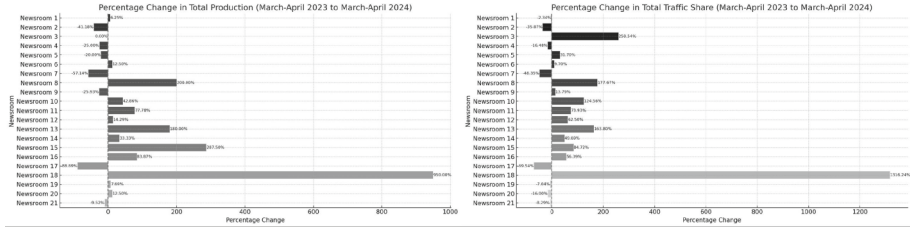


Fig. 5. The figure shows the change in traffic share (right) and the percentage change in production (left) within the property and housing category for newsrooms using Djinn from March-April 2023 to March-April 2024.

7.2 Data Limitations

It is important to note that this analysis is based on the available data, which may be influenced by several factors:

- **Data completeness:** The data includes only those newsrooms for which comparable data from 2023 and 2024 exist. Many newsrooms using Djinn do not have recent data available.
- **Tagging practices:** Variations in how content is tagged/categorized in the content management system (CMS) can influence the data. Different routines for tagging relevant content might lead to discrepancies. Improved consistency and accuracy in tagging practices, or lack thereof, could account for some of the observed changes.
- **Content focus:** An increased focus on the development beat or specific topics may contribute to the positive effects observed.
- **Temporal limitations:** The analysis covers a specific period (March-April) and may not capture longer-term trends or seasonal variations.

These caveats suggest that while the initial results are promising, a comprehensive evaluation over a more extended period and with a more complete dataset would provide a more accurate assessment of Djinn’s impact.

8 Conclusion

The Djinn platform has demonstrated significant potential for enhancing newsroom productivity and reader engagement. By leveraging a combination of generative AI, machine learning, and rule-based algorithms, Djinn effectively pre-reads documents and surfaces relevant information, enabling journalists to focus

on higher-level tasks such as fact-checking, conducting interviews and editing. The integration of multiple complementary techniques allows Djinn to handle the variability in municipal documents, providing journalists with accurate and timely insights across a range of newsrooms, both large and small.

The traffic and production data from March-April 2023 and 2024 highlights the positive impact of Djinn, suggesting that AI-powered tools like Djinn can drive both productivity and audience engagement.

When it comes to municipal data availability, the need for normalization and common practices is paramount. Variations in data hosting solutions, interpretation of transparency laws, and publishing practices across municipalities pose challenges that must be addressed. Standardized procedures for document publication and exemption can enhance the consistency and reliability of the data, facilitating better integration with AI-driven platforms like Djinn. The challenges of scaling the platform, such as data availability and lack of transparency, highlight the need for ongoing collaboration with municipalities, data providers, and relevant government institutions.

Overall, the Djinn platform—and the combination of generative AI with other machine learning approaches—represents a promising advancement in data journalism, offering valuable tools to support journalists in uncovering newsworthy stories and fostering greater public participation in local democratic processes. Continued refinement and expansion of Djinn will further enhance its capabilities and impact, contributing to a more informed and engaged society.

References

1. Al-Adwan, A., Li, N., Al-Adwan, A., Abbasi, G., Albelbisi, N., Habibi, A.: Extending the technology acceptance model (TAM) to predict university students' intentions to use metaverse-based learning platforms. *Educ. Inf. Technol.* **28** (2023). <https://doi.org/10.1007/s10639-023-11816-3>
2. Bach, T.A., Khan, A., Hallock, H., et al.: A systematic literature review of user trust in AI-enabled systems: an HCI perspective. *Int. J. Hum.-Comput. Interact.* **40**(5), 1251–1266 (2024). <https://doi.org/10.1080/10447318.2022.2138826>
3. Barbosa, N., Chen, M.: Rehumanized crowdsourcing: a labeling framework addressing bias and ethics in machine learning, pp. 1–12 (2019). <https://doi.org/10.1145/3290605.3300773>
4. Broder, A.: A taxonomy of web search. In: *ACM SIGIR Forum*, vol. 36, pp. 3–10. ACM, New York (2002)
5. Minh, D., Wang, H.X., Li, Y.F., Nguyen, T.N.: Explainable artificial intelligence: a comprehensive review. *Artif. Intell. Rev.* 1–66 (2021). <https://doi.org/10.1007/s10462-021-10088-y>
6. Freedom of Information Act: Act relating to the right of access to documents held by public authorities and public undertakings (lov-2006-05-19-16) (2009). <https://lovdata.no/dokument/NLE/lov/2006-05-19-16/>

7. Guo, L., Daly, E.M., Alkan, O., Mattetti, M., Cornec, O., Knijnenburg, B.: Building trust in interactive machine learning via user contributed interpretable rules. In: Proceedings of the 27th International Conference on Intelligent User Interfaces, IUI 2022, pp. 537–548. Association for Computing Machinery, New York (2022). <https://doi.org/10.1145/3490099.3511111>
8. IBM: IBM Carbon Design System (2024). <https://carbondesignsystem.com/all-about-carbon/what-is-carbon/>
9. IBM: IBM Cloud (2024). <https://www.ibm.com/cloud>
10. IBM: IBM Cloud Code Engine (2024). <https://www.ibm.com/products/code-engine>
11. IBM: IBM Cloud Object Storage (2024). <https://www.ibm.com/products/cloud-object-storage>
12. IBM: IBM App ID (2024). <https://www.ibm.com/products/app-id>
13. IBM: IBM Cloud Databases for PostgreSQL (2024). <https://www.ibm.com/products/databases-for-PostgreSQL>
14. IBM: watsonx.ai (2024). <https://www.ibm.com/products/watsonx-ai>
15. Lang, A.: Fair is fast, and fast is fair: IBM slate foundation models for NLP (2023). <https://medium.com/@alex.lang/fair-is-fast-and-fast-is-fair-ibm-slate-foundation-models-for-nlp-3508412a4b04>
16. Liu, X., Li, Q., Nourbakhsh, A., et al.: Reuters tracer: a large scale system of detecting & verifying real-time news events from twitter. In: Proceedings of the 25th ACM International Conference on Information and Knowledge Management (CIKM 2016) (2016). <https://doi.org/10.1145/2983323.2983363>
17. Lockey, S., Gillespie, N., Holm, D., Asadi Someh, I.: A review of trust in artificial intelligence: challenges, vulnerabilities and future directions (2021). <https://doi.org/10.24251/HICSS.2021.664>
18. Laubheimer, P.: UI cards component (2024). <https://www.nngroup.com/articles/cards-component/>
19. Lukyanenko, R., Maass, W., Storey, V.: Trust in artificial intelligence: from a foundational trust framework to emerging research opportunities. *Electron. Mark.* **32**, 3 (2022). <https://doi.org/10.1007/s12525-022-00605-4>
20. Moon, T., Awasthy, P., Ni, J., Florian, R.: Towards lingua franca named entity recognition with BERT (2019). <https://arxiv.org/abs/1912.01389>
21. Norwegian Ministry of Local Government and Modernisation: Public participation in planning (2014). https://www.regjeringen.no/contentassets/7fa15b41220849c9adba3eaea28538ec/medvirkning_veileder_engelsk.pdf
22. Papachristos, E., Skov Johansen, P., Møberg Jacobsen, R., Bjørn Leer Bysted, L., Skov, M.B.: How do people perceive the role of AI in human-AI collaboration to solve everyday tasks? In: CHI Greece 2021. Association for Computing Machinery, New York (2021). <https://doi.org/10.1145/3489410.3489420>
23. Russell-Rose, T., Tate, T.: Designing the Search Experience: The Information Architecture of Discovery (2012). https://www.researchgate.net/publication/273544205_Designing_the_Search_Experience_The_Information_Architecture_of_Discovery
24. Samuel, D., et al.: NorBench – a benchmark for Norwegian language models. In: Alumäe, T., Fishel, M. (eds.) Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pp. 618–633. University of Tartu Library, Tõrshavn (2023). <https://aclanthology.org/2023.nodalida-1.61>
25. Saward, G., Hall, T., Barker, T.: Assessing usability through perceptions of information scent, pp. 337–346 (2004). <https://doi.org/10.1109/METRIC.2004.1357919>

26. Spangher, A., Peng, N., May, J., Ferrara, E.: Modeling “newsworthiness” for lead-generation across corpora (2021). <https://arxiv.org/pdf/2104.09653>
27. Touvron, H., et al.: Llama 2: open foundation and fine-tuned chat models (2023). <https://arxiv.org/abs/2307.09288>
28. Udoudom, U., George, K., Igiri, A., Aruku, K.: Media literacy and its implications for the understanding of truth and reality: a philosophical exploration. *Int. J. Multidisc.: Appl. Bus. Educ. Res.* **4**, 4244–4257 (2023). <https://doi.org/10.11594/ijmaber.04.12.08>
29. Vectera: Hallucination leaderboard (2024). <https://github.com/vectera/hallucination-leaderboard>
30. Velagapuri, V., Rekha, S.: Role of information scent and link position in a successful navigation on web. In: Holzinger, A., Ziefle, M., Hitz, M., Debevc, M. (eds.) *SouthCHI 2013. LNCS*, vol. 7946, pp. 447–456. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-39062-3_28



Advancing Financial Text Sentiment Analysis with Deep Learning and Ensemble Models

Wei Liang Russell Tang^(✉) 

University of Manchester, Manchester, UK
russelltangw11@gmail.com

Abstract. Sentiment analysis in financial text plays a crucial role in understanding market trends and predicting financial outcomes. This research investigates the impact of fine-tuning and ensemble learning techniques on the performance of Large Language Models (LLMs) for financial sentiment analysis. We focus on comparing individual fine-tuned FinBERT models and various ensemble methods, particularly stacking ensembles with different meta-learners, on the FiQA (Financial Opinion Mining and Question Answering) 2018 benchmark dataset. Our methodology involves dataset selection, model development, and rigorous evaluation using multiple metrics. The results demonstrate the effectiveness of domain-specific adaptation and the potential benefits of combining multiple models in an ensemble to improve sentiment classification performance. The stacking ensemble with a Random Forest meta-classifier achieves state-of-the-art performance on both datasets, outperforming individual fine-tuned models and other ensemble methods.

1 Introduction and Background

Sentiment analysis within the finance sector has garnered attention for its ability to derive insights from textual data in the financial domain. The purpose of financial sentiment analysis is to recognise and measure the stance or viewpoint conveyed in literature such as news pieces, earnings statements and social media posts [1]. Grasping sentiment can offer data for making investment choices, evaluating risks and forecasting market trends [2]. However, financial sentiment analysis presents unique challenges compared to sentiment analysis in other domains. Financial texts often contain numerous factors expressed in nuanced language such as complex numerical information, finance terminology, and other market-specific factors that can influence sentiment [3]. Additionally, the sentiment expressed in financial texts can be more nuanced and harder to detect compared to general sentiment analysis tasks [2].

The importance of sentiment analysis in finance has been increasingly recognised by both academics and industry professionals. Studies have shown that sentiment expressed in financial news and social media can have a significant impact on stock prices, trading volumes, and market volatility [4, 5]. Furthermore, the rapid expansion of retail investment in recent years has led to the expectation that Financial Sentiment Analysis will become even more critical. This can be seen in the increase in user base and active users

on various retail investment platforms [9, 18]. As more individuals engage in investing and trading activities, the volume of informal financial text data on social media platforms, such as Twitter and Reddit, has surged. This trend is further amplified by the growing viewership of formal financial documents, which leads to these documents holding even more importance when it comes to sentiment analysis.

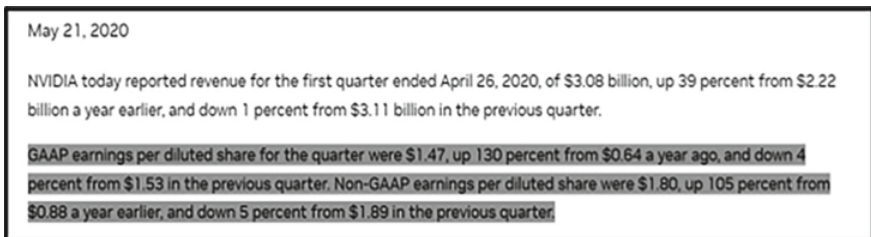
This means that there is an increasing importance of sentiment analysis in making informed financial decisions, as it helps them gauge market sentiment, identify potential risks and opportunities, and react to shifting market dynamics in a timely manner.

As a result, the role of predicting sentiment in Financial Text is expected to become even more critical, serving as a key tool for investors navigating the complex and fast-paced world of financial markets.

2 Problem Definition and Goals

The problem addressed in this research is the challenge of accurately analysing sentiment in financial texts, which is crucial for various applications in finance, including market trend prediction, risk assessment, and investment decision-making. Financial texts often contain domain-specific jargon, complex numerical information, and subtle sentiment expressions that can be difficult for general-purpose sentiment analysis models to interpret correctly. Additionally, the sentiment expressed in financial texts can be more nuanced and harder to detect compared to general sentiment analysis tasks.

2.1 Examples



May 21, 2020

NVIDIA today reported revenue for the first quarter ended April 26, 2020, of \$3.08 billion, up 39 percent from \$2.22 billion a year earlier, and down 1 percent from \$3.11 billion in the previous quarter.

GAAP earnings per diluted share for the quarter were \$1.47, up 130 percent from \$0.64 a year ago, and down 4 percent from \$1.53 in the previous quarter. Non-GAAP earnings per diluted share were \$1.80, up 105 percent from \$0.88 a year earlier, and down 5 percent from \$1.89 in the previous quarter

Excerpt 1. An excerpt Nvidia’s Financial Results for First Quarter Fiscal 2021 [19].

This excerpt includes detailed revenue figures, growth percentages, and a financial term “GAAP gross margin”. Here, both GAAP and non-GAAP earnings per share are mentioned, along with their respective growth percentages compared to previous periods (Excerpts 1 and 2).

The large stock gains this year is positive at >68% but is tempered by the fact the stock is still well below previous highs. Terms like “price war” also make the sentiment more ambiguous. A model may struggle with weighing the competing positive and negative aspects.

Musk warned that the prospect of recession and higher interest rates meant the EV maker could lower prices to sustain growth at the expense of profit. In January, Musk said the price cuts had stoked demand.

Tesla shares have soared more than 68% this year on hopes the company would win the price war it started, although the stock remains more than 50% below its November 2021 peak.

Shares have fallen since Tesla's investor day on March 1 when Musk said little about how soon the EV maker might launch a more affordable, mass-market vehicle.

Excerpt 2. An excerpt from Reuters discussing Tesla Inc's report [20].

Sentiment Analysis can be used to forecast market volatility, market conditions and sector-wide trends. In this section, we will be discussing a few instances in which the use of Sentiment Analysis has signalled volatility and thus acts as a warning to investors using these indicators.

2.2 Problems with Traditional Volatility Indicators

Before NLP breakthroughs, there were other methods of sentiment analysis on the market, and they are Market Indicators. Just to mention a few, Chicago Board Options Exchange's Volatility Index (VIX), Baker and Wurgler's Sentiment Index. It is worth noting that these indicators only show broader market sentiment.

On the other hand, NLP techniques can detect sentiments both towards broader market sentiment, and specific entities (and consequently, stocks), being aware of how major factors like economic news, elections, central bank announcements, and the stage of the economic cycle helps to give informed sentiment information on specific entities.

In the next section, we will discuss 2 case studies that elaborate on the limitations of traditional broad volatility indicators.

2.3 Case Studies

To illustrate the drawbacks of Market Indicators only being able to detect broad sentiments, I have assembled 2 case studies where the price of the Investment Products has moved drastically, and we will be exploring them below:

In Graph 1, during the 2008 crisis, the VIX index (market volatility) and MACD (Moving Average Convergence Divergence) were in sync, indicating a bearish market trend. The extremely high VIX reflected intense fear and pessimism, confirming the bearish MACD crossover. Investors using both tools could have avoided long positions and potentially profited from the market decline.

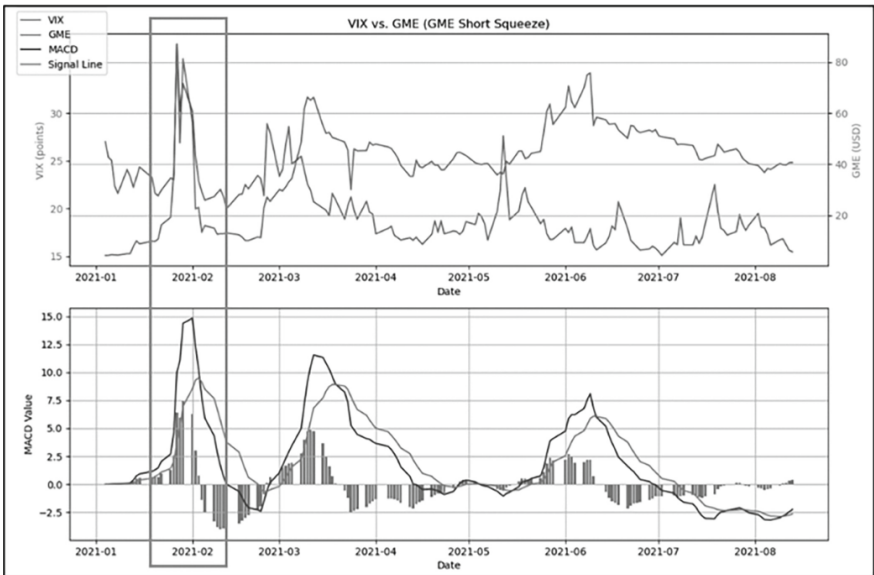
In contrast with Graph 2, during the GameStop Short Squeeze, the VIX contradicted the stock's extreme price increase driven by retail investor sentiment on social media. Traditional indicators failed to fully capture the dynamics at play, highlighting the value of NLP sentiment analysis on social media data.

2.4 Potential Use Cases

The proposed approach for financial sentiment analysis has several potential real-world applications in the financial industry:



Graph 1. The graph depicts VIX, S&P 500 index price, and MACD.



Graph 2. The graph depicts VIX, GameStop stock price, and MACD.

- Investment decision support: The sentiment predictions generated by the model can be used to inform investment decisions by providing insights into market sentiment and potential price movements.

- **Risk assessment:** By analysing the sentiment of financial news and company reports, the model can help financial institutions assess the risk associated with investments or market segments.
- **Trading strategies:** The sentiment scores produced by the model can be incorporated into algorithmic trading strategies to make more informed buy or sell decisions.
- **Financial monitoring:** Regulatory bodies and financial institutions can use the model to monitor sentiment in financial markets and detect potential market manipulations or anomalies.

These real-world applications highlight the practical relevance of the research and its potential to support better decision-making processes in the financial industry.

3 Related Work

The field of Natural Language Processing (NLP) has seen significant advancements with the introduction of transformer-based models. In this section, we will discuss how different transformer-based models have been adapted to address the challenge of financial sentiment analysis, which our ensemble model then aims to improve upon.

3.1 Related Models

- **FinBERT**

This model is an open-source financial language model based on the BERT architecture, has gained widespread acceptance within the research community for analysing sentiments in the financial domain [2, 3]. FinBERT uses a two-stage training process: further pre-training on financial corpora, followed by fine-tuning on financial sentiment datasets. This approach allows FinBERT to capture finance-specific language nuances and sentiment expressions more effectively than general-purpose BERT models [6]. The first stage utilises a large collection of financial texts, such as the financial section of the Reuters TRC2 dataset, while the second stage focuses on fine-tuning using labelled datasets like the Financial PhraseBank. This domain-specific adaptation has shown significant improvements in financial sentiment analysis tasks.

- **FinMA**

Based on the LLaMA architecture, this model introduces a multi-task instruction fine-tuning approach to financial language modelling [4]. Unlike FinBERT, which focuses primarily on sentiment analysis, FinMA is trained on a diverse set of financial NLP tasks, including named entity recognition and question answering. The key innovation of FinMA is its ability to handle both textual and numerical data simultaneously, allowing it to capture complex interactions between language and financial information. This integrated approach provides a more comprehensive view of sentiment expressed in financial texts and enables FinMA to adapt quickly to new sentiment analysis tasks with minimal training data. However, FinMA has shown limitations in tasks requiring strong quantitative reasoning abilities, highlighting the challenges of balancing multiple financial NLP tasks within a single model.

- **FinGPT**

This is an efficient model adaptation for financial sentiment analysis by leveraging the GPT architecture and employing low-rank adaptation techniques [17]. The main innovation of FinGPT lies in its use of Low-Rank Adaptation (LoRA), which significantly reduces the memory footprint and computational cost of fine-tuning large language models. This approach allows FinGPT to be efficiently adapted to finance-specific tasks using relatively small datasets, making it more accessible for researchers and practitioners with limited computational resources. FinGPT's strong few-shot learning capabilities and competitive performance on financial sentiment benchmarks demonstrate the potential of efficient adaptation techniques in specialised domain tasks.

3.2 Motivation for Addressing Limitations

While these models have shown impressive results in financial sentiment analysis, they each have their strengths and limitations. Many existing models in the market rely on single models, which can lead to overconfidence in predictions, especially when dealing with diverse financial texts. Single models may struggle to accurately analyse sentiment across different types of financial language, such as formal financial reports versus informal social media posts and may face challenges in integrating quantitative data present in texts.

This research aims to address these limitations through ensemble approaches that leverage the complementary strengths of multiple models. By experimenting with ensembles with its base models trained on a different type of financial text, we create specialists for various domains of financial language. In voting ensembles, we hypothesise that the different weighting mechanisms will yield different performance results, and different meta-classifiers in our Stacking Ensembles can potentially assign different vote weights to each base model depending on the type of financial text being analysed. This approach aims to mitigate the issues of single-model approaches, providing more robust and accurate sentiment analysis across diverse financial texts.

4 Methodology

4.1 Base Model Selection for Building Ensembles

Our model selection process for this research considered baseline performance on similar text analytics tasks, models used in similar research, and model size. FinBERT [2] was selected for our experiments as its performance on sentiment analysis tasks, while decent, was not as accurate as state-of-the-art models like FinMA and leaves room for improvements in prediction through the application of ensembles thus allowing us to observe if there are performance gains through ensemble learning.

4.2 Training Pipeline and Training Datasets

To create these specialist base models for the various types of financial language specified earlier, we fine-tune the FinBERT models [5] on the selected datasets.

The datasets chosen should aim to expose the model to a wide range of financial text types and sources. There are two key aspects we want to focus on when selecting datasets for training sentiment analysis models in finance: quantification and understanding [10, 16]. Quantification is crucial because financial texts often contain numerical data, such as stock prices, financial ratios, and percentage changes, which play a significant role in conveying sentiment and impact. A model trained on datasets rich in quantitative data learns to associate numerical values with sentiment labels, capturing the magnitude and direction of sentiment. Understanding is equally important, as financial sentiment is expressed through complex language, domain-specific jargon, and nuanced opinions. Datasets that promote understanding should cover a wide range of financial topics, entities, and events, enabling the model to grasp the underlying meaning and intent behind the sentiment. Described below are the training datasets used.

- Financial PhraseBank (FPB) [11] consists of financial text annotated with sentiment labels. This dataset has been widely used for fine-tuning LLMs on financial sentiment analysis tasks. The annotated sentences cover a range of financial entities and provide valuable examples for training models to predict sentiment polarities.
- Kaggle-financial-sentiment [12] contains a significant amount of textual quantitative data, providing the model with exposure to numerical and financial metrics alongside sentiment labels. Training on this dataset enables the model to understand the interplay between quantitative information and sentiment, a crucial aspect of financial sentiment analysis.
- FinanceInc/auditor_sentiment [13] consists of several thousand sentences from English language financial news. This dataset offers a diverse range of financial text from news sources, allowing the model to learn the language and sentiment patterns prevalent in financial news articles.
- Zeroshot/twitter-financial-news-sentiment [14] brings in the perspective of social media sentiment related to financial news. Training on this dataset allows the model to capture the unique characteristics of sentiment expressed in short, informal text snippets, such as tweets. This exposure enhances the model's adaptability to various forms of financial sentiment expression.

These diverse sources capture different aspects of financial sentiment, from formal to informal financial text data, and from market reactions to company-specific events and broader economic trends ensures that the model can handle the complexities and nuances of financial sentiment across different domains.

4.3 Ensembles

Ensemble learning is a technique that combines different models together to obtain better predictive performance than could be obtained from any of the constituent learning algorithms alone [7]. In the context of financial text sentiment analysis, ensemble learning could potentially help to improve the accuracy and robustness of sentiment classification

models by leveraging the strengths of different base learners [8]. We explored ensemble learning approaches like unweighted voting and stacking with different meta-learners.

- For the unweighted voting ensemble, we combined the predictions of the fine-tuned FinBERT models using a simple majority voting scheme, where each model’s prediction was given equal weight.
- For the stacking ensemble, we used the predictions of the fine-tuned FinBERT models as input features to train a meta-learner. We experimented with three different meta-learners: Random Forest, Logistic Regression, and Gradient Boosting. The meta-learners were trained on the concatenated predictions of the base models, along with their corresponding confidence scores.

To handle the class imbalance present in the datasets, we calculated class weights based on the inverse class frequencies. These class weights assigned higher importance to the minority classes and were used in the training of the Logistic Regression and Gradient Boosting meta-learners.

4.4 Target Task

To create a comprehensive testing environment, Part 1 of the FiQA 2018 dataset [15] is used for our test data. It was introduced as part of the “The Financial Opinion Mining and Question Answering” challenge [16], consisting of both financial news articles and numerical data. By incorporating this dataset, it enables an accurate assessment of our sentiment analysis model’s performance on a wide range of financial text data, resulting in an accurate assessment of the performance of our sentiment analysis model.

To create a comprehensive testing environment, we selected Part 1 of the FiQA 2018 dataset [15] for our testing. It was introduced as part of the “The Financial Opinion Mining and Question Answering” challenge [16], including both financial news articles and numerical data. By incorporating this dataset, our testing is robust and enables an accurate assessment of our sentiment analysis model’s performance on a wide range of financial text data, resulting in an accurate assessment of the performance of our sentiment analysis model.

5 Results and Analysis

5.1 Model Performance Table

(See Table 1).

Table 1. ** All experiments were conducted using the same hyperparameters of 12 layers and 768 hidden units, a learning rate of $2e-5$, a batch size of 32, and for 3 epochs.

Model	Metric	FIQA 2018	Datasets used for training
FinMA	F1 Score	0.825	Financial PhraseBank (FPB),
	Recall	0.705	FiQA-SA dataset
FinBERT	F1 Score	0.559	Reuters TRC2,
	Recall	0.488	Financial PhraseBank
FinBERT tuned with 4 datasets	F1 Score	0.850	FinBERT training,
	Recall	0.844	4 datasets mentioned earlier
FinBERT tuned with FPB	F1 Score	0.503	FinBERT training,
	Recall	0.418	FPB
FinBERT tuned with kaggle-financial-sentiment	F1 Score	0.816	FinBERT training,
	Recall	0.826	chiapudding/kaggle-financial-sentiment
FinBERT tuned with twitter-sentiment	F1 Score	0.490	FinBERT training,
	Recall	0.416	zeroshot/twitter-financial-news-sentiment
FinBERT tuned with auditor-sentiment	F1 Score	0.428	FinBERT training +
	Recall	0.359	FinanceInc/auditor_sentiment
FinBERT Ensemble: unweighted voting	F1 Score	0.572	4 instances of FinBERT, each trained on 1 of
	Recall	0.491	the 4 datasets respectively.
FinBERT Ensemble: Stacked with Random Forest as Meta Classifier	F1 Score	0.985	4 instances of FinBERT, each trained on 1 of
	Recall	0.985	the 4 datasets respectively.
FinBERT Ensemble: Stacked with Logistic Regression as Meta Classifier	F1 Score	0.802	4 instances of FinBERT, each trained on 1 of
	Recall	0.807	the 4 datasets respectively.
FinBERT Ensemble: Stacked with Gradient Descent as Meta Classifier	F1 Score	0.771	4 instances of FinBERT, each trained on 1 of
	Recall	0.764	the 4 datasets respectively.

5.2 Analysis of Model Performance

Analysing the single model performances, it is expectedly lower than the Ensemble Models, with the highest accuracy of **0.850** achieved by the FinBERT tuned with 4 datasets. Turning our attention to the ensemble methods, we see a notable improvement in performance. The **stacking ensemble with a Random Forest meta-classifier** achieves the highest accuracy of **0.985** on FiQA 2018, significantly outperforming all individual models. This demonstrates the power of ensemble learning in leveraging the strengths of multiple models and mitigating their weaknesses.

The central aim of the experiment setup is to compare two approaches: Fine-tuning a single instance of FinBERT on all four datasets combined, against an ensemble model of four separate instances of fine-tuned FinBERT instances, with a meta-classifier on top. This experimental setup is particularly interesting as it explores the impacts of how different Machine Learning methodologies with essentially the same datasets, can affect prediction performance. By comparing the two approaches, we can gain valuable insights into the effectiveness of fine-tuning strategies against ensemble strategies. The performance results highlight the potential of ensemble learning in improving robustness and generalisation, especially for the challenging dataset for FiQA 2018, and can help with performance in quantitative understanding.

5.3 Error Analysis

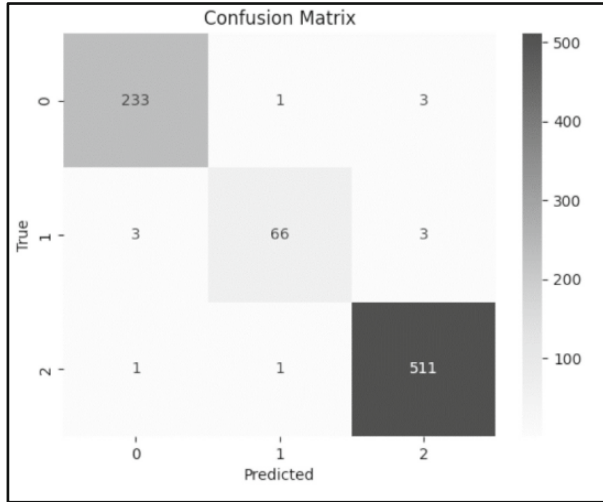


Fig. 1. Confusion Matrix for Stacked Ensemble with Random Forest meta-learner tested against FiQA-2018.

This Confusion Matrix summarises the model’s performance by comparing the predicted labels against the true labels. The matrix shows that the model correctly classified most instances for each sentiment class. The diagonal elements represent the true positives: 233 for class 0 (negative sentiment), 66 for class 1 (neutral sentiment), and 511 for class 2 (positive sentiment) (Fig. 1).

Overall, the confusion matrix demonstrates the model’s high accuracy in sentiment classification across all three classes. The misclassifications are relatively low, indicating the model’s robustness in capturing the sentiment expressed in financial text data (Fig. 2).

Classification Report:					
	precision	recall	f1-score	support	
0	0.98	0.98	0.98	237	
1	0.97	0.92	0.94	72	
2	0.99	1.00	0.99	513	
accuracy			0.99	822	
macro avg	0.98	0.97	0.97	822	
weighted avg	0.99	0.99	0.99	822	

Fig. 2. Class wise performance for Stacked Ensemble with Random Forest meta-learner tested against FiQA-2018.

The high precision and recall values for each class indicate that the model is capable of correctly identifying and classifying instances within each sentiment category.

All classes perform relatively well, having F1 scores of 0.98, 0.94 and 0.99 respectively.

5.4 Comparing Meta-learners in Stacking Ensembles

The experimental results show that the **Random Forest** meta-classifier achieves the **highest accuracy** scores on the FiQA 2018 dataset in stacking ensembles, with an accuracy of 0.985. **Logistic Regression** also demonstrates strong performance, achieving 0.802 on FiQA 2018 and **Gradient Descent** achieves lower accuracies of 0.771 on FiQA 2018. Random Forest's superior performance can be attributed to its ability to handle complex relationships and its resistance to overfitting. Logistic Regression effectively models the relationship between base models' predictions and sentiment labels using a linear combination. Gradient Descent's lower performance may indicate challenges in optimization with class imbalance.

Overall, the experimental results demonstrate the effectiveness of stacking ensembles in handling imbalanced datasets for sentiment analysis tasks and Random Forest's superior performance in this circumstance. The choice of meta-learner should consider dataset characteristics, problem complexity, and interpretability requirements.

6 Conclusion

In this research, we investigated the impact of using various Machine Learning techniques on the performance of predicting sentiment in Financial Texts. Our results demonstrate that using an Ensemble Model with fine-tuned FinBERT models on domain-specialised datasets enhances its ability to capture the nuances of financial language and sentiment. Furthermore, the stacking ensemble approach with a Random Forest meta-learner achieves state-of-the-art performance on benchmark datasets, outperforming individual fine-tuned models and other ensemble methods such as unweighted and weighted voting as well as other meta-classifiers.

The experimental findings highlight the importance of leveraging diverse financial datasets in an ensemble, as it allows the model to learn from a wide range of text types, sources, and sentiment expressions and it is represented in a way that minimises overconfidence. The ensemble learning technique, particularly stacking with meta-learners, proved to be effective in combining the strengths of multiple fine-tuned models and mitigating their weaknesses, leading to improved sentiment classification performance.

6.1 Potential Shortcomings

- While our ensemble method demonstrates superior performance, it's important to acknowledge the increased computational requirements compared to single-model approaches. This raises valid concerns about energy efficiency and computational costs. However, it's crucial to consider the context of its application. In large financial institutions, where this model would likely be deployed, the analysis would be run centrally and its insights distributed to numerous traders and analysts. This economy of scale significantly improves its cost-effectiveness. The value derived from more

accurate sentiment analysis, potentially influencing decisions on multi-million dollar trades or investments, could far outweigh the computational costs. Nevertheless, we recognize the need for further research into quantifying this cost-benefit ratio across different use cases. Future work should focus on optimising the ensemble's efficiency without compromising its performance, possibly through techniques like model distillation or more efficient ensemble architectures. Additionally, as green computing advances, the environmental impact of such models may decrease, further improving their viability.

- Our current model focuses on sentiment analysis at the sentence level, which, while crucial, is just one step towards comprehensive document-level sentiment analysis. We acknowledge that financial documents, such as lengthy reports or regulatory filings, require a more nuanced approach that considers the overall context and the relationships between sentences. However, achieving high accuracy at the sentence level is a fundamental building block for more complex analyses.

6.2 Future Research

Future research should explore hierarchical models that can effectively aggregate sentence-level sentiments into coherent document-level predictions. This could involve techniques such as attention mechanisms to weigh the importance of different sentences, or recurrent architectures to capture the flow of sentiment throughout a document. Additionally, investigating how to incorporate document structure, such as sections and subsections in financial reports, could provide valuable context for more accurate sentiment predictions. As we advance towards document-level analysis, maintaining the high accuracy achieved at the sentence level will be crucial for building reliable and interpretable models for longer financial texts.

While we conducted preliminary experiments to determine the hyperparameters for fine-tuning FinBERT and training the meta-learners, a more rigorous and systematic hyperparameter search could potentially uncover optimal settings that further boost performance. Techniques such as grid search, random search, or Bayesian optimization could be employed to efficiently explore the hyperparameter space and identify the best configurations for each component of the pipeline.

The research primarily focused on comparing individual fine-tuned models with parallel ensemble architectures, where the base models are trained independently, and their predictions are combined through voting or stacking. An interesting avenue for future work is to investigate the effects of serial ensemble architectures, where the base models are trained sequentially, with each model learning from the errors of its predecessors. This iterative approach, often referred to as boosting, has the potential to create a stronger ensemble by progressively focusing on the more challenging instances and refining the overall prediction.

Moreover, the current research can be extended by exploring the integration of additional financial datasets, both for fine-tuning and evaluation purposes. Incorporating a wider variety of financial text sources, such as earnings call transcripts, analyst reports, and social media data, would expose the model to an even broader range of sentiment expressions and potentially improve its ability to generalise across different domains and writing styles.

In conclusion, this study demonstrates the effectiveness of ensemble learning techniques in enhancing the performance of financial sentiment analysis models. The proposed approach, centred around fine-tuned FinBERT models and stacking ensembles, achieves state-of-the-art results on benchmark datasets. However, there remain several promising avenues for future research. By addressing these aspects, we can further push the boundaries of financial sentiment analysis and develop even more accurate, robust, and reliable models to support decision-making in the financial domain.

Acknowledgments. I would like to thank my project supervisor, Professor Sophia Ananiadou from the University of Manchester for mentoring my project as well as the researchers Jimin Huang and Qianqian Xie, leading the Fin AI group, for our discussions and helping with my understanding of the topic for this paper.

Appendix

Model Evaluation Python Implementation

```
import pandas as pd
from transformers import pipeline, AutoModelForSequenceClassification, AutoTokenizer
from sklearn.metrics import accuracy_score, f1_score, precision_score, recall_score

# Prepare sentiment analysis pipeline
sentiment_analyzer = pipeline('sentiment-analysis', model=model, tokenizer=tokenizer)

def evaluate_model(df):
    # Get predictions
    predictions = [int(sentiment_analyzer(text)[0]['label']).split('_')[-1] for text in df['text']]

    # Calculate metrics
    accuracy = accuracy_score(df['sentiment'], predictions)
    f1 = f1_score(df['sentiment'], predictions, average='weighted')
    precision = precision_score(df['sentiment'], predictions, average='weighted')
    recall = recall_score(df['sentiment'], predictions, average='weighted')

    return accuracy, f1, precision, recall
```

References







1. Malo, P., Sinha, A., Korhonen, P., Wallenius, J., Takala, P.: Good debt or bad debt: Detecting semantic orientations in economic texts. *J. Assoc. Inf. Sci. Technol.* **65**(4), 782–796 (2014). <https://doi.org/10.1002/asi.23062>
2. Araci, D.: FinBERT: Financial sentiment analysis with pre-trained language models. arXiv preprint [arXiv:1908.10063](https://arxiv.org/abs/1908.10063) (2019)
3. Xie, Q., et al.: BUFIN: a large-scale financial dataset for blockchain and DeFi research. arXiv preprint [arXiv:2306.05443](https://arxiv.org/abs/2306.05443) (2023)
4. Nyakurukwa, K., Seetharam, Y.: Can investor sentiment predict cryptocurrency returns? *Evid. Bitcoin. Sci. Afr.* **20**, e01596 (2023). <https://doi.org/10.1016/j.sciaf.2023.e01596>
5. Bollen, J., Mao, H., Zeng, X.: Twitter mood predicts the stock market. *J. Comput. Sci.* **2**(1), 1–8 (2011). <https://doi.org/10.1016/j.jocs.2010.12.007>
6. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. arXiv preprint [arXiv:1810.04805](https://arxiv.org/abs/1810.04805) (2018)
7. Dietterich, T.G.: Ensemble methods in machine learning. In: Kittler, J., Roli, F. (eds.) *Multiple Classifier Systems*. LNCS, vol. 1857, pp. 1–15. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-45014-9_1

8. Nti, I.K., Adekoya, A.F., Weyori, B.A.: A systematic review of fundamental and technical analysis of stock market predictions. *Artif. Intell. Rev.* **53**, 3007–3057 (2020). <https://doi.org/10.1007/s10462-019-09754-z>
9. Airnow: Monthly number of active users selected leading apps that allow for online share trading worldwide from January 2017 to July 2021, by app (in 1,000s) [Graph]. Statista (2021). <https://www-statista-com.manchester.idm.oclc.org/statistics/1259822/global-etradng-app-monthly-active-users/>
10. Chen, W., Li, M., Wang, X., Zou, Y.: FinGPT: instruction-following financial large language models. arXiv preprint [arXiv:2310.15205](https://arxiv.org/abs/2310.15205) (2023)
11. Malo, P., Sinha, A., Takala, P., Korhonen, P., Wallenius, J.: FinancialPhraseBank-v1.0 (2013). https://www.researchgate.net/publication/251231364_FinancialPhraseBank-v10
12. Chiapudding: Kaggle Financial Sentiment. Hugging Face Datasets (2024). <https://huggingface.co/datasets/chiapudding/kaggle-financial-sentiment>
13. FinanceInc: Auditor Sentiment Fine-tuned. Hugging Face Datasets (2024). <https://huggingface.co/datasets/FinanceInc/auditor-sentiment-finetuned>
14. Zeroshot: Twitter Financial News Sentiment. Hugging Face Datasets (2024). <https://huggingface.co/datasets/zeroshot/twitter-financial-news-sentiment>
15. Maia, M., et al.: WWW'18 open challenge: financial opinion mining and question answering. In: Companion Proceedings of the web Conference 2018, pp. 1941–1942. International World Wide Web Conferences Steering Committee, Geneva (2018). <https://doi.org/10.1145/3184558.3192301>
16. Chen, Z., Shang, T., Chen, Y., Zhao, L.: FinSense: an assistant system for financial journalists and investors. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pp. 3697–3711. Association for Computational Linguistics (2021). <https://doi.org/10.18653/v1/2021.emnlp-demo.9>
17. Xie, Q., Han, W.: FinGPT: open-source financial large language models. arXiv preprint [arXiv:2402.12659](https://arxiv.org/abs/2402.12659) (2024)
18. Interactive Brokers: Number of accounts of Interactive Brokers from Jan 2008 to October 2021 (in 1,000s) [Graph]. Statista (2022). <https://www-statista-com.manchester.idm.oclc.org/statistics/1263318/interactive-brokers-number-accounts/>
19. NVIDIA Corporation: NVIDIA Announces Financial Results for First Quarter Fiscal 2021. NVIDIA Newsroom (2020). <https://nvidianews.nvidia.com/news/nvidia-announces-financial-results-for-first-quarter-fiscal-2021>
20. Sriram, K., Jin, H.: Tesla posts record quarterly deliveries after price cuts, up 4% from Q4. Reuters (2023). <https://www.reuters.com/business/autos-transportation/tesla-misses-first-quarter-delivery-estimates-2023-04-02/>

Other AI Applications



Explaining a Staff Rostering Problem Using Partial Solutions

GianCarlo A. P. I. Catalano¹ , Alexander E. I. Brownlee¹ ,
David Cairns¹ , John A. W. McCall² , Martin Fyvie² ,
and Russell Ainslie³ 

¹ University of Stirling, Stirling, UK

{g.a.catalano,sandy.brownlee}@stir.ac.uk, dec@cs.stir.ac.uk

² Robert Gordon University, Aberdeen, UK

{j.mccall,m.fyvie1}@rgu.ac.uk

³ BT Technology, Applied Research Department, Ipswich, UK
russell.ainslie@bt.com

Abstract. There are many critical optimisation tasks that metaheuristic approaches have been shown to be able to solve effectively. Despite promising results, users might not trust these algorithms due to their intrinsic lack of interpretability. This paper demonstrates the use of **explainability** to resolve this issue by producing human-interpretable insights that focus on **simplicity, fitness and linkage**.

Our explainability approach revolves around the concept of Partial Solutions, which assist in breaking up the solutions of optimisation problems into smaller components. We first expand upon our previous research proposing the technique, and then provide a use case on the Staff Rostering task: a large and otherwise uninterpretable optimisation problem with ethical implications due to its direct impact on humans. The explanations consist in rota assignments for interacting groups of workers, along with the reasons why they are interacting. Lastly, some experiments are used to ascertain that the algorithms work as intended and for hyperparameter tuning.

The results suggest that our methodology is capable of presenting insightful information for the Staff Rostering problem, by producing both **local explanations** of solutions and **global explanations** of the problem definition.

Keywords: Explainability · XAI · Job Scheduling · Metaheuristics

1 Introduction

Optimisation problems appear in many real-world applications, and often can only be efficiently solved using metaheuristic methods such as Genetic Algorithms (GAs). Important examples are found in medicine [9, 16], with many other applications ranging from logistics to engineering [19].

Considering the impact metaheuristics can have, their adoption requires users to **trust** them for both ethical and legal reasons, as discussed in recent European Regulations [11]. Metaheuristics are particularly hard to trust due to their non-deterministic behaviour and often uninterpretable trial-and-error strategy.

This paper will demonstrate the use of **Partial Solutions** (PSs) [3] as a potential explainability tool for many metaheuristics that is able to act as an **interpretable** intermediary between the user and the solutions to the problem.

Following the definitions of [1,14] we consider **interpretable** to mean the inherent ability to explain or to convey meaning in understandable terms to a human, and **explainability** meaning techniques to provide details or reasons which make the algorithm's functioning clear or easy to understand. In this paper the explanations will clarify the results of a metaheuristic after its execution, making them post-hoc.

A Partial Solution represents a **trait associated with fitter solutions**. In [3], the structure of PSs is defined, as well as how to obtain them. The core principle is that while a large solution might not be interpretable by the user, the smaller PSs it contains can be more digestible while still conveying important information. This aligns with the idea proposed in [17] that an explanation should be **meaningful** towards the user's goal, which in the case of an optimisation problem mainly relates with improving the fitness value.

This paper aims to demonstrate the applicability of PSs to a complex, real-world problem, extending the work presented in [3]. We present a use case on a Staff Rostering problem as an example of a real task benefitting from explainability, derived from a real problem encountered by our industrial partner. Within this context our research questions are:

- **Mining: How can PSs be obtained for our case study problem?**
- **Explainability: How can PSs be used for explainability?**
- **Insights: What insights do our methods produce for the Staff Rostering problem?**

The document proceeds as follows: Sect. 2 is a literature review regarding PSs and explainability, Sect. 3 presents our methodology, Sect. 4 defines the Staff Rostering problem, and Sect. 5 is a use case of our methodology on the problem. Finally, Sect. 6 contains the experiments used to tune the parameters of our algorithms and Sect. 7 summarises our findings.

2 Literature Review

2.1 Partial Solutions, Explanation and Innovization

Partial Solutions, introduced in [3], are designed to provide explanations to the solutions of combinatorial optimisation tasks. Given an optimisation problem and the metaheuristic used to solve it, our method consists of an algorithm that analyses the candidate solutions produced by the metaheuristic (which we call the **reference population**) and produces a list of traits which are associated with desirable fitness, the **PSs** (Fig. 1).

Each of these traits is composed of a subset of the parameters, and the values they should take, such as the pattern *4**1*3.

In order to find the patterns that are most suited for explanations, we use 3 metrics to assess them: **simplicity**, **mean fitness** and **atomicity** (derived from the concept of **linkage**). Simplicity is the number of “wildcards” in the pattern: more wildcards make the pattern simpler and therefore easier to understand. Mean fitness is the average fitness of the solutions containing the pattern, capturing its contribution to solution quality. Atomicity is a measure of the strength of interaction between the non-wildcard variables, to punish patterns which involve “unnecessary” variables.

In the explainability context, PSs align with the aims of “**innovization**” [4] as positive traits that a human might find useful. We present in this paper the methods which can be used to discover such traits (improving upon our previous work in [3]), as they were noted in [4] to be absent.

2.2 Metaheuristics and Explainability

We follow the terminology used in XAI for machine learning where **explanations** are categorised as **global** (applicable to the general problem) and **local** (relating to specific solutions) [1]. The methods presented in [6–8] by Fyvie *et al.* closely resemble our line of enquiry in that they produce global explanations, whereas our work implements both global and local explanations which are metaheuristic-agnostic. There are relatively few works that produce local explanations of this kind, although model agnostic approaches might be applicable (such as LIME [18]). Explainability can be thought of as presenting higher-order information, and machine learning methods lend themselves well because they contain models of generalised information which can then be analysed. A traditional GA on the other hand has no such model, and thus the literature for explainability in GAs and similar metaheuristics is relatively scarce.

The methods of Fyvie *et al.* introduced the idea of analysing the **trajectory** taken by the population of a GA to obtain a post-hoc explanation for the behaviour of the metaheuristic. Similarly, our approach uses the candidate

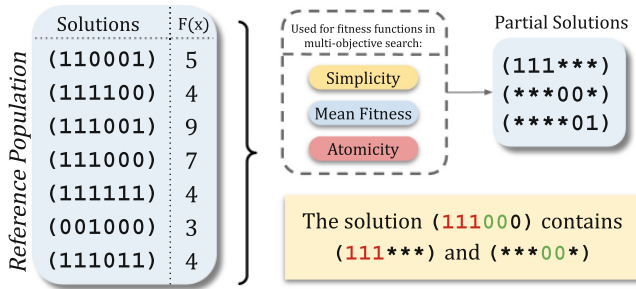


Fig. 1. The reference population, consisting of solutions, is used to construct the objective functions used to search for Partial Solutions.

solutions produced by the metaheuristic to find useful PSs that can be used to construct explanations. In our case, our industrial partner developed a Simulated Annealing (SA) algorithm specifically for their Staff Rostering problem [5], and our proposed system can make use of its outputs with little modification.

3 Methodology

In this section we present our methodology to find PSs, with Sect. 3.1 and Sect. 3.2 summarising and expanding the techniques first presented in [3]. The general process starts by generating a reference population via Simulated Annealing, and using it to find the PSs via the PS Mining algorithm. Finally, **local explanations** for a solution are based on the PSs it contains, and **global explanations** are obtained by analysing all of the PSs.

3.1 Reference Population

The methodology described in [3] defines how the input data (the reference population) is used to evaluate the PSs, and this section will describe how it is obtained. In order to analyse the behaviour of an algorithm, we consider all of the candidate solutions it produces. This includes the starting population and all of its subsequent improvements and resembles the strategy used by *Fyvie et al.* in [8]. In order to distinguish positive and negative traits, it is necessary to evaluate both accurately. The input data will under-represent negative traits (due to convergence by the metaheuristic), and the PSs representing these negative patterns might not be evaluated accurately due to lack of samples. To mediate this issue, we can introduce solutions obtained through uniform random sampling, specifically by having half the population derive from the metaheuristic and the other half being random. The benefit of this approach is tested in Sect. 6.

The metaheuristic used in this work is a form of Simulated Annealing developed by our industrial partner, documented in [5]. Since SA could be swapped for any other approach that produces candidate solutions, our system is fairly “algorithm agnostic”, resembling “model agnostic explanations” in XAI [18].

3.2 Partial Solution Mining

In [3] we described a method which, given a list of candidate solutions, obtains the PSs to be used for explainability. The task of finding the PSs is stated as an optimisation problem in itself, where the search space is that of PSs (same as the original problem’s search space but the * symbol is also available to any variable). Uniform mutation is used (each variable has a $1/n$ chance of mutation) where non-wildcard variables have a 50% chance of turning into a wildcard when mutated, and binary uniform crossover is used.

It is a multi-objective search problem, where the objectives to be maximised are **simplicity**, **mean fitness** and **atomicity** as introduced in Sect. 2.1.

As previously mentioned, atomicity represents the strength of the interactions between the variables which are not wildcards in the PS. This is important so that all the variables in the pattern are necessary and contributing to the fitness improvement. The formula proposed in [3] appeared to be unstable for larger problems, and was here improved in Eq. 2 by being restated in terms of pairwise linkage (Eq. 1) as defined in DSMGA-II [12]. While in [3] the three objectives were aggregated into one by normalising and summing them, here they will be handled as separate objectives by a **Multi-Objective** GA such as NSGA-III and MOEAD.

$$\text{link}(A, B) = \sum_{\substack{a \in A \\ b \in B}} p(a, b) \cdot \log \left(\frac{p(a, b)}{p(a) \cdot p(b)} \right) \quad (1)$$

$$\text{atomicity}(x^\psi) = \text{avg} \left(\left\{ \text{link}(A, B) \mid x_A^\psi \neq *, x_B^\psi \neq *, A \neq B \right\} \right) \quad (2)$$

In order to comprehensively explain solutions, it is important to find many diverse PSs. Most Multi-Objective GAs implement an **objective-space** diversity enhancing mechanism, but in this case we also want **decision-space** diversity. In our work we opted for **sequential decision-space crowding** where multiple GA runs are executed and each run's crowding metric is based on the results from previous runs, known as the **archive**.

Decision space crowding is implemented using Eq. 3, derived from fitness sharing [10, 13] with the distance metric in Eq. 4 which counts the proportion of shared fixed variables. Here we use $\sigma_{\text{shared}} = 0.5$, so that PSs are considered too similar at the 50% threshold.

$$C_{\text{archive}}(x^\psi) = \frac{|\{y^\psi \mid y^\psi \in \text{archive}, d(x^\psi, y^\psi) < \sigma_{\text{shared}}\}|}{|\text{archive}|} \quad (3)$$

$$d(x^\psi, y^\psi) = \frac{|\{x_i^\psi \mid x_i^\psi \in x^\psi, x_i^\psi \neq *, x_i^\psi = y_i^\psi\}|}{\text{avg}(\text{order}(x^\psi), \text{order}(y^\psi))} \quad (4)$$

$$\text{where } \text{order}(x^\psi) = |\{x_i^\psi \mid x_i^\psi \in x^\psi, x_i^\psi \neq *\}|$$

3.3 Explaining Partial Solutions

In order to improve the interpretability of Partial Solutions, we can identify **problem specific descriptors** that can help the user understand them at a glance. More specifically, these are quantitative descriptors relating to the optimisation task, that the system can formally analyse. In the case of the Staff Rostering problem, a PS consists of a group of workers and their rota assignments, and some relevant descriptors might be the similarity of their working patterns or the similarity of their skill sets (more are described in Sect. 4).

Generally, it is useful to know whether a descriptor is comparatively high or low, or in more formal terms the position of the descriptive value within its

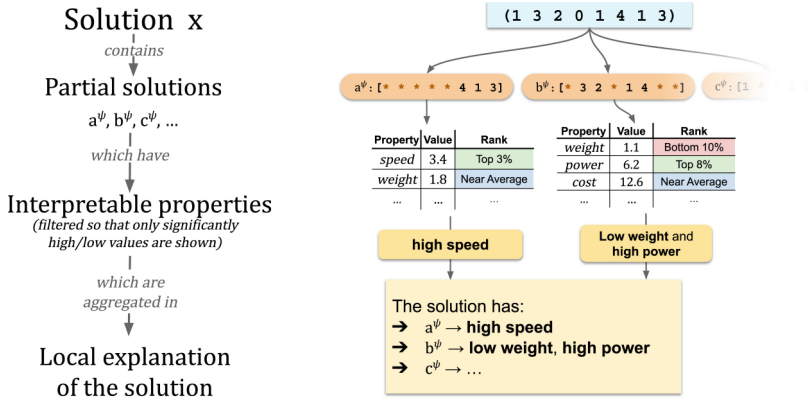


Fig. 2. A local explanation of a solution is composed of the PSs it contains and their descriptors, which are the properties with near-extreme values.

distribution (as defined in Algorithm 1), which we call the **percentile**. Once the descriptors and their polarities are obtained, the PSs are used in the following:

- **Local Explanations:** (Fig. 2) Given a solution, we find the PSs it contains and present them alongside their descriptors, ordered by ‘percentile’
- **Global Explanations:** All the PSs obtained are analysed statistically to obtain information about the overall solution set

Given a PS and a descriptor value, we use Algorithm 1 to determine its percentile by using the **empirical distribution function**. For example, values which are in the upper quartile correspond to a percentile greater than 0.75. The percentile is influenced by the distribution of the descriptor, which means that it is necessary to get a set of control PSs to compare the value against, and this is implemented by the function GET_CONTROL_DISTRIBUTION in Algorithm 1. It returns a set of randomly generated PSs which have the same quantity of fixed variables as the Partial Solution being evaluated, so that any bias produced by the number of active variables is represented in the distribution. For this work, generating 1000 samples was found to be sufficient.

In our system, only descriptor values which are in the bottom 10% and top 10% of their distribution are considered significantly extreme and are presented to the user. This threshold was selected since higher percentages would have included less impactful or potentially irrelevant descriptors.

4 The Staff Rostering Problem

The Staff Rostering problem involves assigning rota patterns to a set of workers, with the aim of having a consistent quantity of workers for a certain calendar period. The problem definition consists of:

Algorithm 1. `get_percentile_of_property(ps, prop : PS → ℝ)`

```

1: function GET_CONTROL_PARTIAL_SOLUTIONS(ps, sample.size)
2:   control_items ← []
3:   for i ← 1 to sample.size do
4:     control_items.append(random_ps_with_same_order_as(ps))
5:   end for
6:   return control_items
7: end function

8: control_items ← GET_CONTROL_PARTIAL_SOLUTIONS(ps, 1000)
9: distribution ← map(prop, control_items)
10: return eCDF(prop(ps), distribution)

```

- A list of **workers**, each having some rota pattern options and a skill-set
- The **rota patterns**, made of “Working” and “Not Working” days.
- A calendar length (e.g. 12 weeks)

More specifically, the rota pattern assigned to each worker is between the choices that are available for them. In optimisation terms each worker is a variable and their rota assignment is their value (e.g. `worker#6 → rota#2`), and a solution corresponds to the assignments of rotas for all workers.

The fitness function is determined by the “range” of the quantity of scheduled employees *with a certain skill on a certain day of the week*. For example, of Fibre Optic engineers on Tuesdays there might be $\{2,1,5,2,5\}$ throughout five weeks, but having $\{3,2,2,3,2\}$ would be better since the quantity only ranges between 2 and 3. Let $Q_x(\text{skill}, \text{weekday})$ be the function returning the quantity of workers trained in `skill` working on `weekday` as determined by solution x , for each week (e.g. $Q_x(\text{Fibre}, \text{Tuesday}) = \{2, 1, 5, 2, 5\}$).

The aim is to **minimise the range of workers for each skill-weekday combination** (Eq. (5)), with some additional components that contribute to the objective function in Eq. (8).

$$\text{Range}_x(\text{skill}, \text{weekday}) = \left(1 - \frac{\min(Q_x(\text{skill}, \text{weekday}))}{\max(Q_x(\text{skill}, \text{weekday}))} \right)^2 \tag{5}$$

$$\text{weight}(\text{weekday}) = 10 \text{ if } \text{weekday} \in \{\text{Saturday}, \text{Sunday}\}, \text{ else } 1 \tag{6}$$

$$\text{preference_score}(x) = W \cdot \#\{\text{non-favourite rotas in } x\} \tag{7}$$

$$F(x) = \sum_{\substack{s \in \text{skills} \\ d \in \text{weekdays}}} (\text{Range}_x(s, d) \cdot \text{weight}(d)) + \text{preference_score}(x) \tag{8}$$

Firstly, **weekends** have increased importance and thus their contribution to the fitness is increased by using the **weight** function (Eq.6). Additionally, each employee has a **favourite rota** within their choices, and the fitness improves as more of these are picked (Eq.7). Finally, the weighted ranges for each skill-weekday combination are added with the preference score (with a weight W , here set to 0.001) to form the final fitness function (Eq. (8)).

The instance used here is (almost) identical to [8], with the same workers and rotas available to them. There are 141 workers, each having at most 4 rota options of which there are 95 in total. Unlike [8] where **skills** are not considered, there are 18 possible skills and each worker is trained in at most 5. Lastly, the calendar length is 3 months (13 weeks).

The desired interpretable descriptors for a Partial Solution of this problem have been informed by our industrial partner and are as follows:

- Mean number of rota choices per worker
- Rota difference (Hamming Distance, in days) between the assigned rotas
- Number of Saturdays and Sundays with at least one worker assigned
- Skill coverage ($|\bigcup_{i=1}^n S_i|$) and overlap $\left(\frac{|\bigcup_{i=1}^n S_i|}{\sum_{i=1}^n |S_i|}\right)$ between the workers
- Proportion of workers assigned to their preferred rota

5 Analysis of Explanations

Here we demonstrate some real examples of the explanations produced by our system in regards to our Staff Rostering problem instance. These consist of local and global explanations implemented via a command line program, and the outputs are reported in the following sections. The parameters used for this section were determined from the results in Sect. 6.

5.1 Local Explanations

The program we implemented presents the top solutions found by SA to the user and ask which one they would like to see explained. Once the solution is selected, the program displays the PSs contained within it, and their descriptors. We selected the best solution present, and of the PSs shown we picked 3 representative examples from the top 5 (Fig. 3). Each explanation consists of a group of employees and their rota assignments (the PS), and its descriptors accompanied by their “percentile” (described in Sect. 3.3). Each PS has an “effect on fitness” representing the mean difference in fitness between solutions which contain the PS and those that do not, with negative values being preferred for this problem.

Example A presents a group of workers whose chosen rotas appear to be convenient for covering Sundays and that are also the worker’s preference. Moreover, their rotas are different from each other by nearly 24 days on average (pairwise). Their interaction is also explained by the similarity of their skills.

Example B is a smaller group of people, and this is reflected in a smaller effect on fitness. As opposed to Example A, these workers have similar rotas (extremely, according to the percentile) and their rotas are better suited to cover Saturdays rather than Sundays.

Finally, **Example C** contains another configuration, but with less descriptors than the previous examples. This group specialises in covering Sundays but not Saturdays, and their skills are similar (i.e. there are overlaps).

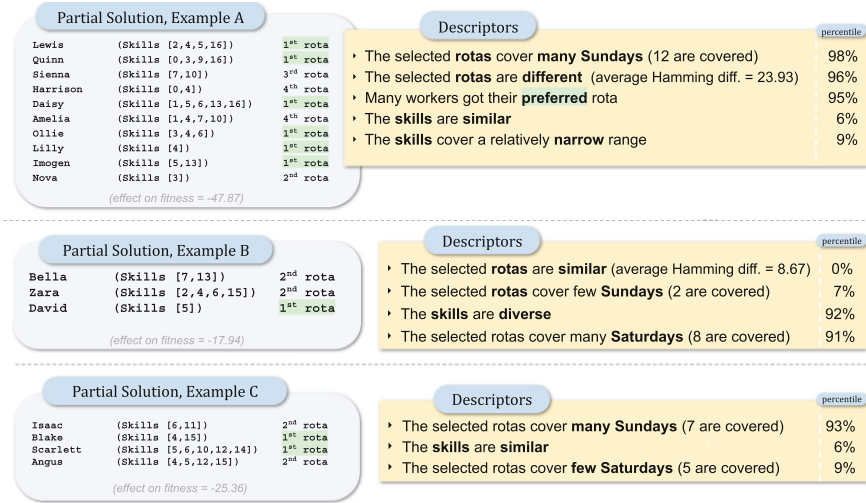


Fig. 3. The 1st, 2nd and 5th explanations produced by the system. The 3rd and 4th have been omitted due to their similarity with the 1st and 2nd.

In case a worker would like to know why they were assigned to a certain rota, some explanations can provide a lot of insight. If, for example, Sienna wanted to know why she was assigned to her 3rd rota, she would be presented with Example A and any other explanations pertaining her. From Example A she would infer that her rota choice needs to work well with the workers in that group because they have similar skills, and that her rota should be very different from them. It should be noted that the ability to explain an assignment depends on relevant Partial Solutions being present, and in the solution used here 80% of the variables were explained. It is worth noting that of the 28 unexplained variables, 24 are workers with only one rota choice (*i.e.* no choice).

Looking at all the explanations produced, it is clear that many are very similar to each other, despite the use of the objective-space crowding operator. Of the first 10 returned, 5 were redundant, which suggests that a post-processing filtering step might be beneficial.

5.2 Global Explanations

Additional insights can be gained by analysing all of the PSs obtained by the PS mining algorithm. The first result is the frequency with which the workers appear in PSs, likely to be correlated to their importance to the problem. The frequency of certain employees being high would imply that they interact with more groups, and thus that their rota choices are more critical. This was compared against the variable importance obtained through LIME on a random forest regression model trained on the reference population, yielding a statistically significant but weak correlation (p-value = 0.004, explains $R^2 = 5.7\%$ of variance, coeff. = 0.24).

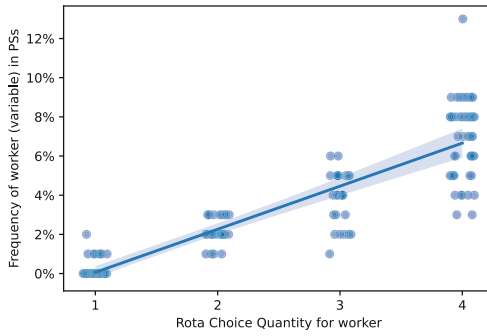


Fig. 4. A scatter-plot to show the correlation between the frequency of workers (each corresponding to a variable) appearing in PSs, and the quantity of rota choices which are available to them. A light horizontal jitter was applied.

Our variable importance was observed to positively correlate with the number of rota choices (p-value $\ll 0.05$, explains $R^2 = 78\%$ of variance, coeff. = 0.88, Fig. 4). **This suggests that employees with more rota choices have a greater impact on the fitness**, possibly due to increased tunability, which was observed by *Fyvie et al.* in [6] as well.

More specific information can also be obtained in correlation to variable importance, and in this particular instance it was observed that workers with skills 5 and 15 tend to have greater importance (p-values are 0.063 and 0.024 respectively). These skills are not significantly over or under-represented within the workforce, but a closer inspection reveals that the weekday ranges for these skills are better than most (they are both in the top 5).

5.3 Utility of Explanations

The information obtained in the local and global explanations can be used by decision makers to decide what changes to implement within their workforce in order to improve the settings of their Staff Rostering task. For example, if a manager wanted to improve the importance of the less critical workers, they could train them in the skills found to be important in the global explanations. Additionally, knowing which skills are important could be used to guide the hiring process and explain to candidates why they were turned down and what they could do to improve their chances in the future.

Once the workers are hired, the managers will be aware that having multiple rotas is also beneficial, and therefore they will make many rota choices available to the new hires. To decide these rotas, they might look at the local explanations of the current solutions and find groups of workers that already appear to be interacting due to those skills. Looking at the rotas of those workers (and at descriptors relating to whether they are similar/different), the manager can propose rotas that are likely to harmonise with the existing workforce.

6 Experiments and Results

The experiments in this section will assess the PS Mining algorithms' ability to find PSs. We designed 2 Staff Rostering problems to have pre-determined interacting groups of workers, and the algorithm's performance is measured by counting how many are found through a run with a limited evaluation budget. More specifically, it should find the optimal configurations for these groups, of which there might be multiple, and these are referred to as **target PSs**.

The specific construction of these problems is available at [2]. The first benchmark problem contains 5 groups of 4 workers each, and in order for a group to perform well each worker should be assigned to their second rota of the two they have available. This setup resembles Royal Road in its behaviour, and therefore it will be referred as **SR-RR** [15]. This tests the algorithm's performance on an instance where there is only one optimal configuration for each group.

The second instance contains more complicated relationships, with 5 groups of 3 workers each. For a group to do well, the rotas assigned to the workers need to all be different, and each worker has 3 options available. This is designed to resemble a Graph Colouring problem with 5 independent triangular cliques, and it will be called **SR-GC**. This instance was designed to test the algorithm's performance when each clique has many optimal configurations, in this case 6 per group (and 30 target PSs in total). This problem also tests the potentially negative influence that SA might have on the results: the SA runs are unlikely to find all of the optimal configurations, causing their absence in the reference population and preventing their discovery.

The experiments aim to determine the best PS Miner parameters, specifically:

- Reference Population origin $\in \{100\% \text{ SA}, 50\% \text{ SA} + 50\% \text{ Uniform}\}$
- Multi-Objective algorithm $\in \{\text{NSGA-II}, \text{NSGA-III}, \text{MOEAD}, \text{SMS-EMOA}\}$
- Crowding-operator $\in \{\text{Objective space (default)}, \text{Decision Space}\}$
- Population size $\in \{50, 100, 200\}$
- Evaluation Budget per run $\in \{1000, 2000, 5000, 10000\}$

Each run consists of a sequence of smaller executions of the Multi-Objective search algorithm which are limited by a total PS evaluation budget of 50000, and the reference population is always limited to be 10000 individuals in total. At the end of the run, we count the proportion of found target groups of workers. Each combination of problems and parameters is executed 100 times, with Table 1 presenting the average proportions across those runs.

The results of the experiments suggest the following to find the target groups:

- Using **NSGA-II** with **decision space crowding**
- Using many **smaller populations with smaller budgets** (50 individuals, budget of 1000 PS evaluations). Larger population sizes (even beyond 200) yield similar rewards per run but prevent sufficient repeated runs by wasting the evaluation budget.
- Using a reference population which is composed of individuals from a **SA and uniform random sampling**

Table 1. Average proportion of discovered groups for the SR-RR and SR-GC problems, applied on differing reference populations (Best performing shown in **BOLD**)

	Multi Objective GA	Run settings		Ref. Pop = 100% SA		Ref. Pop = 50% uniform, 50% SA		
		Pop. Size	Eval. Budget	SR-RR	SR-GC	SR-RR	SR-GC	
Objective space Crowding	MOEAD	50	1000	11%	14%	40%	25%	
		100	5000	25%	6%	25%	11%	
		200	10000	28%	4%	25%	7%	
	NSGA-II	50	1000	59%	35%	82%	64%	
		100	5000	16%	10%	50%	17%	
		200	10000	10%	7%	44%	10%	
	NSGA-III	50	1000	81%	37%	74%	61%	
		100	5000	43%	10%	46%	13%	
		200	10000	34%	6%	30%	7%	
	SMS-EMOA	50	1000	59%	38%	86%	66%	
		100	5000	15%	11%	54%	19%	
		200	10000	10%	8%	46%	11%	
	Decision space crowding	MOEAD	50	1000	8%	14%	42%	23%
			100	5000	27%	6%	26%	10%
			200	10000	30%	4%	21%	7%
NSGA-II		50	1000	67%	43%	93%	68%	
		100	5000	33%	18%	86%	27%	
		200	10000	22%	13%	77%	18%	
NSGA-III		50	1000	58%	47%	25%	32%	
		100	5000	33%	27%	41%	26%	
		200	10000	30%	16%	29%	14%	
SMS-EMOA		50	1000	56%	39%	87%	67%	
		100	5000	20%	12%	61%	20%	
		200	10000	12%	8%	48%	12%	

The results on the experiments indicate that the algorithm handles SR-RR much better than SR-GC, supporting the suspicion that the algorithm struggles to find all patterns when the optimisation problems has many global optima, which is likely to be related to the convergence properties of SA and could be reduced by using diversity enhancing mechanisms.

The local explanations can be used to explain the variable assignments of a solution produced by SA, but are not guaranteed to be able to explain all of the variables. In the use case presented in Sect. 5, 20% of the variables were not included in any explanation, making them uninterpretable, but many of these had a cardinality of 1 and thus did not need explanation.

The speed of PS Miner is highly dependent on the number of variables. For these small problems of 20 variables at most it takes around 1 min, and around 10 min for the Staff Rostering problem (on an i7-3770 CPU @ 3.40 GHz with 20 GB RAM running Windows 10). This is caused by the complex objectives in use. Further work to reduce execution time is underway.

7 Conclusion

The research questions set out in Sect. 1 are answered as follows:

- **Mining** → PSs are found using the algorithms discussed in Sect. 3.2, applied on the reference population described in Sect. 3.1.
- **Explainability** → Partial Solutions can be used to obtain both **local** and **global** explanations, improved by finding interpretable problem-specific descriptors which have sufficiently extreme values
- **Insights** → In the Staff Rostering problem, the PSs can be used to explain the solutions produced by SA. **Local explanations** show which groups of workers are interacting, and likely reasons, but they can also be used for **global explanations**. The variable importances we obtained correlate (weakly but significantly) to those produced by LIME, and they are positively correlated with the number of rota options available to the workers, which was also found by [6].

While our methodology produces novel explanations, the results also indicate that the algorithms could still be improved in their ability to find PSs on problems with more complex search spaces, such as continuous variables and permutations. In general, the structure of PSs presented here is limited to fairly low-level patterns and does not translate to higher order relationships (such as $var_1 = var_4$), which would be able to explain a much wider class of problems. Additionally, the PSs produced by the system are sometimes redundant (*i.e.* the same pattern is present multiple times with slight variations) and should be culled in a systematic manner. This can be resolved using a simple heuristic which removes PSs for which a more general PSs has already been found. Moreover, in the context of explainability it is often argued that the user should have some control over the explanations, which is currently lacking in this project but could be implemented by allowing selection of the weightings for simplicity, mean fitness and atomicity. Future work will explore alternative methods of finding PSs in order to improve PS mining speeds and tunability of explanations.

Source code for this project is available at [2].

Acknowledgments. This paper was written as part of a funded PhD project supported by The Data Lab and BT Group plc.

Disclosure of Interests. Author G. Catalano is being sponsored by British Telecom.

References

1. Barredo Arrieta, A., et al.: Explainable artificial intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI. *Inf. Fusion* **58**, 82–115 (2020)
2. Catalano, G.: PS Assisted Explainability (2024). <https://github.com/Giancarlo-Catalano/PS-descriptors>
3. Catalano, G., Brownlee, A.E.I., Cairns, D., McCall, J., Ainslie, R.: Mining potentially explanatory patterns via partial solutions (2024). <https://arxiv.org/abs/2404.04388>
4. Deb, K., Srinivasan, A.: Innovization: discovery of innovative design principles through multiobjective evolutionary optimization. In: *Multiobjective Problem Solving from Nature*, pp. 243–262 (2008)
5. Dimitropoulaki, M., Kern, M., Owusu, G., McCormick, A.: Workforce rostering via metaheuristics. In: Bramer, M., Petridis, M. (eds.) *SGAI 2018. LNCS (LNAI)*, vol. 11311, pp. 277–290. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-04191-5_25
6. Fyvie, M.: Explainability of non-deterministic solvers: explanatory feature generation from the data mining of the search trajectories of population-based metaheuristics. Ph.D. thesis, Robert Gordon University (2024)
7. Fyvie, M., Mccall, J., Christie, L., Brownlee, A.: Explaining a staff rostering genetic algorithm using sensitivity analysis and trajectory analysis. In: *Proceedings of GECCO 2013*, pp. 1648–1656 (2023)
8. Fyvie, M., McCall, J.A.W., Christie, L.A., Brownlee, A.E.I., Singh, M.: Towards explainable metaheuristics: feature extraction from trajectory mining. *Expert Syst.* e13494 (2021). <https://doi.org/10.1111/exsy.13494>
9. Ghaheri, A., Shoar, S., Naderan, M., Hoseini, S.S.: The applications of genetic algorithms in medicine. *Oman Med. J.* **30**, 406–416 (2015). <https://doi.org/10.5001/omj.2015.82>
10. Glibovets, N., Gulayeva, N.M.: A review of niching genetic algorithms for multimodal function optimization. *Cybern. Syst. Anal.* **49**, 815–820 (2013). <https://doi.org/10.1007/s10559-013-9570-8>
11. Goodman, B., Flaxman, S.: European union regulations on algorithmic decision making and a “right to explanation”. *AI Mag.* **38**(3), 50–57 (2017)
12. Hsu, S.H., Yu, T.L.: Optimization by pairwise linkage detection, incremental linkage set, and restricted/back mixing: DSMGA-II. In: *Proceedings of GECCO 2015*, pp. 519–526. ACM, New York (2015)
13. Li, X., Epitropakis, M.G., Deb, K., Engelbrecht, A.: Seeking multiple solutions: an updated survey on niching methods and their applications. *IEEE Trans. Evol. Comput.* **21**(4), 518–538 (2017). <https://doi.org/10.1109/TEVC.2016.2638437>
14. Minh, D., Wang, H.X., Li, Y.F., Nguyen, T.N.: Explainable artificial intelligence: a comprehensive review. *Artif. Intell. Rev.* 1–66 (2022)
15. Mitchell, M., Forrest, S., Holland, J.: The royal road for genetic algorithms: fitness landscapes. In: *European Conference on Artificial Life*, vol. 1, pp. 23–33 (1992)
16. Ochoa, G., Christie, L.A., Brownlee, A.E., Hoyle, A.: Multi-objective evolutionary design of antibiotic treatments. *Artif. Intell. Med.* **102**, 101759 (2020). <https://doi.org/10.1016/j.artmed.2019.101759>
17. Phillips, P.J., et al.: Four principles of explainable artificial intelligence (2021). <https://doi.org/10.6028/NIST.IR.8312>

18. Ribeiro, M.T., Singh, S., Guestrin, C.: “why should i trust you?”: explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2016, pp. 1135–1144. ACM, New York (2016). <https://doi.org/10.1145/2939672.2939778>
19. Roeva, O.: Real-world applications of genetic algorithms. Books on Demand (2012)



Formalise Regulations for Autonomous Vehicles with Right-Open Temporal Deontic Defeasible Logic

Pak Yin Chan¹(✉) , Xue Li¹ , Yiwei Lu², Yuhui Lin^{1,3} , and Alan Bundy¹

¹ School of Informatics, University of Edinburgh, Edinburgh, UK
histo.chanpy@outlook.com, {xue.shirley.li, a.bundy}@ed.ac.uk

² School of Law, University of Edinburgh, Edinburgh, UK
y.lu-104@sms.ed.ac.uk

³ Computer Sciences, Heriot-Watt University, Edinburgh, UK

Abstract. To negotiate the safety and liability concerns of Autonomous Vehicles (AVs), manufacturers desire AVs to conform to traffic laws and regulations with auto-reasoning capabilities and the ability to detect passengers' misbehaviour. Our study focuses on formalising regulations for AVs based on Temporal Deontic Defeasible Logic (TDDL). We adopt a directed obligation to introduce the person in charge of events that can differ from the event executor. We also relax the time intervals from bounded to right-open to enable AVs to conduct auto-reasoning even without knowing when events end. Based on the modified deontic and temporal features, a Right-Open TDDL (RTDDL) is developed. The UK Highway Code is selected to illustrate the proposed RTDDL to ensure the behaviours of AVs obey traffic laws in complex traffic scenarios.

Keywords: Temporal Deontic Defeasible Logic · Formalisation · Directed Obligation · Autonomous Vehicles

1 Introduction

The deployment of *autonomous vehicles* (AVs)¹ is considered as a possible solution to reducing vehicle accidents. Since they are not affected by certain physical conditions that humans are prone to [2, 20], and they exercise greater execution abilities than humans [2, 30], it is possible that the widespread adoption of AVs could eliminate traffic congestion [25] and improve traffic efficiency [26]. To achieve these visions, the design of AVs has to address multiple concerns. While it is important to enhance AVs' perception abilities, sensing and interpreting objects from sensors to human-level proficiency [2], we focus on how AVs' behaviour can comply with existing traffic regulations.

¹ We refer AVs as the vehicles with the highest driving automation level (level 5) according to the Society for Automotive Engineers (SAE) [28], which can operate without any human intervention.

Traffic laws encompass not only the operational details of vehicles but also the duties and responsibilities of drivers, passengers, and other stakeholders. AVs take on the human drivers’ roles, making appropriate decisions when operating the vehicles [17]. The definition of an AV “driver” becomes questionable, whether it refers to any human sitting behind the steering wheel, the AV itself, or others [21, 31]. Yet, with the disappearance of human drivers in AVs, it is essential to identify the person in charge when the driving duties are conducted by the AV. For instance, drivers, the responsible human users of AVs, need to guarantee that passengers adhere to traffic regulations like wearing seat belts, and it is unclear who bears the responsibility when AVs replace drivers’ roles. Despite not being liable for drivers’ duties, AVs may also be responsible for ensuring passengers follow traffic laws for safety concerns. Passengers may be more encouraged to risk-take to misbehave as they feel safe to take AVs [13], such as not wearing seat belts. It becomes another challenge for AVs to detect passengers’ misbehaviour and respond accordingly. In addition, AVs are required to consider both temporal information and law priority when deciding whether actions are appropriate. In particular, some decisions involve the guarantee that future events will happen, and AVs have to consider them in their navigations. For example, the maximum 50 m travel distance in **Rule** 99 of the UK Highway Code [5]² is a precondition of permitting not wearing seat belts, which must be guaranteed before driving starts but this is not relevant for AVs.

Our work emphasises designing a logic for the decision-making of AVs, which not only involves how to operate the vehicle but also considers passengers’ duties and responsibilities. AVs are required to behave correctly while avoiding undesirable actions to ensure the safety of both passengers and other road users. To address these concerns, we adopt Temporal Deontic Defeasible Logic (TDDL) [22], an extended Defeasible Logic [19] with features of Deontic Logic [27] and temporal information, as the foundational logic for representing traffic laws. We modify the logic to allow AVs to conduct decision-making at any time, anticipating future events for predictive safety. We refer to our modified version as Right-Open TDDL (RTDDL). We evaluate our formalisation by selecting several rules in the current UK Highway Code [5] in our study. In contrast to other traffic laws, such as the Road Traffic Act 1988, it is less rigorous and structured but includes more discussions on when road users are guilty of offences or have duties. Our hypothesis is as follows.

By extending TDDL with right-open time intervals and introducing the person in charge, our RTDDL can formalise traffic regulations such as the UK Highway Code.

In the next section, we briefly introduce the background of our work, including previous attempts to formalise traffic laws for AVs and a brief introduction to TDDL. Section 3 introduces the grammar and semantics of RTDDL, together with the discussion of proof in the theory. We then formalise several rules in the

² All rules in the Highway Code are annotated as “**Rule** X”, where X is a reference number. We only present excerpts of **Rules** for concise discussion.

Highway Code in Sect. 4. Finally, we conclude with a discussion of future work of coding traffic laws with RTDDL.

2 Background

2.1 Related Work

The term *formalisation* varies in different publications. [24] considered formalisation as a two-phase task, first representing a natural language text in a formal language, which is named “codification”, then giving computable definitions of vague concepts for AVs’ judgment on traffic conditions, which refers as “concretisation”. In this work, we only refer to the former task. Works on traffic laws formalisation for AVs focused on controlling vehicle trajectories, such as rules for intersections [6, 11, 14], road junction [1], overtaking and safe distance [3, 4, 7, 24]. To the best of our knowledge, no existing work on traffic law formalization for AVs addresses the behavioural rules for AV passengers.

There were attempts at formalising traffic rules into Answer Set Programming [11] or Defeasible Deontic Logic [3]. Yet, they missed the integration of time relations in the formalisation [14]. Linear Temporal Logic (LTL) is capable of modelling sequences of events and became the choice of logic for [1, 4, 7]. Later works [14, 15] adopted Metric Temporal Logic (MTL) as it also models durations. Nevertheless, formalising traffic regulations in LTL or MTL have two main drawbacks. First, it is necessary to identify all exceptional cases for a rule, and either include their negations in the rule’s premise [7] or include them in the rule’s conclusion [15]. Although this reduces the readability of the formalised rules, it also assumes AVs are factual omniscient; otherwise, they cannot conclude from partial but sufficient knowledge [9]. Secondly, these works aimed to regulate AV trajectories, as mentioned in [23]. They did not apply Deontic Logic as AVs perform what the engineers design. Yet, traffic regulations also involve human behaviour, and there is a difference between the ideal one (outlined in the regulation) and the actual one (that may violate the laws). If these are not aligned, AVs may need to remind passengers to follow regulations. As a result, we are motivated to select a logic which contains all of these features, and we select TDDL [22] as a basis of our logic to formalise traffic laws.

2.2 Temporal Deontic Defeasible Logic (TDDL)

As the backbone of TDDL [22], Defeasible Logic (DL) [19] is a non-monotonic logic that is closer to human reasoning, where we retract previous conclusions when a contrary evidence arises. Evidence defeating one another forms a hierarchy, represented by superiority relations in DL. Definition 1 illustrates the components of a DL theory.

Definition 1 (DL Theory). *A DL theory \mathbb{T} is a tuple $(\mathcal{R}, \mathcal{F}, \mathcal{S})$, that consists of a set of rules \mathcal{R} with unique labels \mathcal{L} , a set of facts \mathcal{F} and a set of superiority relations \mathcal{S} on \mathcal{R} .*

There are three types of rules in DL. Strict rules (\rightarrow) are indisputable and cannot be challenged by any exceptions, while defeasible rules (\Rightarrow) are debatable, and their conclusions can be contested. Undercutting defeaters (\rightsquigarrow) do not support an inference but are used as counter-evidence to decline conclusions drawn from defeasible rules. Each rule is assigned a unique label, which is used to define the rules' priorities in the set S . S aims to resolve conflicting conclusions. If two rules with different priorities derive contradictory conclusions, the conclusion derived from the higher priority rule overrides the one from the lower priority rule, which is then discarded. The rules and the priority are defined in Definition 2 and 3 respectively.

Definition 2 (DL Rules). *A DL rule is in definite clauses, i.e. single-headed, and is in the form of*

$$r : \phi_1 \wedge \dots \wedge \phi_n \leftrightarrow \psi \quad (1)$$

for $n \geq 0$, where ϕ_i and ψ are literals, and $r \in \mathcal{L}$ is a rule label. A rule is

- a strict rule if the arrow is \rightarrow ,
- a defeasible rule if the arrow is \Rightarrow , and
- an undercutting defeater if the arrow is \rightsquigarrow .

Definition 3 (Rule Priority). *The hierarchy between two rules is defined by the predicate \succ /2, which rule with label r_1 is superior to the rule with label r_2 is denoted as $r_1 \succ r_2$.*

The types of accepted literals in Definition 2 differ between various kinds of DL. In DL, literals are simply propositions (p) and their negations ($\neg p$). TDDL introduces temporal information and deontic modalities, so literals that include these elements are referred to as *temporal modal literals*. They are composited by a literal with one of the following temporalised modal operators: $@^T$, \mathcal{O}^T or \mathcal{P}^T . Each operator is assigned a bounded time interval T to indicate the period that the literal holds. The first modal operator $@$ serves as a placeholder, while \mathcal{O} and \mathcal{P} are from Deontic Logic [18] indicating obligations and permissions respectively. They emphasise the discrepancy between people's actual performances and the norms specified in laws, such as duties and responsibilities. While all temporalised modal operators can associate with literals, $@^T$ can also associate with rules to indicate the period of existence. Due to the introduction of the time dimension, a TDDL theory also includes a set for time \mathcal{T} , a discrete totally ordered set of instants of time.

3 Formalisation in Right-Open TDDL (RTDDL)

3.1 Overview

The Right-Open TDDL, our modified TDDL, is designed for modelling events-based systems, like AVs, to support automated reasoning. We assume that

machine observations can be represented as some attributes. In RTDDL, these attributes are represented as propositions. For instance, an AV represents the scenario of a child standing on the road by a proposition $locate(child, road)$.

RTDDL theory maintains the same structure as TDDL's, though we modify the time set \mathcal{T} in TDDL. We remove the discrete restriction as \mathcal{T} can apply on continuous data points. In addition, \mathcal{T} in RTDDL may not have supremum. AVs do not know when an observation ends during operation, and it may continue indefinitely. Thus, we remove the upper bound of \mathcal{T} . Yet, the minimal element t_{min} of the set still exists, marking the starting moment for the model record. As a remark, a rule in TDDL is associated with two periods: the period of force and the period of efficacy. The former states the period for the legal norm is in force, while the latter indicates the time that provision causes its legal effect [22]. In this paper, we omit the distinction of the two periods for brevity and to maintain clarity in our examples.

Definition 4 (RTDDL Theory). *An RTDDL theory \mathbb{T} is a tuple $(\mathcal{T}, \mathcal{R}, \mathcal{F}, \mathcal{S})$, where $\mathcal{R}, \mathcal{F}, \mathcal{S}$ is a DL theory as in Definition 1 and \mathcal{T} is a totally ordered set for the time with the minimal element t_{min} .*

RTDDL language is defined by modifying two main TDDL features to fit the concerns of modelling regulations for AVs. These modifications will be discussed later, including (i) changing the time interval from bounded to right-open, and (ii) integrating the person in charge in the deontic operators. We present the Backus-Naur Form (BNF) of the RTDDL language below. All RTDDL expressions are in Skolem Normal Form, i.e. all quantifiers are removed.

Definition 5 (RTDDL Language). *For a set of propositions $Prop$, and a subset of constants that include all people Per , and other sets in Definitions 2 and 4, the RTDDL language is defined as follows.*

```

<F> ::= <GenLit>
<R> ::= <L> "." " $\hat{h}^{<T>}$ " "(" <Preconds> <Arrow> <GenLit> ")"
<S> ::= <L> ">" <L>
<Preconds> ::= <GenLit> | <GenLit> "&" <Preconds>
<GenLit> ::= <Lit> | <DLit> | <TLit>
<TLit> ::= " $\hat{h}^{<T>}$ " "(" <Lit> ")" | " $\hat{h}^{<T>}$ " "(" <DLit> ")"
<DLit> ::= <Deon> <Per> "(" <Lit> ")"
<Lit> ::= <Prop> | "¬" <Prop>
<Deon> ::= "O" | "P"
<Arrow> ::= "→" | "⇒" | "↔"

```

We provide a simple RTDDL theory in Example 1 to illustrate how to formalise regulations for AVs. The semantics of RTDDL theories, such as the meaning of \hat{h}^T , are discussed in detail in the coming subsection. \mathcal{R} stores formalised

traffic laws, specifications by manufacturers and commonsense knowledge. \mathcal{F} is composed of observations from AVs, which could be a capture of a person sitting in the AVs, or information given by users, together with other commonsense knowledge.

Example 1. An RTDDL theory with a set of dates \mathcal{T} for over-simplified **Rule 99**. Here $r_{99.a}$ and $r_{99.b}$ depict that passengers (Pax) shall wear a seat belt unless exempted due to medical conditions ($exMed$). \mathcal{F} describes Alice’s situation, and the \mathcal{S} defines applying $r_{99.b}$ instead of $r_{99.a}$ when they are in conflict.³

$$\begin{aligned} \mathcal{R} &= \{r_{99.a} : \hat{h}^T(adult(Pax)) \wedge \hat{h}^T(sit(Pax, Car)) \\ &\quad \Rightarrow \hat{h}^T(\mathcal{O}^{Pax}(wear(Pax, belt))), \\ r_{99.b} &: \hat{h}^T(exMed(Pax)) \Rightarrow \hat{h}^T(\mathcal{P}^{Pax}(\neg wear(Pax, belt))), \\ \mathcal{F} &= \{\hat{h}^{02Jul}(adult(alice)), \hat{h}^{08Jul}(sit(alice, tesla)), \hat{h}^{01Jan}(exMed(alice))\}, \\ \mathcal{S} &= \{r_{99.b} \succ r_{99.a}\} \end{aligned}$$

3.2 Semantics

RTDDL modifies the semantics of both deontic and temporal operators. For deontic ones, we adopt the concept of directed obligation⁴ [10] on deontic operators. Directed obligation introduces a stakeholder, or the *bearer*, who has responsibility for certain obligations. For AV regulations, the assumption that “an event executor is the person in charge” does not necessarily hold, as there are no human drivers in AVs as event executors. For instance, an AV could be the executor of taking the emergency kit mentioned in **Rule 228**. Yet, another party such as the manufacturer, not the AV itself, is liable for such responsibility. We utilise the notation of directed obligations in RTDDL to waive the above assumption. Definition 6 depicts the meanings of deontic literals.

Definition 6 (Deontic literals in RTDDL). *For a literal Lit , which is either a proposition or its negation, and a person in charge Per ,*

- $\mathcal{O}^{Per}(Lit)$ represents “it is obligatory, toward Per , that Lit ”.
- $\mathcal{P}^{Per}(Lit)$ represents “it is permissible, toward Per , that Lit ”.

For the temporal dimension, we employ *after*(t) from Event Calculus [12] as a unary predicate for the time period. An observation starts after the moment t ⁵. Its end is undefined but may be suggested in other predicates. We extend TDDL with right-open time intervals to incorporate the cases where the termination moments of observations are not known at the moments of initialization.

³ All constants and predicates are in camelCase, except the deontic operators. Variables are in PascalCase.

⁴ Not to be confused with the one proposed by [29], which introduced an individual protecting the interests through exercising the obligations.

⁵ In RTDDL, an observation begins at t , whereas in Event Calculus, it occurs after t .

Definition 7 (Temporal Literals in RTDDL). For a literal Lit (that may be deontic) and a moment $T \in \mathcal{T}$, a temporal literal is in the form of $\hat{h}^T(Lit)$ which means “ Lit happens from the moment T ”. For simplicity, $\hat{h}^{t_{min}}(Lit)$ can be written as Lit .

The shorthand notation refers to the idea that any time-independent knowledge can be considered as a time-dependent observation that starts from the beginning. Notice that in RTDDL, there is no negative temporal literal, e.g. $\neg\hat{h}^t(\mathcal{O}^{per}(l))$, or double-temporal literal, e.g. $\hat{h}^{t_1}(\hat{h}^{t_2}(\mathcal{P}^{per}(l)))$ that TDDL does. The former one is disallowed as its meaning is ambiguous. The negative temporal literal refers to the temporal literal not holding at some unknown moment t' after t . Yet, it is equivalent to explicitly stating Lit holds at t (i.e., $\hat{h}^t(Lit)$) and fails at t' (i.e., $\hat{h}^{t'}(\neg Lit)$). The latter is defined in TDDL language but is not used in the TDDL theory [22].

The languages of rules are defined in Definition 8, which are similar to those in TDDL. Although the body of a rule can be empty in RTDDL, we encourage users not to make it empty in actual practice. Even universal rules apply to stakeholders who are not explicitly mentioned, such as “no parking” for human drivers. Rewriting universal rules into their conditional form (in *conclusions if conditions* structure) [16] helps to minimise any ambiguity. This can be done by specifying variables’ sorts as predicates in the conditions of the rules. Due to space limitation, we omit all such predicates in this paper.

Definition 8 (RTDDL Rules). An RTDDL rule is a temporalised DL rule whose body is non-empty and is in the form of

$$r : \hat{h}^T(\phi_1 \wedge \dots \wedge \phi_n \hookrightarrow \psi)$$

for $n \geq 0$, where ϕ_i and ψ are temporal literals. It reads as “A rule with label r applies from the moment T ”.

Types of rules are defined in the same way as in Definition 2. When $T = t_{min}$, the predicate \hat{h}^T associated with the rule can be omitted for simplicity.

3.3 Proof Theory

The method of proving a temporal literal in RTDDL is derived from TDDL [22], but it is simpler than TDDL’s because of its simpler grammar. The major modification is to change the time intervals from bounded to right-open. While TDDL considers three different relations between two time intervals during the proof, RTDDL focuses only on the relationship between two starting moments. Also, RTDDL ignores the case of two consecutive time intervals within a proof. It is important to note that if a conclusion applies to two disjoint but consecutive time intervals, it suggests that some precondition has changed at some moment. We suspect it is necessary to present such a change during the proof.

Proving a temporal literal involves considering possible attacks by justifying whether one of its complements is also provable. The complementary set of a temporal literal is defined below.

Definition 9 (Complementary Set). *The complementary set of a temporal literal ψ is defined as*

$$\mathcal{C}(\psi) = \begin{cases} \{\neg\psi\}, & \psi \text{ is a proposition} \\ \{\mathcal{O}^{Per}(\neg Lit), \mathcal{P}^{Per}(\neg Lit)\}, & \psi = \mathcal{O}^{Per}(Lit) \\ \{\mathcal{O}^{Per}(Lit)\}, & \psi = \mathcal{P}^{Per}(Lit) \\ \{\hat{h}^T(\bar{Lit}) | \bar{Lit} \in \mathcal{C}(Lit)\}, & \psi = \hat{h}^T(Lit). \end{cases}$$

The proof theory of RTDDL is similar to that of TDDL. Readers are encouraged to refer to the section of *Proof Theory* in [22] for more details. The only difference is the part for definitely provable ($+\Delta(\psi)$). Recall that the termination of observation is represented by a temporal negative literal ($\hat{h}^t(\neg Lit)$) instead of the upper bound of the time interval, which is the complement of the observation. To determine if a temporal literal is definitely provable at the moment t , not only do we need to find a rule or fact to support the observation occurring on or before t , but we also need to ensure no complement stops the observation. The proof of the absence of such a complement is new in RTDDL because we adopt the right-open time interval.

4 Case Studies

In this section, we explore how RTDDL's features could express traffic regulations to monitor AV trajectories and passengers' behaviours. We also illustrate how AVs use RTDDL to deduce the desired responses in real-life situations. Unless specified, we assume the person in charge of any passengers' *Pax* behaviour in AVs is the manufacturer *Manu* in this section.

4.1 On AVs' Trajectories

Entering Road Junction. Rule 170 discusses the conditions when a vehicle enters a road junction, which shows a sequence of events as the preconditions of a rule.

Rule 170. *You should watch out for cyclists, motorcyclists and pedestrians (road users) and give way to pedestrians crossing or waiting to cross a road into which or from which you are turning. Do not cross or join a road until there is a gap large enough for you to do so safely.*

[1] formulates the **Rule** in an extended-LTL language called RoR language, which uses the binary operator \cup representing "until" to connect two events. In RTDDL, we formulate the sequence of events by introducing time variables T_i and using an inequality to compare these variables, ensuring that the AV observes road users (*RU*) for a moment before entering the junction (*Jct*).

Example 2. An RTDDL-formulation for **Rule 170** for AVs.

$$r_{170} : \hat{h}^{T_1}(\text{watch}(\text{car}, \text{Jct}, \text{RU})) \wedge \hat{h}^{T_2}(\neg \text{cross}(\text{RU}, \text{Jct})) \\ \wedge \hat{h}^{T_2}(\text{exists}(\text{SafeGap}, \text{Jct})) \wedge T_1 < T_2 \\ \rightarrow \hat{h}^{T_2}(\mathcal{P}^{\text{Manu}}(\text{enter}(\text{car}, \text{Jct}))) \quad (2)$$

$$r_{170_a} : \hat{h}^T(\mathcal{P}^{\text{Manu}}(\text{enter}(\text{car}, \text{Jct}))) \Rightarrow \hat{h}^T(\text{enter}(\text{car}, \text{Jct})) \quad (3)$$

The right-open time interval in RTDDL allows us to express that the watching event continues after it starts at T_1 . The same variable T_2 for no road users crossing and the existence of a safe gap refers to them holding concurrently at T_2 , but not necessarily starting at T_2 . As we assume the manufacturer is the one that bears the responsibility when the AV enters the junction, we assign this information in the \mathcal{P} operator. To connect the permission of an action with actual performance, we also assign another defeasible rule for applying the permission. It is not a strict rule, as such actions may be interrupted by other factors, such as an accident at the junction.

Overtaking. While **Rules 162** to **164** depict the preconditions and details for overtaking. **Rules 165** to **167** are the cases that prohibit overtaking. Listing all of them in the preconditions of rules that permit overtaking is computationally inefficient [8]. We take **Rule 165** as an example to show how RTDDL tackles exemptions such as in DL.

Rule 165. *You must not overtake after a “No Overtaking” sign and until you pass a sign cancelling the restriction.*

The duration of the prohibition is determined by passing the signs. The RTDDL formulation shown in [Example 3](#) tells how they correlate, where *NoOT* represents the sign of “No Overtaking”. Two defeasible rules (4) and (5) tell the prohibition starts and ends when an AV passes the “No Overtaking” sign and the cancelling sign respectively. Notice that the conclusions of the two rules are complementary by [Definition 9](#), as the start of the permission indicates the end of the prohibition.

Example 3. An RTDDL-formulation for **Rule 165** for AVs.

$$r_{165_1} : \hat{h}^T(\text{pass}(\text{car}, \text{NoOT})) \Rightarrow \hat{h}^T(\mathcal{O}^{\text{Manu}}(\neg \text{overtake}(\text{car}))) \quad (4)$$

$$r_{165_2} : \hat{h}^T(\text{pass}(\text{car}, \text{CancelSign})) \Rightarrow \hat{h}^T(\mathcal{P}^{\text{Manu}}(\text{overtake}(\text{car}))) \quad (5)$$

$$r_{165_1} \succ r_{165_2} \quad (6)$$

When an AV passes the “No Overtaking” sign and cancelling sign sequentially, there are two results from (4) and (5), and the latter starts at a later moment. As the two results are complementary, RTDDL selects the conclusion that is established later. Therefore, the AV cannot overtake until it passes the cancelling sign when the conclusion from (5) is established. Similarly, when the AV passes

the cancelling sign and then the “No Overtaking” sign, RTDDL selects the latter conclusion. Thus, it cannot overtake after passing the second sign. If the AV passes both signs simultaneously, which could happen if there is a road work near the cancelling sign and the “No Overtaking” sign is placed, the two rules conflict with each other, resulting in no clear conclusion about whether overtaking is prohibited or allowed. In this situation, we introduce the priority of these rules with (6) to resolve this conflict, which here we decide to overtake is prohibited in this case.

4.2 On Passengers’ Behaviour

Wearing Seat Belt. **Rule 99** requires everyone in a vehicle to use appropriate seat belts. It introduces the separation of the event executor and person in charge when this phenomenon does not uniquely arise from the appearances of AVs. If a child passenger does not follow the **Rule**, the driver of a non-autonomous vehicle, instead of the passenger, is penalised. We describe the scenario as two defeasible rules in Example 4. For the case of AVs, the absence of a human driver suggests that we should rename the person in charge in (7) from driver to manufacturer.

Table 1. Excerpt of the Table of Seat Belt Requirements in **Rule 99**, where the height limitation is omitted.

	Rear seat	Who is responsible?
Child aged 12 or 13 years	Seat belt MUST be worn if available	Driver
Adult passengers aged 14 and over	Seat belt MUST be worn if available	Passenger

In Example 4 we also include the following exemption from **Rule 99**. Thus, we also define the hierarchy as (10) and (11) to avoid inconsistency between the exemption and obligations. Notice that the proposition $travel(car, Route, Dist)$ is associated with the same time variable with the permission, which indicates the travel distance is only a prediction instead of an actual measurement.

Rule 99. *Exemptions (of wearing a seat belt) are allowed for those making deliveries in goods vehicles when travelling less than 50 metres.*

Different from controlling AV trajectories, AVs cannot control human behaviours directly. To ensure passengers’ safety, AVs must also ensure that passengers adhere to any applicable norms stated in traffic regulations. It is not a logical fault when an AV detects passengers do not follow the traffic laws, and what manufacturers can do is design a warning system. We demonstrate it in Example 4. When an AV detects a passenger not wearing a seat belt but it is obligatory, the AV warns the passenger after a fixed period (as a constant t_{act}) as in (12).

Example 4. An RTDDL-formulation for **Rule 99** for AVs.

$$r_{99.1} : \hat{h}^T(\text{age}(Pax, A)) \wedge 12 \leq A < 14 \wedge \hat{h}^T(\text{sit}(Pax, car)) \\ \Rightarrow \hat{h}^T(\mathcal{O}^{\text{Manu}}(\text{wear}(Pax, belt))) \quad (7)$$

$$r_{99.2} : \hat{h}^T(\text{age}(Pax, A)) \wedge A \geq 14 \wedge \hat{h}^T(\text{sit}(Pax, car)) \\ \Rightarrow \hat{h}^T(\mathcal{O}^{\text{Pax}}(\text{wear}(Pax, belt))) \quad (8)$$

$$r_{99.3} : \hat{h}^T(\text{delivery}(Pax, car)) \wedge \hat{h}^T(\text{travel}(car, Route, Dist)) \\ \wedge Dist \leq 50 \Rightarrow \hat{h}^T(\mathcal{P}^{\text{Pax}}(\neg \text{wear}(Pax, belt))) \quad (9)$$

$$r_{99.3} \succ r_{99.1} \quad (10)$$

$$r_{99.3} \succ r_{99.2} \quad (11)$$

$$r_{99a} : \hat{h}^T(\mathcal{O}^{\text{Per}}(\text{wear}(Pax, belt))) \wedge \hat{h}^{T+t_{act}}(\neg \text{wear}(Pax, belt)) \\ \rightarrow \hat{h}^{T+t_{act}}(\text{warn}(car, Pax, wearBelt)) \quad (12)$$

$$r_{99b} : \hat{h}^{T+t_{act}}(\mathcal{O}^{\text{Per}}(\text{wear}(Pax, belt))) \wedge \hat{h}^T(\neg \text{wear}(Pax, belt)) \\ \rightarrow \hat{h}^T(\text{remind}(car, Pax, wearBelt)) \quad (13)$$

Similarly, an AV can also be responsible for reminding passengers to obey norms in the future. A possible scenario would be that the AV discovers a road diversion and changes the planned route during the travel. The predicted travel distance is updated by adding both the new one and the negation of the old one as facts, associating with that moment. As (9) cannot be used to override (8), if the AV deduces that the passenger will need to wear the belt soon, it can issue an early reminder using (13) to ensure the passenger has enough time (t_{act}) to respond.

Using Mobile Phones. As another example of adopting predictions in AV planning, **Rule 149** discusses the permission to use mobile phones.

Rule 149. *You must not use a hand-held mobile phone when driving. There is an exception if you are using a hand-held mobile phone to make a contactless payment at a contactless payment terminal and the goods must be received at the same time as, or after, the contactless payment.*

Assume the forbidden use of mobile phones still holds for the one sitting in the driver's seat in AVs. The **Rule** mentioned that the guarantee of goods received is also a precondition for allowing the use of the mobile phone, as the **Rule** does not intend to include the case of using the phone at the terminal without ordering anything. This is an example of a displaced effect, where norms are established either before or after their preconditions [16]. RTDDL allows AVs to associate the guarantee in the future moment as a fact and uses this fact and apply (15) in Example 5 to allow the passenger to use the mobile phone⁶.

⁶ We change the predicate from *drive* to *sit* in (14) as the passenger may not drive the AV.

Example 5. An RTDDL-formulation for **Rule 149** for AVs.

$$r_{149.1} : \hat{h}^T(\text{sit}(\text{Pax}, \text{car})) \Rightarrow \hat{h}^T(\mathcal{O}^{\text{Manu}}(\neg \text{usePhone}(\text{Pax}))) \quad (14)$$

$$\begin{aligned} r_{149.2} : T_1 \leq T_2 \wedge \hat{h}^{T_1}(\text{locate}(\text{Pax}, \text{terminal})) \wedge \hat{h}^{T_2}(\text{receive}(\text{Pax}, \text{goods})) \\ \Rightarrow \hat{h}^{T_1}(\mathcal{P}^{\text{Manu}}(\text{usePhone}(\text{Pax}))) \end{aligned} \quad (15)$$

$$r_{149.2} \succ r_{149.1} \quad (16)$$

5 Conclusion

This paper presents RTDDL, a modification of TDDL, for formalising traffic laws. We address challenges specific to formalising laws for AV driving systems and tackle them with our formalisation. Considering the shift in responsibility with AVs, we extend the deontic operator to express the person in charge who may differ from the person involved. For the planning and reasoning functions of AV driving systems, we modify the representation of time intervals to be right-open. We use RTDDL to express various rules from the UK Highway Code, demonstrating how our logic can reason about AV trajectories and passenger behaviour with traffic laws.

Since we provide only several simplified examples from the UK Highway Code, we do not justify that RTDDL is a perfect solution for traffic law formalisation. There are additional features in traffic laws that we do not cover here. For instance, **Rule 277** allows drivers to choose different options based on the traffic situation. This suggests that norms may require non-Horn clauses, which can be a future work to extend RTDDL for this. Nonetheless, we show that RTDDL is sufficient for addressing major concerns of traffic laws on time and applicability.

Acknowledgments. This study was supported by UKRI grant EP/V026607/1. The authors would also acknowledge Prof. Alex Lascarides, Prof. Andrew Ireland, Prof. Burkhard Schafer, Prof. Gonzalo A. Aranda-Corral and anonymous reviewers for valuable discussions and feedback.

Disclosure of Interests. All authors have no competing interests to declare that are relevant to the content of this article.

References

1. Alves, G.V., Dennis, L., Fisher, M.: Formalisation and implementation of road junction rules on an autonomous vehicle modelled as an agent. In: Sekerinski, E., et al. (eds.) FM 2019. LNCS, vol. 12232, pp. 217–232. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-54994-7_16
2. Anderson, J.M., Kalra, N., Stanley, K.D., Sorensen, P., Samaras, C., Oluwatola, O.A.: Autonomous Vehicle Technology: A Guide for Policymakers. RAND Corporation (2014). <http://www.jstor.org/stable/10.7249/j.ctt5hhwgz>
3. Bhuiyan, H., Governatori, G., Bond, A., Rakotonirainy, A.: Traffic rules compliance checking of automated vehicle maneuvers. *Artif. Intell. Law* **32**(1), 1–56 (2024). <https://doi.org/10.1007/s10506-022-09340-9>

4. Costescu, D.M.: Keeping the autonomous vehicles accountable: legal and logic analysis on traffic code. In: Varhelyi, A., Žuraulis, V., Prentkovskis, O. (eds.) VISZERO 2018. LNITI, pp. 21–33. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-22375-5_3
5. Department for Transport: The Highway Code (2023). <https://www.gov.uk/guidance/the-highway-code>
6. Esterle, K., Aravantinos, V., Knoll, A.: From specifications to behavior: maneuver verification in a semantic state space. In: 2019 IEEE Intelligent Vehicles Symposium (IV), pp. 2140–2147 (2019). <https://doi.org/10.1109/IVS.2019.8814241>
7. Esterle, K., Gressenbuch, L., Knoll, A.: Formalizing traffic rules for machine interpretability. In: 2020 IEEE 3rd Connected and Automated Vehicles Symposium (CAVS), pp. 1–7 (2020). <https://doi.org/10.1109/CAVS51000.2020.9334599>
8. Governatori, G.: Practical normative reasoning with defeasible deontic logic. In: d’Amato, C., Theobald, M. (eds.) Reasoning Web 2018. LNCS, vol. 11078, pp. 1–25. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00338-8_1
9. Governatori, G., Padmanabhan, V., Rotolo, A., Sattar, A.: A defeasible logic for modelling policy-based intentions and motivational attitudes. *Logic J. IGPL* **17**(3), 227–265 (2009). <https://doi.org/10.1093/jigpal/jzp006>
10. Herrestad, H., Krogh, C.: Obligations directed from bearers to counterparties. In: Proceedings of the 5th International Conference on Artificial Intelligence and Law, ICAIL 1995, pp. 210–218. Association for Computing Machinery, New York (1995). <https://doi.org/10.1145/222092.222243>
11. Karimi, A., Duggirala, P.S.: Formalizing traffic rules for uncontrolled intersections. In: 2020 ACM/IEEE 11th International Conference on Cyber-Physical Systems (ICCPs), pp. 41–50 (2020). <https://doi.org/10.1109/ICCPs48487.2020.00012>
12. Kowalski, R., Sergot, M.: A logic-based calculus of events. *N. Gener. Comput.* **4**(1), 67–95 (1986). <https://doi.org/10.1007/BF03037383>
13. Litman, T.: Autonomous vehicle implementation predictions: implications for transport planning (2020). <https://trid.trb.org/View/1678741>
14. Maierhofer, S., Moosbrugger, P., Althoff, M.: Formalization of intersection traffic rules in temporal logic. In: 2022 IEEE Intelligent Vehicles Symposium (IV), pp. 1135–1144 (2022). <https://doi.org/10.1109/IV51971.2022.9827153>
15. Manas, K., Paschke, A.: Legal compliance checking of autonomous driving with formalized traffic rule exceptions. In: Proceedings of the Workshops co-located with 39th International Conference on Logic Programming (ICLP) (2023). <https://doi.org/10.24406/publica-1709>
16. Marín, R.H., Sartor, G.: Time and norms: a formalisation in the event-calculus. In: Proceedings of the 7th International Conference on Artificial Intelligence and Law, ICAIL 1999, pp. 90–99. Association for Computing Machinery, New York (1999). <https://doi.org/10.1145/323706.323719>
17. McLachlan, S., Neil, M., Dube, K., Bogani, R., Fenton, N., Schaffer, B.: Smart automotive technology adherence to the law: (de)constructing road rules for autonomous system development, verification and safety. *Int. J. Law Inf. Technol.* **29**(4), 255–295 (2022). <https://doi.org/10.1093/ijlit/eaac002>
18. McNamara, P., Van De Putte, F.: Deontic logic. In: Zalta, E.N., Nodelman, U. (eds.) *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, fall 2022 edn. (2022). <https://plato.stanford.edu/archives/fall2022/entries/logic-deontic/>
19. Nute, D.: Defeasible logic. In: Bartenstein, O., Geske, U., Hannebauer, M., Yoshie, O. (eds.) INAP 2001. LNCS (LNAI), vol. 2543, pp. 151–169. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-36524-9_13

20. Olson, P.L., Farber, E.: Forensic aspects of driver perception and response, 2nd edn. Lawyers and Judges Publishing Company, Tucson, AZ (2003)
21. Prakken, H.: On the problem of making autonomous vehicles conform to traffic law. *Artif. Intell. Law* **25**(3), 341–363 (2017). <https://doi.org/10.1007/s10506-017-9210-0>
22. Riveret, R., Rotolo, A.: Temporal deontic defeasible logic: an analytical approach. In: Casanovas, P., Sartor, G., Casellas, N., Rubino, R. (eds.) *Computable Models of the Law. LNCS (LNAI)*, vol. 4884, pp. 239–253. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-85569-9_15
23. Rizaldi, A., Althoff, M.: Formalising traffic rules for accountability of autonomous vehicles. In: 2015 IEEE 18th International Conference on Intelligent Transportation Systems, pp. 1658–1665 (2015). <https://doi.org/10.1109/ITSC.2015.269>
24. Rizaldi, A., et al.: Formalising and monitoring traffic rules for autonomous vehicles in Isabelle/HOL. In: Polikarpova, N., Schneider, S. (eds.) *IFM 2017. LNCS*, vol. 10510, pp. 50–66. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-66845-1_4
25. Rojas-Rueda, D., Nieuwenhuijsen, M.J., Khreis, H., Frumkin, H.: Autonomous vehicles and public health. *Annu. Rev. Public Health* **41**, 329–345 (2020). <https://doi.org/10.1146/annurev-publhealth-040119-094035>
26. Roozmond, D.A.: Using intelligent agents for pro-active, real-time urban intersection control. *Eur. J. Oper. Res.* **131**(2), 293–301 (2001). [https://doi.org/10.1016/S0377-2217\(00\)00129-6](https://doi.org/10.1016/S0377-2217(00)00129-6)
27. Royakkers, L.L.: *Extending Deontic Logic for the Formalisation of Legal Rules*, 1st edn. Springer, Dordrecht (2011). <https://doi.org/10.1007/978-94-015-9099-0>
28. SAE International: *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles. Standard J3016. Technical report*, SAE International, Warrendale, PA (2021). https://www.sae.org/standards/content/j3016_202104/
29. Sartor, G.: Fundamental legal concepts: a formal and teleological characterisation. *Artif. Intell. Law* **14**(1), 101–142 (2006). <https://doi.org/10.1007/s10506-006-9009-x>
30. Taeihagh, A., Lim, H.S.M.: Governing autonomous vehicles: emerging responses for safety, liability, privacy, cybersecurity, and industry risks. *Transp. Rev.* **39**(1), 103–128 (2019). <https://doi.org/10.1080/01441647.2018.1494640>
31. Vellinga, N.E.: From the testing to the deployment of self-driving cars: legal challenges to policymakers on the road ahead. *Comput. Law Secur. Rev.* **33**(6), 847–863 (2017). <https://doi.org/10.1016/j.clsr.2017.05.006>



SLANGO - The Initial Blueprint of Privacy-Oriented Legal Query Assistance: Exploring the Potential of Retrieval-Augmented Generation for German Law Using SPR

Jérôme Agater^(✉)  and Ammar Memari 

Jade University of Applied Sciences, Friedrich-Paffrath-Straße 101,
26389 Wilhelmshaven, Germany

{jerome.agater, ammar.memari}@jade-hs.de

<https://www.jade-hs.de/en/the-university/departments/engineering/>

Abstract. This paper introduces an application of Large Language Models (LLMs) in the context of Retrieval-Augmented Generation (RAG) to the problem of privacy-preserving question answering in a legal setting. As Germany's voluminous legal documents, including laws and court decisions, are not stored accurately in the inherent knowledge of LLMs, LLMs are prone to producing unreliable or non-existent references. By augmenting the inherent knowledge with ground truth facts retrieved from a Neo4J database, the answer-generating system can cite the facts directly. By using a locally run LLM, we mitigate the need for cloud-based data processing, preventing privacy-relevant data from leaving the system. Our preliminary results with selected legal questions show the system's ability to provide plausible legal answers. This research lays the foundation for further studies, opening the possibility for integrating more sophisticated RAG techniques and building a user interface with deterministic quoting for precise citation and ease of use. Our study presents a step towards deploying AI in sensitive legal settings, promising a future where legal questions can be answered correctly by LLMs without sacrificing data privacy.

Keywords: Local Large Language Models · Retrieval-Augmented Generation · Legal Information Retrieval · Data Privacy · Neo4J Database · Open-Source Llama Model · Legal Documents · Machine-Readable Law Data Sources · Secure Legal Queries · Artificial Intelligence in Legal Context · Privacy-Preserving AI · Germany Federal Law · Legal Technology · Non-Parametric Memory · Legal Knowledge Base

1 Introduction

Legal experts, such as lawyers, require quick access to written laws and legal documents. In Germany, there are several thousand federal and state laws. Fur-

thermore, court decisions, which serve as case law, contribute to the pool of legal precedents. Locating the relevant paragraphs and their content within this data can be challenging. For non-experts, obtaining answers to legal questions becomes even more difficult when dealing with written laws and court decisions.

Large language models (LLMs) excel at many tasks of natural language processing [3]. They contain a significant amount of world knowledge extracted from their training data during their training [18]. However, LLMs need to be retrained with new material when incorporating updated knowledge. The current generation of LLMs does not store the texts of all paragraphs of the laws completely and truthfully in their weights. Furthermore, LLMs are prone to hallucinations, especially when tasked to cite references supporting their answers. Zuccon et al. investigated whether the LLM ChatGPT can provide evidence in the form of references for its answers in different topics and whether these references would actually support the claims ChatGPT makes in them. They found that, in 86% of the time, the reference did not exist. For the remaining 14%, the references did not substantiate the claims ChatGPT had generated to a large extent, even when an expert had labeled the answer as *correct* or *partially correct* [24]. Akyurek et al. investigated a path to allow LLMs to trace their answers back to their training data and concluded that much more work is needed before reliable fact tracing becomes possible [1]. This severely limits the applicability of LLMs for knowledge retrieval in legal settings. However, different solutions exist to augment the world knowledge contained in an LLM with additional facts, serving as ground truth. One paradigm is retrieval-augmented generation or RAG for short. In RAG, the inherent and static knowledge and reasoning capabilities of an LLM are augmented with facts from another ground truth data source, which can also be updated easily without expensive training of an entire LLM. Shuster et al. investigated the RAG architecture and found that implementing RAG can reduce hallucinations [19]. A RAG solution seems to be a fitting choice for the problem regarding the retrieval of texts: For the legal context, we can store the extensive law corpus in a database and then query this ground truth database for excerpts of the law relevant to the task, i.e., finding all paragraphs relevant to a specific legal question. Once the excerpts have been collected, an LLM uses these excerpts and the information about their origin to fulfill the task by answering the legal question based on the actual text of the law. As both the excerpts and their citation are given as input to the LLM, the LLM can potentially reference the paragraphs of the law in its answer.

However, the most prominent LLM, *ChatGPT*, is only available as a cloud service. This is also true for its different embedding models, including `text-embedding-3-small`, `text-embedding-3-large`, and `text-embedding-ada-002`. When questioning ChatGPT, private data contained in the legal questions must therefore leave the local machine to be used in the computations of the cloud infrastructure realizing the ChatGPT service. Due to the very sensitive nature of some legal data, such as privacy-relevant data, business contractual data, or criminally relevant data, i.e., data only available under attorney-client privilege, this is not acceptable. In the absence of an equivalent to

homomorphic encryption¹ for LLM execution, the only way to guarantee the privacy of personal data entered into query systems is the total avoidance of LLM services running on third-party servers, such as ChatGPT. In February 2023, a team from Meta Inc. released the open-weight LLaMA (Llama henceforth) model², which provides performance on par with the performance of ChatGPT [21]. In September 2023, Mistral AI announced³ an open-source LLM with a permissive Apache 2.0 license. By leveraging instances of Llama and Mistral on a local system, the problem of private data leaving the local environment could be alleviated, while retaining similar performance. In this paper, we explore whether we can construct a totally local RAG system, i.e., a system that uses an LLM *locally*, an embedding model *locally*, as well as a *locally run* ground truth knowledge database, preventing the escape of privacy-relevant data. We call the system SLANGO, short for Semantic Legal Analysis and Graph Operations.

Contributions. We present the machine-readable German law data sources, upgraded an existing dataset that was no longer usable to be importable by the current version of Neo4J. Furthermore, we enhanced the dataset with Sparse Priming Representation (SPR) compressed summaries, created the chunks and embeddings, and made them available as part of the new dataset. Based on this improved dataset, we implemented an enhanced version of the Naive RAG architecture as a proof of concept. We present the graph queries used inside that PoC while answering some case questions. We make the database available in the form of a compressed file containing Cypher commands on the Open Science Framework (OSF) website⁴. The code of the proof of concept is available at our GitHub repository⁵.

2 Background of Data

While the upcoming standard *Akoma Ntoso* (also known as *LegalDocML*) for representing legal documents in XML is in the process of being adopted in Germany in the form of the local profile *LegalDocML.de* [9], currently, the legal documents of interest, such as the actual law texts and court decisions (case law), are only available in custom and semantically less structured proprietary XML variants. Nevertheless, efforts are underway to make the legal content available in machine-readable form.

In the following, we describe the state of the base data corpora from the official *Gesetze im Internet* and *Rechtsprechung im Internet* online platforms,

¹ See also [2].

² Llama 3, released in April 2024, see <https://ai.meta.com/blog/meta-llama-3/> (last accessed: 2024-06-27).

³ Via a blog post on their homepage, <https://mistral.ai/news/announcing-mistral-7b/>.

⁴ SLANGO's DB Dump is available as a project at <https://osf.io/yvez7/>.

⁵ SLANGO's GitHub repository at <https://github.com/Ingenieurinformatik/Slango>.

provide a short overview of the efforts so far to make the legal data more machine-accessible, and describe the database containing the dataset we decided on using for our approach.

2.1 Data Corpus from *Gesetze im Internet* Platform

Nearly the entire corpus of the German federal laws has been made available by the German Federal Ministry of Justice and the German Federal Office of Justice in machine-readable formats via the *Gesetze im Internet*⁶ (GII) web platform [5]. While the official version of the laws is still published exclusively in the *Bundesgesetzblatt* as PDF documents⁷ via the federal proclamation platform⁸, the consolidated texts available are continuously curated and updated by the documentation center of the Federal Office of Justice [5]. Besides presentation-oriented formats like HTML, PDF, and EPUB, the law texts are retrievable as XML documents following a distinct Document Type Definition (DTD) for the GII platform⁹, referred to as *gii-norm* and originally created by the juris GmbH on behalf of the German Federal Ministry of Justice. There is no documentation available regarding the semantics of the DTD-defined XML tags or instructions on how to translate them via XSLT to, for example, HTML, see also [13]. However, as the tags relate to the rendered HTML version and the tags themselves are words from German, semantic meaning can be derived by deduction anyway.

The platform provides an index of all available documents. The index is updated daily and implemented as an XML document itself¹⁰. The index document contains the title of each law, as well as an HTTP-URL pointing to a ZIP archive containing at least the compressed XML file encoding the law using the *gii-norm* DTD and possibly complementary data in the form of images in GIF or JPEG format, as well as documents in PDF format. As of June 1st, 2024, the index contains 6,771 items, which, once retrieved, expand to around 600MB of decompressed XML and complementary data.

Semantics of DTD Tags in XML Documents. Each XML document representing a law text is composed of a list of norms. Each norm has associated metadata and possibly textual/structural content. Since laws are not written with machine-readability in mind, some formatting of the originating law texts has been preserved by using HTML-like markup tags, for example, **B** for marking bold text, *I* for italic text, and **SP** for setting a part of the text letter-spaced. The **metadata**, as annotated in the norms, contains the different names/titles of the law, the identification of the paragraphs of the law, as well as outline information. Furthermore, information about the version (or status) can be annotated, and the originating document/publication can be specified. The **law text**

⁶ “*Laws on the internet*”.

⁷ Since January 1st, 2023.

⁸ *Verkündungsplattform des Bundes*, see also [4].

⁹ <https://www.gesetze-im-internet.de/dtd/1.01/gii-norm.dtd>.

¹⁰ <https://www.gesetze-im-internet.de/gii-toc.xml>.

body content consists of sequences of the actual subparagraphs/sentences of the law interspersed with formatting, tables, images, and so on. While some meta-information about the law, as well as the paragraphs and the lists inside the laws, are semantically structured, a lot of the definitions of the DTD deal with the presentation instead of the semantics of the content. For example, legal citations/references have no markup associated with them and therefore need to be extracted from the textual content stream via another mechanism.

2.2 Data Corpus from *Rechtsprechung im Internet Platform*

About 72,000 selected decisions of the Federal Constitutional Court, the supreme courts of the Federal Republic of Germany, and the Federal Patent Court are available through the Federal Ministry of Justice and the Federal Office of Justice in machine-readable formats via the *Rechtsprechung im Internet*¹¹ (RII) web platform [6]. Besides presentation-oriented formats like HTML and PDF, the court decisions are also retrievable as XML documents following a distinct Document Type Definition (DTD) for the RII platform, referred to as *rii-dok*, similar to the GII data corpus described in the previous section.

Semantics of DTD Tags. The *rii-dok* DTD imports the entities defined in the DTD for XHTML-Transitional. It then defines a sequence of elements as part of a **dokument** (document), the XML document’s root element. Many of the tags defined as part of **dokument** can contain just plain text streams; however, some tags can contain an **any** element, meaning they can contain all the entities from an HTML document. As the DTD imports all of the HTML entities, the content of the tags can, in general, be HTML fragments with all the tags from the HTML standard. As Harshil et al. mention, parsing the HTML is problematic and error-prone [7].

2.3 Using the Data Corpus

Different efforts have been undertaken by researchers to make the German laws and case laws available in machine-readable form, building on each other, for example, [7, 17]. With *Open Legal Data*, Ostendorff et al. have created a dataset and corresponding platform¹² [17] by downloading and processing the different government data sources as well as scraping websites containing further legal documents. Assorted data dumps in CSV and JSON format are available for references, laws, and cases.¹³

Milz et al. build on this effort in [16], by creating a Neo4J graph database representation of the *Open Legal Data* dataset to perform an analysis on the citation

¹¹ “Court decisions on the internet”.

¹² The source code is available on GitHub <https://github.com/openlegaldata/oldp> (last accessed: 2024-06-25).

¹³ While the intention had been to regularly provide updates to these documents, the latest partial dump is currently from 2022-10-18 (last checked: 2024-06-26).

network of the German case law and German law texts. Due to the mentioned absence of semantical or structural markup from the DTD for the citations, the law references in the base material are just part of the character sequence stream of the plain text components of the XML documents. Furthermore, as Milz et al. describe, court decisions in the material have no well-structured and unique identifier, complicating the identification of case-to-case citations. Consequently, Milz et al. only take citations into account that are led by the article sign character, using regular expressions in a rule-based approach for the extraction of said citations [16]. Milz et al. made the graph database content available for download as a database dump online¹⁴. In 2022, Harshil et al. created and published a cleaned-up version of the case law aspect of Open Legal Data on Zenodo [7]¹⁵. Apart from these efforts, Fobbe has created multiple Zenodo repositories containing datasets from the GII and RII platforms, for example, the corpus of the German Federal Law [12], decisions of the Federal Administrative Court [11], and the official collection of decisions of the Federal Constitutional Court [10]. By leveraging data extraction pipelines, the repositories have been updated about every three months¹⁶.

The laws contain references to other paragraphs and other laws, forming a citation network. For a given paragraph of a law, the content of referenced paragraphs is also relevant. To extract the references from the textual content of the paragraphs, Ostendorff et al. created the library REfEx for the extraction of legal references using regular expressions¹⁷. In [8], Darji et al. describe the use of a fine-tuned German BERT transformer model for the extraction of legal references from datasets such as Open Legal Data (see also [17]). They created a new dataset of references, which has been refined by the addition of annotations by legal human experts.

We had intended to use the original XML documents, as Ostendorff et al. suggested, for more complex use cases. We wrote an XML ingestor to break up the laws into addressable law atoms, constituting the textual content of the law, where each atom could be referenced individually, for example, a specific sub-sub-bulletin of a list in a sentence of a paragraph of a norm. However, we deemed the resulting law atoms' addresses to be too overwhelmingly specific for the users, due to the complex and convoluted nesting on one hand and the relatively strong complexity of the derived addresses for these law atoms on the other hand.¹⁸ Furthermore, the individualized law atoms provide not nearly enough context for the generator part of our RAG system, making the coalescing

¹⁴ <https://osf.io/8d2v4/> (last checked: 2024-06-25).

¹⁵ Harshil et al. also mention problems with parsing the HTML of the case law content, further underlining the need for semantically cleanly structured legal documents.

¹⁶ While a part of the structural information represented through the tags defined in the `gii-norm` DTD is lost in the process, not much semantic is lost due to the mostly presentational nature of the tags used for *GII* platform text markup.

¹⁷ <https://github.com/openlegaldata/legal-reference-extraction>.

¹⁸ e.g., a law atom can theoretically be part of a revision that itself can be part of a revision and so on, resulting in an uncommon, deeply nested reference not usable in practice.

of more atoms necessary until we had finally reconstructed an entire paragraph. We therefore decided to use entire paragraphs of the laws, making our atomic approach unattractive. With this, we used the dataset from Milz et al. because the references had already been extracted and proven to form a citation network, thus making the ingestion and reference extraction work on our part unnecessary. We describe the usage as well as the advantages of Neo4J as a database for a RAG system in detail in Sect. 4.1 on page 8.

3 Background of Retrieval-Augmented Generation

In 2020, Lewis et al. introduced the paradigm of Retrieval-Augmented Generation (RAG) [15] for knowledge-intensive tasks, which is a hybrid generation architecture: For the RAG paradigm, a parametric language model is augmented with non-parametric memory, thus forming a hybrid system. The RAG paradigm defines a **retriever** component and a **generator** component. The retriever is responsible for querying the non-parametric memory for facts, documents, or other information excerpts. The generator uses the retrieved knowledge facets as further input. The inherent knowledge and reasoning capability of the generator is thereby augmented with the relevant factual knowledge from the non-parametric memory, for example, by adding the context to the prompt for the generator [15]. Even if the retrieved knowledge facets do not contain a definitively relevant fact, the clues in the facets can still contribute to the generation of desirable output from the generator component [15].

Gao et al. perform a survey of current and past developments in the RAG paradigm for LLMs in the form of a continuously updated review paper¹⁹. The review paper [14] examines the state-of-the-art, the foundations, and benchmarks, and describes the different branches of development. Gao et al. differentiate the approaches in literature and practice into three categories with increasing complexity, namely *Naive RAG*, *Advanced RAG*, and *Modular RAG*. As this paper serves as a proof of concept in the context of German law, we initially focused on Naive RAG, the original architecture introduced by Lewis et al. in [15]. According to Gao et al., for Naive RAG, an index of the data corpus is created to be used for finding relevant documents later. To form this index, textual data is extracted from the original documents. From this plain text, embedding vectors are computed using an embedding model. As embedding models generally have a limited content size, the textual data is usually partitioned into chunks of equal size, namely the size of the embedding vectors. Once these embeddings are computed, they are stored in a vector database. Typically, vector databases for AI use cases allow different similarity strategies to be employed for searching the index. This searching of the index occurs through the RAG system by transforming the user query into a vector representation, crucially using the same embedding model used for the creation of the index. This allows comparing the embedding of the user query against the stored embedding vectors of

¹⁹ Currently available in version 5 at the arXiv repository, last revised on March 27th, 2024. See [14] for the URL pointing to the current version.

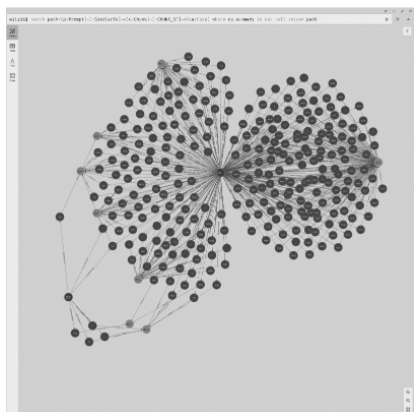


Fig. 1. Matching the user’s prompt to the chunks of court cases using similarity search in Neo4J with the *Cypher* query.

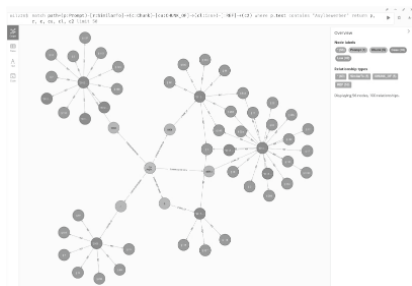


Fig. 2. Searching for chunks matching the user prompt with further hints.

the different chunks. The chunks with the greatest similarity are then chosen and retrieved [14]. The retrieved chunks, or their originating document in its entirety, form the augmentation input for the generator component. In recent years, cosine similarity has been one of the most widely used ways to quantify semantic similarity. In practice, cosine similarity is a widely common metric used for RAG systems, although Steck et al. investigated whether cosine similarity of embeddings really represents semantical similarity [20] and concluded that this is not always the case.

4 Construction

We developed a prototype as a proof of concept of a *Naive RAG* system, then integrated more techniques from the *Advanced RAG* class described by Gao et al. In this section, we outline our design.

A single workstation was sufficient to compute all embeddings, perform LLM-based text generation, operate the graph database, as well as run the actual system software implemented in Python. The workstation we used for our proof-of-concept has the following specifications: Case *Dell Precision 3660 Tower*; Processor *Intel Core i9-13900K, 13th Generation*; RAM *64 GB DDR5 up to 4400 MHz, Non-ECC*; Storage *512GB M.2 PCIe NVMe SSD*; GPU *Nvidia GeForce RTX 4090*; water-cooling.

4.1 Vector Database and Content

We selected the open-source database Neo4J²⁰ as the vector database for this project. While other open-source options for vector databases in the AI space are

²⁰ Homepage <https://neo4j.com/> (last accessed: 2024-07-04),
GitHub <https://github.com/neo4j/neo4j> (last accessed: 2024-05-04).

available, such as Qdrant²¹ or Chroma²², Neo4J offers several advantages for our endeavor. In addition to the vector storage features, Neo4J is a graph database with an expressive query language for graphs called *Cypher*. Furthermore, we already had robust experience with Neo4J in industry and research as a graph database, and it also has data science plugins to support AI applications, such as cosine embeddings. Additionally, Milz et al. [16] had used Neo4J successfully for their dataset for the graph representation of German laws and court decisions.

We chose to use the Milz et al. dataset, as it already included the texts of laws and court data. The dataset is an export of a Neo4J database in an older version. We upgraded the dataset, making it usable in the current version of Neo4J. To be able to utilize the vector storage functionality of Neo4J as well as vector search using cosine similarity, we added the plugins *graph-data-science*, *bloom*, *apoc*, and *apoc-extended*. Finally, we exported our dataset from Neo4J as a database dump and compressed it with `lrzip`, thereby reducing the dataset size from 12.6 GB to 2.5 GB. The enhanced dataset is available as a download from Open Science Framework (OSF)²³.

4.2 Graph Structure and Enrichment

We enhanced the graph by adding newly computed elements necessary for the RAG implementation as well as new interconnections. For instance, the chunks (about 71,900) of the paragraphs' texts are connected to the paragraphs. This allows for interesting queries to be formulated, as shown in Fig. 2²⁴. We added the attribute `embeddings_e5` to these chunks, which contains the computed embedding for each individual chunk. We used the workstation's GPU with *CUDA* to compute these embeddings, resulting in a speedup of a factor of 25 compared to using the CPU alone. Furthermore, we introduced the attribute `SPR` for many law nodes (50,244), containing the Sparse Priming Representation (SPR) of the corresponding law node's paragraphs. Additionally, we annotated many court cases with the attribute `summary`, containing a summary of the content generated with SPR. This was necessary because we found the content of the laws to be too large to fit in the context of any of the LLM options considered for the generation part. We were able to improve our first draft based on the Naive RAG approach to follow a more sophisticated hybrid approach using the different summaries.

²¹ Homepage <https://qdrant.tech/> (last accessed: 2024-07-04),
GitHub <https://github.com/qdrant/qdrant> (last accessed: 2024-07-05).

²² Homepage <https://www.trychroma.com/> (last accessed: 2024-07-04),
GitHub <https://github.com/chroma-core/chroma> (last accessed: 2024-07-05).

²³ <https://osf.io/yvez7/>.

²⁴ The Cypher query used to create the graph representation in the figure is the following: `match path=(p:Prompt) -[r:SimilarTo]-> (c:Chunk) -[cu:CHUNK_OF]-> (cl:Case) -[:REF]-> (c2) where p.text contains "Asylbewerber" return p, r, c, cu, cl, c2 limit 50.`

4.3 LLM and Embeddings

We experimented with different Mistral models trained with varying numbers of parameters, namely, *Mistral 7B*²⁵ as well as *LLama 3*²⁶ for generating the summaries and SPRs. We found that Mistral 7B leads to better results. However, we encountered court case texts that exceed 40k tokens, exceeding the maximum 32k token limit for Mistral 7B. In general, it was beneficial to include the mentioned paragraphs of the law in the LLM context.

For the generation of the e5 embeddings, we employed a multilingual instruction model,²⁷. The multilingual e5 embedding has been presented by Wang et al. in a technical report in 2023 and outperforms the previous state-of-the-art embedding models on benchmarks [22].

4.4 Enhanced RAG Workflow Using Python

With the locally run LLMs and the enhanced graph structure containing the pre-computed e5 embeddings and summaries, we created the following RAG workflow:

1. The user enters a question²⁸ Q with the text Q_{text} .
2. An LLM is instructed to categorize the question Q into one of the following three classes based on Q_{text} :
 - A The texts of the laws are deemed to be sufficient to answer the question²⁹.
 - B The texts from court decisions are deemed sufficient to answer the question.
 - C Both are deemed necessary to answer the question.
3. The question is added as a question node to the database; the question's embedding $e5(Q_{\text{text}})$ is computed and stored as an attribute of the newly created node for use in the next step.
4. Depending on the classification of Q , cosine similarities are calculated between the question's embedding and the embeddings of the chunks of laws (for class A), of the chunks of court decisions (for class B), or of both (in case of class C); see Fig. 1 for an example path resulting from matching the prompt to chunks of a court case.³⁰

²⁵ TheBloke/Mistral-7B-Instruct-v0.2-GGUF, available on HuggingFace.

²⁶ MazyarPanahi/Llama-3-8B-Instruct-64k-GGUF, available on HuggingFace.

²⁷ intfloat/multilingual-e5-large-instruct, available on HuggingFace.

²⁸ see `cases.py` in the repository for examples.

²⁹ e.g., "What type of residence permit do I need?" (German: "Welche Art Aufenthaltstitel brauche ich?").

³⁰ The Cypher command used to create the graph representation is as follows: `match path=(p:Prompt)-[:SimilarTo]->(c:Chunk)-[:CHUNK_OF]->(ca:Case) where ca.summary is not null return path.`

5. For each law, the computed similarities of the chunks are aggregated to form an overall rating for the relevance of the law to the question.³¹ Only the top n laws with the best overall rating are taken into further account. For this, their SPR is retrieved from their corresponding node’s attribute.³² The same procedure is used for the court decisions.
6. A LLM is instructed to pick the m most relevant sources from the retrieved SPRs of the laws and/or court decisions listed as context in its prompt.
7. From the m sources deemed relevant by the LLM, a new context is assembled. For law sources, the corresponding paragraph will be retrieved and appended as a whole. For court decision sources, the SPR text will be appended without further processing.
8. In the final generation phase, this assembled context is given to an LLM. The LLM is instructed to generate the final answer for Q by using the knowledge contained in the context as references.

We implemented this workflow in Python using the `openai` library, which was configured against `localhost`, i.e., the local workstation used for testing. The workstation hosted an API run by LM Studio³³. The API exposed by LM Studio follows the OpenAI API in its call and signature design, allowing us to use the `openai` library to access the models, even though they were running locally in LM Studio.

When asked questions via the command line, the implemented system generates plausible answers that incorporate the actual law and court decision texts from the database. As a proxy for direct legal expertise evaluating the system, we selected two questions from the website `frag-einen-anwalt.de` (“ask a lawyer”), which offers a service where users can ask questions and receive answers from actual lawyers. We ran the selected questions through our RAG implementation and compared the output with the output of the actual human lawyers. We found that the questions were answered correctly and relevant texts retrieved from the graph database were used for the answers. However, when compared to the answers from a real lawyer, the generated answers still lack in quantity regarding the citations. Nevertheless, the system generated answers incorporating the non-parametric knowledge database completely locally on the workstation.

5 Outlook

While our implementation provides plausible answers and matching legal citations for our test-case questions, the answers given have not been systematically

³¹ For the aggregation, we used the *mean* of all the similarities of a law’s chunks, but the *maximum* similarity from this set or the *sum* of the similarities are also viable choices.

³² If a law does not have an SPR attribute already, the attribute is calculated on the fly and stored in the node for further use.

³³ Homepage: <https://lmstudio.ai/> (last accessed: 2024-07-04), GitHub: <https://github.com/lmstudio-ai> (last accessed: 2024-07-04).

checked by legal experts. We do not expect the system in its current form to function as a “production-ready” expert information system. However, starting from the current state as a proof-of-concept spike, the system could be further developed to incorporate more techniques from the *Advanced* and *Modular RAG* classes, with the answers being thoroughly checked by a legal professional in an upcoming study.

At present, the system does not feature a dedicated user interface. Instead, questions are entered and answered via commands. A dedicated Graphical User Interface (GUI), ideally as a web application, would make the system more accessible to users from the legal domain. In both legal and healthcare settings, factual quoting is crucial. Consequently, when implementing a RAG system, unchanged direct quotes of source material are highly desirable. However, since the output of a RAG system is formulated by an LLM, hallucinations can and will occur even when citing facts and textual excerpts from the augmentation context, altering the quoted text. To mitigate this, Yeaung from the healthcare industry reports in [23] the implementation of the concept of *deterministic quoting* for a RAG system. Here, instead of generating a pure textual answer, the textual answer still contains references to the source material, but the actual contents of the source material are quoted verbatim from the original chunks provided and integrated into the output by a deterministic system *after* the generation phase by the LLM has completed. The direct quotes are communicated to the user through a distinct visual style. A similar approach could be used for a RAG system in a legal context. Additionally, further UI interactions can be incorporated easily, such as linking to the specific paragraphs of the law and displaying them upon user request.

The paper by Gao et al. lists various RAG workflows and approaches. To evaluate the usefulness of these workflows, they could be implemented and tested against a common set of test questions. Using a legal expert, the answers from the different workflows could be assessed and then compared in terms of quality.

6 Conclusion

We found that the state of the data of laws and court decisions lacks semantical structure; however, the Milz et al. dataset proved to be a valuable starting point for our RAG implementation. We were able to integrate a locally run LLM, a locally run embedding model, as well as a locally run ground truth knowledge database. Our implemented system generates plausible answers that incorporate the actual law and court decision texts. Since it does not involve any non-local component, the escape of privacy-relevant data can be prevented using this concept. In a subsequent study, the answering capabilities could be evaluated against legal professionals’ assessments, using improvements in the capabilities as guidance for the implementation of more advanced RAG techniques and tuning of the system. For a production system, the concept of deterministic quoting could be implemented to provide a user-friendly GUI.

References




1. Akyurek, E., et al.: Towards tracing knowledge in language models back to the training data. In: Goldberg, Y., Kozareva, Z., Zhang, Y. (eds.) Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, pp. 2429–2446. Association for Computational Linguistics (2022). <https://doi.org/10.18653/v1/2022.findings-emnlp.180>. <https://aclanthology.org/2022.findings-emnlp.180>
2. Armknecht, F., et al.: A guide to fully homomorphic encryption. Cryptology ePrint Archive, Paper 2015/1192 (2015). <https://eprint.iacr.org/2015/1192>. <https://eprint.iacr.org/2015/1192>
3. Brown, T., et al.: Language models are few-shot learners. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Advances in Neural Information Processing Systems, vol. 33, pp. 1877–1901. Curran Associates, Inc. (2020)
4. Bundesministerium der Justiz: Startseite - Bundesgesetzblatt (n/d). https://www.recht.bund.de/de/home/home_node.html/. Accessed 01 June 2024
5. Bundesrepublik Deutschland, vertreten durch den Bundesminister der Justiz: Gesetze im Internet (n/d). <https://www.gesetze-im-internet.de>. Accessed 01 June 2024
6. Bundesrepublik Deutschland, vertreten durch den Bundesminister der Justiz: Rechtsprechung im Internet (n/d). <https://www.rechtsprechung-im-internet.de/jportal/portal/page/bsjrsprod.psml>. Accessed 25 June 2024
7. Darji, H., Mitrović, J.: Clean openlegaldata - german (2022)
8. Darji, H., Mitrović, J., Granitzer, M.: A dataset of German legal reference annotations. In: Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law, ICAIL 2023, pp. 392–396. Association for Computing Machinery, New York (2023). <https://doi.org/10.1145/3594536.3595173>
9. Deutscher Bundestag: Moderne digitale Rechtsetzung. Antwort der Bundesregierung auf die Kleine Anfrage der Abgeordneten Victor Perli, Niema Movassat, Dr. Gesine Löttsch, weiterer Abgeordneter und der Fraktion DIE LINKE (2020). Drucksache 19/25438, 19. Wahlperiode. <https://dserv.bundestag.de/btd/19/256/1925654.pdf>. Accessed 25 June 2024
10. Fobbe, S.: Corpus der amtlichen Entscheidungssammlung des Bundesverfassungsgerichts (C-BVerfGE) (2024). <https://doi.org/10.5281/zenodo.10783177>
11. Fobbe, S.: Corpus der Entscheidungen des Bundesverwaltungsgerichts (CE-BVerwG) (2024). <https://doi.org/10.5281/zenodo.10809039>
12. Fobbe, S.: Corpus des deutschen bundesrechts (c-dbr) (2024). <https://doi.org/10.5281/zenodo.10908139>
13. FragDenStaat: Quelltext für Darstellung von Gesetze im Internet-XML (2020). <https://fragdenstaat.de/anfrage/quelltext-fur-darstellung-von-gesetze-im-internet-xml/>. Accessed 01 June 2024
14. Gao, Y., et al.: Retrieval-augmented generation for large language models: a survey (2024). <https://arxiv.org/abs/2312.10997>
15. Lewis, P., et al.: Retrieval-augmented generation for knowledge-intensive NLP tasks. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Advances in Neural Information Processing Systems, vol. 33, pp. 9459–9474. Curran Associates, Inc. (2020)
16. Milz, T., Granitzer, M., Mitrović, J.: Analysis of a German legal citation network. In: Proceedings of the 13th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2021) - KDIR, pp. 147–154. INSTICC, SciTePress (2021). <https://doi.org/10.5220/0010650800003064>

17. Ostendorff, M., Blume, T., Ostendorff, S.: Towards an open platform for legal information. In: Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2020, JCDL 2020, pp. 385–388. Association for Computing Machinery, New York (2020). <https://doi.org/10.1145/3383583.3398616>
18. Petroni, F., et al.: Language models as knowledge bases? In: Inui, K., Jiang, J., Ng, V., Wan, X. (eds.) Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, pp. 2463–2473. Association for Computational Linguistics (2019). <https://doi.org/10.18653/v1/D19-1250>. <https://aclanthology.org/D19-1250>
19. Shuster, K., Poff, S., Chen, M., Kiela, D., Weston, J.: Retrieval augmentation reduces hallucination in conversation. In: Moens, M.F., Huang, X., Specia, L., Yih, S.W.t. (eds.) Findings of the Association for Computational Linguistics: EMNLP 2021, Punta Cana, Dominican Republic, pp. 3784–3803. Association for Computational Linguistics (2021). <https://doi.org/10.18653/v1/2021.findings-emnlp.320>. <https://aclanthology.org/2021.findings-emnlp.320>
20. Steck, H., Ekanadham, C., Kallus, N.: Is cosine-similarity of embeddings really about similarity? In: Companion Proceedings of the ACM on Web Conference 2024, WWW 2024, pp. 887–890. Association for Computing Machinery, New York (2024). <https://doi.org/10.1145/3589335.3651526>
21. Touvron, H., et al.: Llama: open and efficient foundation language models (2023). <https://arxiv.org/abs/2302.13971>
22. Wang, L., Yang, N., Huang, X., Yang, L., Majumder, R., Wei, F.: Multilingual E5 text embeddings: a technical report (2024). <https://arxiv.org/abs/2402.05672>
23. Yeung, M., Herring, C., Katzfey, L.: Deterministic quoting: making LLMs safer for healthcare (2024). <https://mattyYeung.github.io/deterministic-quoting>. snapshot at <https://web.archive.org/web/20240622015650/https://mattyYeung.github.io/deterministic-quoting>
24. Zuccon, G., Koopman, B., Shaik, R.: Chatgpt hallucinates when attributing answers. In: Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region, SIGIR-AP 2023, pp. 46–51. Association for Computing Machinery, New York (2023). <https://doi.org/10.1145/3624918.3625329>

Short Application Papers



An Ensemble Modelling of Feature Engineering and Predictions for Enhanced Fake News Detection

Patricia Asowo , Sangeeta Lal^(✉) , and Uchenna Daniel Ani 

School of Computer Science and Mathematics, Keele University, Newcastle, UK
{s.sangeeta,u.d.ani}@keele.ac.uk

Abstract. The threat of fake news jeopardizing the credibility of online news platforms, particularly on social media, underscores the need for innovative solutions. This paper proposes a creative engine for detecting fake news, leveraging advanced machine learning techniques, specifically Bidirectional En-coder Representations by Transformers (BERT). Our approach involves feature selection from news content and social contexts, combining predictions from multiple models, including Random Forest, BERT, GRU, LSTM, and a voting ensemble model. Through extensive evaluation of the WELFake dataset, our method highlights an impressive accuracy of 99%, surpassing baselines and existing systems. Our study highlights the crucial role of hyperparameter tuning, improving the performance of the BERT model to 100%.

Keywords: Fake News · Machine Learning · Grid Search

1 Introduction

In the digital era, the spread of false information significantly undermines public discourse. Fake news, defined as deliberately false information intended to deceive, often employs sophisticated techniques to appear credible, using targeted headlines, photos, or videos. Studies indicate that consuming fake news can lead to political misperceptions [3]. The rampant dissemination of false news on digital platforms, such as social media, poses serious societal concerns, impacting public health, electoral processes, and law and order [7]. To address this challenge, contemporary research has explored various methods for detecting fake news, including language-based, topic-agnostic, machine-learning, knowledge-based, and hybrid approaches. Despite these efforts, existing detection methods have limitations, necessitating the development of more effective techniques with enhanced accuracy and performance metrics. This study employs machine learning methods to detect fake news, evaluating models such as BERT, LSTM, GRU, and Random Forest (RF). We used Bi-GRU implementation of the GRU model. Hence, throughout the text GRU model represents the Bi-GRU version. It examines the impact of hyperparameter tuning on these models' performance and explores the effect of ensembling for improved detection accuracy.

2 Related Work

Verma et al. [9] introduced WELFake, a two-phase model using machine learning to detect fake news. It combines linguistic features and word embedding to analyze news content, achieving a 96.73% accuracy rate. The WELFake dataset includes around 72,000 articles. Karimi et al. [5] proposed the Deep Hierarchical Semantic Fusion (DHSF) method, on five datasets, including BuzzFeedNews and PolitiFact. DHSF outperformed other methods with an accuracy of 82.19%. Mishra et al. [6] compared multiple machine learning models, finding that deep learning models, especially Bi-LSTM, performed best with a 95% accuracy rate. Kaliyar et al. [4] developed FakeBERT, integrating BERT with CNN, achieving a 98.90% accuracy rate. They suggest further research to improve classifier performance through advanced pre-processing techniques. Our research builds on these studies, exploring ways to enhance the prediction performance of models for fake news detection.

3 Experimental Dataset

We used the WELFake dataset available on Zenodo [9]. This dataset includes four attributes: the serial number, article headline, article content, and label. It combines information from sources such as Kaggle, McIntire, Reuters, and BuzzFeed Political to provide a rich set of news content. The initial dataset contained 72,134 articles. To manage the processing limitations and GPU resource usage, we split the dataset into two halves using the train-test split function from the sklearn module, ensuring a balanced distribution of real and fake news labels in both halves. This resulted in two DataFrames, `dataset_half1` and `dataset_half2`, each containing half of the total number of articles. We pre-processed the text data in the title and content columns using TF-IDF (Term Frequency-Inverse Document Frequency) vectorization.

4 Model Details

4.1 BERT Algorithm

In this study, the BERT neural network was employed to train the model. We use TensorFlow (TFBertModel) for this. The architecture created through the `'get_model ()'` function served as the backbone for classifying input text into one of two classes. The input layer was established using `'input_ids'` and `'input_mask'`, representing the tokenized input text and its corresponding mask for padding. The two input layers (`input_ids`, `input_mask`), each of shape (None, 100). The BERT model outputs a hidden state of shape (None, 768) that is passed through two dense layers with dropout in between. A dropout layer with a rate of 0.2 was applied to the embeddings. The final layer was a dense layer with a single unit and a sigmoid activation function, yielding a binary classification output. The final dense layer reduces the output to a single unit, indicating a regression or binary classification task. This layer was marked as trainable to fine-tune its weights during training.

4.2 LSTM and GRU Models

This section focuses on Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) neural networks, advanced forms of recurrent neural network (RNN) architectures widely used in natural language processing. LSTM addresses the vanishing gradient problem in traditional RNNs, capturing long-term dependencies with its input, forget, and output gates. GRU simplifies the architecture by omitting the output gate, offering a faster alternative [2]. For this work, LSTM and GRU architectures were developed and evaluated using three models:

- Model 1: Combines Bidirectional GRU and LSTM layers to assimilate contextual information from both directions.
- Model 2: Utilizes Bidirectional GRU layers only, simplifying the architecture while retaining the ability to capture context from both directions.
- Model 3: Merges Bidirectional LSTM with LSTM and GRU layers for nuanced discrimination in textual data.

The bidirectional layers enhanced the models' ability to capture content from both directions.

4.3 Random Forest Classifier

A study done by Song et al. [8] has demonstrated the robust performance of the Random Forest (RF) algorithm in predictive tasks. In this study, we use an RF classifier with the Term Frequency-Inverse Document Frequency (TF-IDF) vectorizer. Our RF classifier employs 100 (i.e., `n_estimators = 100`) decision trees.

5 Results and Discussion

5.1 RQ1: What is the Performance of BERT, LSTM, GRU, and RF for Fake News Detection?

We employed BERT, LSTM, GRU, and RF models for fake news detection. Table 1 presents the results. **The BERT model achieved the highest accuracy at 99%** for both fake and real news articles. Precision, recall, and F-measure metrics were all 99%, highlighting the model's exceptional accuracy and effectiveness. Model 1 (a combination of GRU and LSTM) showed a precision of 95% and recall of 92% for real news, resulting in an F-measure of 93%. Although Model 2 performed well, the absence of detailed precision and recall metrics makes direct comparison challenging. However, its overall performance indicates effectiveness in distinguishing real from fake news. The RF model demonstrated a precision of 96% and a recall of 92% for fake news detection. For real news, the precision was 93% and recall was 96%, leading to an overall F-measure of 95%.

Table 1. Classification results for BERT, LSTM, GRU and RF, Acc: Accuracy, Pre: Precision, F-meas: F-measure

Method	Model	Acc.	Class	Pre	Recall	F-meas.
BERT	BERT-based-uncased	99%	fake	99%	99%	99%
			real	99%	99%	99%
GRU + LSTM	Model 1: Bi-GRU followed by LSTM	93.1%	Fake	92%	94%	93%
			Real	95%	92%	93%
GRU	Model 2: Bi- GRU only	93.7 %	-	-	-	
GRU + LSTM	Model 3: Bi-LSTM followed by LSTM and GRU	92.8 %	-	-	-	
RF		94%	fake	96%	92%	94%
			real	93%	96%	95%

Table 2. Classification results for BERT and Ensemble-GRL Model, Acc: Accuracy, Pre: Precision, F-meas: F-measure

Model	Method	Accuracy	Class	Precision	Recall	F-Measure
BERT	Keras Tuner	100%	fake	99%	100%	100%
		100%	real	100%	99%	100%
Ensemble-GRL Model	Keras Tuner	95%	Fake	92%	98%	95%
		95%	Real	98%	92%	95%

5.2 RQ2: How Does Hyperparameter Tuning Affect the Performance of BERT and an Ensemble of (LSTM, GRU, and RF) Models?

To answer RQ2, we optimized two models: a BERT model and an ensemble-GRL (ensemble of GRU, LSTM, and RF) model using the Keras Tuner, focusing on hyperparameter tuning to enhance individual and ensemble performance. Table 2 shows the results of the BERT model's hyperparameter tuning and performance when the models were combined in ensemble learning. The hyperparameter-tuned BERT model demonstrated high test accuracy, highlighting the efficacy of hyperparameter tuning. The precision for fake news detection is 99%, with recall and F-measure at 100%. Table 2 shows that the ensemble-GRL model surpassed individual models, reaching higher accuracy and F-measure. The ensemble-GRL models' superior performance suggests its ability to leverage strengths and compensate for weaknesses in individual models through ensemble learning. **These results show that the hyperparameter tuning significantly influenced model performance**, optimizing accuracy, and balancing precision and recall in the ensemble, displaying its effectiveness in improving the fake news detection task.

Table 3. The performance of individual and ensemble model (that combines BERT, RF, GRU, and LSTM) models

Method	Model	Class	Accuracy	Precision	Recall	F-Measure
Voting Ensemble	RF	Real	94%	96%	92%	94%
		Fake	94%	93%	96%	95%
	BERT	Real	99%	98%	100%	99%
		Fake	99%	100%	98%	99%
	GRU	real	97%	95%	98%	97%
		fake	97%	98%	95%	97%
	LSTM	real	93%	91%	94%	93%
		fake	93%	94%	91%	93%
	Ensemble-All	real	98%	96%	99%	98%
		fake	98%	99%	96%	98%

5.3 RQ3: What is the Effect of Ensemble Learning on BERT, GRU, LSTM, and RF Models?

Ensemble learning, which combines multiple models, can help enhance the predictive performance of ML models [1]. In this RQ, we developed an ensemble-all model that combines RF, BERT, GRU, and LSTM models. The goal was to leverage their complementary strengths to improve fake news detection. Table 3, shows the results obtained. The BERT model achieved the highest accuracy of 99%. The ensemble model, which aggregates predictions from RF, BERT, GRU, and LSTM, achieved an accuracy of 98% for both real and fake news. **This ensemble model outperformed the individual LSTM, RF, and GRU models.**

Table 4. Confusion matrix for the individual and ensemble model performance

Model	True Negative	False Positive	False Negative	True Positive
RF	3235	268	131	3525
BERT	3495	8	79	3577
GRU	3441	62	186	3470
LSTM	3300	203	327	3329
Ensemble-all	3477	26	144	3512

Confusion Comparison for the Ensemble Model: The confusion matrix in Table 4 outlines the performance metrics of each model, BERT, GRU, LSTM, and the Ensemble model in distinguishing between real and fake news articles. It details the counts of True Negatives, False Positives, False Negatives, and True

Positives for each model. BERT demonstrated the highest accuracy, achieving the lowest counts of both false positives and false ranked second, showing lower false positives and negatives compared to individual models like GRU, RF, and LSTM.

6 Conclusion and Future Work

In conclusion, our research assessed the efficacy of various models in detecting fake news using the WELFake dataset. BERT emerged as the top-performing model with an accuracy of 99%. Hyperparameter tuning further enhanced BERT's performance, achieving 100% accuracy, and emphasizing the importance of meticulous parameter tuning. The ensemble-all model which is an ensemble of (BERT, RF, GRU and LSTM) achieved an accuracy of 98% and hence, surpassed the individual model (except BERT), highlighting the potential of ensemble learning to improve the fake news detection accuracy. The comparative analysis ranked the models as follows: BERT, ensemble-all, GRU, Random Forest, and LSTM. These results under-score BERT's effectiveness in discerning fake news and the complementary benefits of ensemble learning. Future research should focus on further experiments and refinements to enhance the ensemble model's performance.

Acknowledgments. We like to thank Chandra Shekhar for providing feedback at the early stage of this project.

Disclosure of Interests. The authors have no competing interests.

References

1. Brownlee, J.: A gentle introduction to ensemble learning algorithms. *Mach. Learn. Mastery* **7** (2021)
2. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint [arXiv:1412.3555](https://arxiv.org/abs/1412.3555) (2014)
3. Guess, A.M., Lockett, D., Lyons, B., Montgomery, J.M., Nyhan, B., Reifler, J.: 'fake news' may have limited effects beyond increasing beliefs in false claims (2020)
4. Kaliyar, R.K., Goswami, A., Narang, P.: Fakebert: fake news detection in social media with a bert-based deep learning approach. *Multimed. Tools Appl.* **80**(8), 11765–11788 (2021)
5. Karimi, H., Tang, J.: Learning hierarchical discourse-level structure for fake news detection. arXiv preprint [arXiv:1903.07389](https://arxiv.org/abs/1903.07389) (2019)
6. Mishra, S., Shukla, P., Agarwal, R.: Analyzing machine learning enabled fake news detection techniques for diversified datasets. *Wirel. Commun. Mob. Comput.* **2022**(1), 1575365 (2022)

7. Robertson, D.J., Shephard, M.P., Anderson, A., Huhe, N., Rapp, D.N., Madsen, J.K.: The psychology of fake news on social media, who falls for it, who shares it, why, and can we help users detect it? (2023)
8. Song, X., et al.: Time-series well performance prediction based on long short-term memory (LSTM) neural network model. *J. Petrol. Sci. Eng.* **186**, 106682 (2020)
9. Verma, P.K., Agrawal, P., Madaan, V., Prodan, R.: MCred: multi-modal message credibility for fake news detection using Bert and CNN. *J. Ambient. Intell. Humaniz. Comput.* **14**(8), 10617–10629 (2023)



A Child-Robot Interaction Experiment to Analyze Gender Stereotypes in the Perception of Mathematical Abilities

Madalina Croitoru^{1,2,3(✉)}, Pablo Laviron², Sio Bando², Eric Gilles²,
Amine Miled², Royce Anders³, Nathalie Blanc³, Gowrishankar Ganesh¹,
and Emmanuelle Brigaud³

¹ LIRMM, CNRS, University of Montpellier, Montpellier, France

² Faculty of Science, University of Montpellier, Montpellier, France

³ Epsilon, Paul Valery University, Montpellier, France

croitoru@lirmm.fr

Abstract. This study examines the use of social robotics in education, focusing on reducing the gender biases child students may have in the perception of their mathematical ability and potential. Our initial pilot with twenty 7-year-olds provides insights into the potential of combining AI with educational strategies. We discuss our findings, and experimental setup involving ChatGPT4, and future research directions.

Keywords: Social robotics · Human robot interaction · Educational science · Dialogues · Generative Artificial Intelligence

1 Paper in a Nutshell

In this paper, we explore the integration of Social Robotics [1] within educational contexts, particularly focusing on the pedagogical process for children [2]. This pilot study investigated gender biases in mathematics, through guided interactions with a robot that displayed explicit gender characteristics (e.g., pink skirt/dress and blue tie) to a group of twenty 7-year-old children. While further large-scale research is necessary for statistically significant results, its initial promising results lead to our proposition that such exploratory studies could herald a new era of advanced technology-assisted pedagogy, merging the most powerful techniques to-date, such as Artificial Intelligence or Robotics, with education sciences. The significance of our work lies in the fact that negative self-bias in educational contexts can exert substantial influence on students' academic outcomes and psychological health, often resulting in perpetuated negative cycles [7]. Hence, it is imperative to develop and apply effective methodologies to diminish these negative biases or, preferably, transform them into positive self-perceptions and increased self-efficacy.

The structure of this paper is divided into three main sections. The first section introduces foundational concepts in social robotics and cognitive human-robot interaction, contextualizing this research within child interactions. We describe the robot designed specifically for child interaction, equipped with a facial expression display on its screen. We also detail the technological workflow that enabled interactions with a Large Language Model (LLM) - specifically, ChatGPT4 - through verbal dialogues [3]. Additionally, this section discusses our methods for accurately reflecting the emotions detected during the child-robot conversations on the robot’s display screen. The second section elaborates on the technical elements introduced previously and delineates the experimental setup and the findings derived therefrom. Lastly, the discussion section contextualizes this study within the broader scope of potential future developments in the field.

2 Detecting Emotions for Social Robotics

Artificial Intelligence, particularly with the rapid emergence and development of generative text models, is profoundly revolutionizing the education sector [9]. This advance enables personalized learning strategies, interactive tutorial support, and makes educational resources more accessible and dynamic. Students can benefit from instant feedback on their work, while teachers have additional tools to assess and support each learner’s progress. Moreover, these technologies facilitate access to quality education, regardless of geographical constraints or available resources, thereby promising to democratize learning and open new avenues for teaching and continuous training.

Embodiment serves as a crucial bridge between robotics, psychology [4] and AI, endowing machines not only with the capacity to think or “understand” abstractly but also to act and interact with the physical world. This integration allows AI-equipped robots to perceive their environment, make decisions, and execute actions autonomously, thus mimicking the intelligent behaviors of living beings [8]. Advances in robotics are supported by several key technologies, including sensors (motion, touch, sound) that allow robots to perceive their environment, and verbal communication tools like TTS (Text-To-Speech) and STT (Speech-To-Text), crucial for human-robot interactions.

Large Language Models (LLMs) represent a significant recent advancement, designed to “understand”, interpret, and produce natural language extensively. Among these models, we find GPT (Generative Pre-trained Transformer), developed by OpenAI since 2018, stands out for its revolutionary architecture and exceptional text generation capabilities. This model is used in numerous applications in robotics [3,9] and opens promising horizons, especially in education by personalizing learning, facilitating access to information, and stimulating learner engagement through smooth and intuitive natural language interactions. The following workflow allows to engage in a dialogue with the robot, that will respond using the LLM’s capabilities:

- The process begins with the recording of the user’s voice, which is processed by the main program in `main.py` to create an audio file in WAV format.

- This audio file is then processed by a class in the file `whisper.py` which converts the voice to text via OpenAI’s Whisper API.
- The text is then analyzed to detect the corresponding emotion through a class in the file `emotion.py`, which will allow us to display the corresponding emotion on the robot’s face.
- Following this, the text along with all previous messages from the conversation are sent to the ChatGPT API to retrieve the response to our text, via the file `assistant.py`.
- We retrieve the response from ChatGPT, and then the file `tts.py` is used to make the robot speak using voice synthesis.
- The cycle continues as long as “Goodbye” is not expressed by the user.
- If “Goodbye” is said, then we make a final round of our loop and the robot will perform a farewell gesture as well, marking the end of our program.

2.1 Emotion Detection

To enhance our robot’s embodiment, we incorporated an emotional dimension and aimed for the QT robot’s LCD face screen to reflect varying emotions based on conversational context, mimicking human reactions. An experiment [5] revealed superior results from fine-tuning the GPT-3.0 model over prompt engineering on the GPT-3.5 model, with F1 scores of 0.90 versus 0.48, respectively. We replicated this experiment to identify the best approach for our needs, considering criteria such as execution time, computing power, cost, accuracy, and setup duration.

Initially, we explored using a GPT-3.5 model, trained on a vast dataset via the OpenAI API, for emotion analysis-known as fine-tuning. A challenge arose when no suitable French datasets for emotion (as opposed to sentiment) analysis were found, leading us to translate an English dataset from “dair-ai/emotion” [6]. This dataset, that included 16,000 entries labeled with six basic emotions (joy, sadness, anger, fear, love, and surprise), was adapted to our needs.

Fine-Tuning and System Implementation. We fine-tuned a selected model (gpt-3.5-turbo-1106) on this translated dataset, a process that took about 90 min and involved training over 563,922 tokens across three epochs. Once trained, the model could be deployed via OpenAI’s API, similar to other models. However, testing revealed an F_1 score of only 0.20, which was insufficient for practical use in the QT robot due to issues primarily stemming from data translation challenges and the non-deterministic nature of large language models.

New Approach and Dataset Creation. Given the suboptimal results from fine-tuning, we shifted to prompt engineering using GPT-4.0. We crafted a new dataset of 100 texts with associated emotions, enabling more meaningful statistical analysis. This dataset was developed with the help of ChatGPT, carefully controlling the model’s response diversity by setting the temperature parameter to 0.0 to maintain determinism.

Our emotional analysis involved applying the `analyse(text)` method to each text entry. Post-processing was necessary to standardize emotion outputs, which involved converting emotions to lowercase and removing non-alphabetic characters. To integrate the analyzer into the QT robot, we mapped the predicted emotions to the robot’s displayable emotions. Comparative analysis of predicted versus actual emotions indicated that while the model generally identified correct emotions, it struggled with certain categories like disgust and surprise. With GPT-4.0, we achieved a much improved F_1 score of 0.72, demonstrating the effectiveness of our methodological adjustments.

3 Math Stereotypes: A Gender Study

In this section we explain the experiment we carried out in order to analyze the gender stereotypes in the perception of mathematical abilities in another. We brought the QT robot in a class of 7 year olds and asked them to score a questionnaire that analyzed their gender stereotypes before and after the interaction with the robot. The pupils had a quick introduction to artificial intelligence (as shown below in Fig. 1) and chat bots, and had previous experience test the chat bot capabilities by asking simple arithmetic questions (please note that the pupils in the class were able to perform basic arithmetic operations themselves i.e. additions and subtractions).



Fig. 1. Demonstrating QT in front of a class of 7 year olds.

Half of the class interacted with the robot dressed as a girl (with a pink skirt) and the other half with the robot dressed as a boy (with a blue tie). The robot was displaying emotions according to the emotion detection described in the previous section and as shown in Fig. 2.

Before and after interaction with the robot the pupils were asked the following questions: “Do you like robots?”; “Are you able to finish your math exercises?”; “Is it important for you to succeed in mathematics?”; “Does doing a math assignment make you anxious?”; “Do you think boys are good at mathematics?”; “Do you

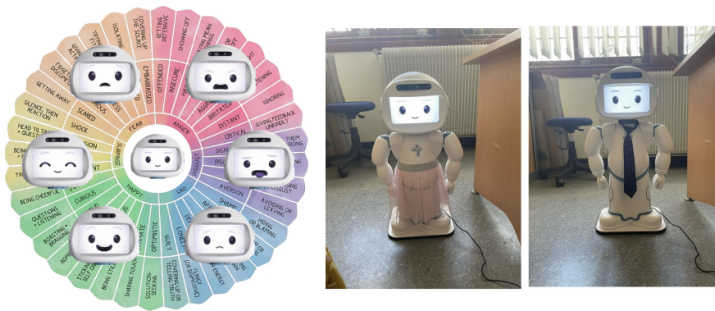


Fig. 2. Emotional capabilities of QT and the chosen outfit to represent common gender norms.

think girls are good at mathematics?”. These questions had to be answered by a 5 point SAM (Self-Assessment Manikin) scale using non-verbal pictorial assessment techniques. At the end of the experiment, the pupils were asked to assess the gender of the robot, and were also asked to report their own gender.

Eighteen participants answered the questions before and after the interaction with the QT robot. A quick overview of the main findings of the experiment show that: 79% of students adore robots; 69% of students thought the robot was a boy; 25% of students thought the robot was neither a boy nor a girl. Girls were more likely to think this; Girls and boys, both consider the other to have similar skills in mathematics (average of 3.6); Girls and boys, both have similar perception of their own level of math (average of 4.4); 17% of students reported being better able to finish their math exercises after the robot’s intervention; 78% of students reported being more capable to finish their math exercises after the robot’s intervention; 72% of students reported that succeeding in math was more important after the robot’s intervention; 66% of students in the “girl” robot experiment increased their opinion about girls’ abilities in math; 66% of students in the “boy” robot experiment increased their opinion about boys’ abilities in math.

More “alarming” results, worth investigating further in currently undergoing future work up, include the following:

- Regarding the question “Is it important for you to succeed in mathematics?” there is an effect, but the opposite of what was expected: less importance was placed on succeeding in mathematics after the presentation of QT (Time 1 Mean 4.556 Standard Deviation 0.984 and Time 2 Mean 4.056 Standard Deviation 1.211).
- The interaction did not reach statistical significance ($p = .18$), but after the robot presentation, girls changed their views and rated boys as being better at mathematics; boys also became more modest about their own abilities after the presentation compared to before.

4 Discussion

Several studies have shown that math gender stereotypes are formed very early. Such stereotypes can yield important economic consequences, and thus addressing these stereotypes is one of the main challenges one has to face from an educational point of view. In this paper, we propose to fight such stereotypes using Artificial Intelligence and Robotics that have made significant advancements in recent years, revolutionizing various sectors, including education. Educational robots, when combined with advanced language models like ChatGPT, can offer personalized and engaging learning experiences. We show how young pupils can be more sensitive to topics such as gender bias, mathematics performance and auto efficiency by interacting with a robot carefully crafted for such interactions.

The current limitations of robotics - AI fusion for education are, however, evident on several fronts. On one side, the complexity of real environments poses significant challenges in terms of perception and adaptability of AI systems, limiting their effectiveness to very specific or controlled contexts. Moreover, although progressing, the cognitive capabilities of AI are still far from the intelligence of humans, particularly concerning deep contextual understanding, creativity, and autonomous learning from diverse experiences.

References

1. Breazeal, C., Dautenhahn, K., Kanda, T.: *Social Robotics Springer Handbook of Robotics*, pp. 1935–1972. Springer, Heidelberg (2016)
2. Johnson, J.: Children, robotics, and education. *Artif. Life Robot.* **7**, 16–21 (2003)
3. Kim, C.Y., Lee, C.P., Mutlu, B.: Understanding large-language model (LLM)-powered human-robot interaction. arXiv preprint [arXiv:2401.03217](https://arxiv.org/abs/2401.03217) (2024)
4. Kushnir, A., Orkibi, H.: Concretization as a mechanism of change in psychodrama: procedures and benefits. *Front. Psychol.* **12**, 633069 (2021)
5. Qin, R., et al.: Enabling on-device large language model personalization with self-supervised data selection and synthesis. arXiv preprint [arXiv:2311.12275](https://arxiv.org/abs/2311.12275) (2023)
6. Saravia, E., Liu, H.C.T., Huang, Y.H., Wu, J., Chen, Y.S.: CARER: contextualized affect representations for emotion recognition. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 3687–3697. Association for Computational Linguistics, Brussels, Belgium (2018). <https://doi.org/10.18653/v1/D18-1404>, <https://www.aclweb.org/anthology/D18-1404>
7. Steele, J.R., Ambady, N.: “math is hard!” the effect of gender priming on women’s attitudes. *J. Exp. Soc. Psychol.* **42**(4), 428–436 (2006)
8. Wainer, J., Feil-Seifer, D.J., Shell, D.A., Mataric, M.J.: Embodiment and human-robot interaction: a task-based perspective. In: *RO-MAN 2007-the 16th IEEE International Symposium on Robot and Human Interactive Communication*, pp. 872–877. IEEE (2007)
9. Wang, J., et al.: Large language models for robotics: opportunities, challenges, and perspectives. arXiv preprint [arXiv:2401.04334](https://arxiv.org/abs/2401.04334) (2024)



Reinforcement Learning for Patient Scheduling with Combinatorial Optimisation

Xi Liu¹, Changgang Zheng², Zhen Chen¹, Yong Liao¹, Ren Chen³,
and Shufan Yang⁴^(✉)

¹ University of Science and Technology of China, Hefei, China

² University of Oxford, Oxford, UK

³ Anhui Medical University, Shushan, China

⁴ University of Leeds, Leeds, UK

s.f.yang@leeds.ac.uk

Abstract. Patient scheduling is a complex task that plays a crucial role in the quality of care. Effective scheduling management mitigates dissatisfaction among patients and physicians, serving as a crucial indicator. Traditionally, the approach to patient scheduling has been ad hoc, often overlooking key factors that may influence scheduling.

In this paper, we propose a reinforcement learning approach that utilises an early stopping mechanism which balances exploration and exploitation to provide combinatorial optimisation from both theoretical and experimental perspectives. Our study utilised datasets from NHS Scotland and The First Affiliated Hospital of Anhui Medical University to evaluate patient scheduling. Our results demonstrate that our Reinforcement Learning (RL) method with early stopping can successfully conduct preliminary practice on realistic examples of the General Practitioner (GP) Scheduling Problem and hospital scheduling issues.

Keywords: Scheduling · Reinforcement Learning · CO Problem

1 Introduction

Patient scheduling emerges as a pivotal component within the realm of healthcare management, involving allocating healthcare resources (like doctors, nurses, equipment, and rooms) to patients based on their needs and the availability of these resources. This intricate scheduling mandates a comprehensive consideration of various factors, including the urgency of medical needs, the availability of medical staff, the duration of appointments, and the operational hours of the healthcare facilities. The inherently unpredictable nature of healthcare demands necessitates a combinatorial optimisation (CO) strategy to enable flexible scheduling. This adaptability is crucial for accommodating emergency situations or unforeseen events, such as the sudden unavailability of medical practitioners. In this paper, we introduce an early stopping method in deep reinforcement learning approach to address the patient scheduling challenge as a

combinatorial optimisation problem, aiming to generate optimal scheduling solutions. This methodology can also be applied to other scheduling problems where objective functions are not well-defined or difficult to model mathematically.

Most challenging problems in the real world are large-scale and often subject to execution time constraints. Therefore, traditional algorithms encounter difficulties when applied to real-world challenging tasks. Recently, Deep reinforcement learning (DRL) has shown significant potential in overcoming the limitations of traditional approaches [3]. Many CO Problems can be transformed into sequence decision-making problems. For example, the TSP problem is to decide in what order to visit each city, and the shop scheduling problem is to decide in what order to process components on the machine. DRL is a very suitable solution for sequence decision-making. The main difficulty is the definition of the Markov Decision Process (MDP) in the CO problems. A diversity of reinforcement learning (RL) based heuristic method has demonstrated a promising solution as it does not require pre-solved examples of these hard problems [1]. However, so far there is a lack of guidance on how to utilise reinforcement learning (RL) to automatically learn good heuristics for various combinatorial problems since RL relies on estimating Q-value to form policy.

This paper proposes a framework leveraging a Deep-Q Network (DQN) [2] based reinforcement learning methodology to conceptualise healthcare management issues as combinatorial optimisation challenges. By employing real-world datasets, we demonstrate the feasibility of our approach in dynamically allocating hospital resources, outperforming traditional methods that struggle with the complexity and variability inherent in real-world scenarios. Significantly, our method exhibits superior adaptability to temporal changes, a critical attribute for effectively managing scheduling tasks in healthcare settings, where conditions can rapidly change. Furthermore, the utilisation of a synthetic dataset underscores our method's capability to manage problems encompassing a large number of variables and constraints, maintaining computational efficiency even as the scale of the optimisation challenge expands.

2 Experiment

2.1 System Setup

In this section, we present experimental results from two real-world scheduling tasks. For each experiment, we first clean and organise the data from real-world tasks, and build simulation environments based on different experimental datasets. We have two real-world tasks: the GP scheduling problem and the hospital patient appointments. From the perspective of the task, the environment we built is a grid world environment, but its basic parameters are different depending on the data (Fig. 1). Our datasets are provided by the National Health Service and the First Affiliated Hospital of Anhui Medical University (Tables 1 and 2).

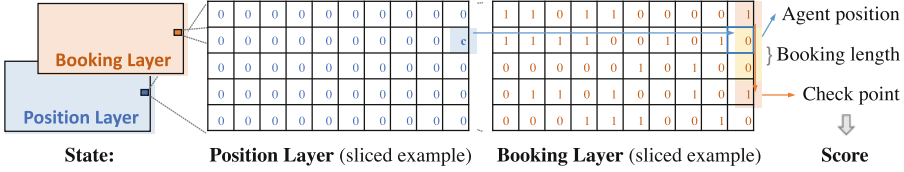


Fig. 1. State structure of Scheduling Problem.

Algorithm 1. DQN with Early Stopping(EDQN)

Require: $C \leftarrow$ a counter, $k \leftarrow$ the early stopping threshold, $Q^\pi \leftarrow$ the policy network, $Q^{\pi^*} \leftarrow$ the target network

- 1: Initialize experience replay buffer $\mathcal{B} = \emptyset$
- 2: In each episode:
- 3: Initialize state s_0
- 4: **for** $t = 1 \rightarrow T$ **do**
- 5: select action based on ϵ -greedy
- 6: Observe s_t, r_t, s'_t
- 7: Store the transition (s_t, a_t, r_t, s'_t) in the buffer \mathcal{B}
- 8: **if** r_t is not a good reward **then**
- 9: the value of counter C add one
- 10: **end if**
- 11: Sample experiences $\{(s_t, a_t, r_t, s'_t)\}$ from \mathcal{B} randomly
- 12: Set $y_i = r + \gamma \max_{a'} Q^{\pi^*}(s', a')$
- 13: Update weights of the neural network
- 14: **if** the counter C reaches threshold **then**
- 15: break
- 16: **end if**
- 17: **end for**

2.2 GP Scheduling

In the UK, the National Health Service (NHS) assigns patients to specific General Practitioners (GPs), with data sourced from the NHS website. The challenge is to efficiently schedule GPs for a group of patients, ensuring appointment slots of varying lengths are arranged to minimize wasted GP resources, similar to optimally placing blocks in a game of Tetris.

Task Description. The General Practitioner (GP) Scheduling Problem involves managing limited GP availability due to pre-booked appointments, while accommodating new patient requests of varying duration’s. The challenge is to maximise time slot utilisation while allowing GPs to reserve time for other activities, such as training. This study considers four types of consultations: face-to-face, home visits, telephone, and video consultations, with relevant data presented in Table 1.

Environment Settings. In our experiment, 100 GPs each work 8-h shifts divided into 32 slots of 15 min. Consultation times vary, as shown in Table 1,

Table 1. Appointments in the England by NHS statistics for one month.

scenarios	face-to-face	home visits	telephone	video
Number of appointments	15404951	171669	9184791	115725
Proportion of appointments	0.6192	0.0069	0.3692	0.0047
slots requirements	1	3	2	1

with ‘3’ indicating 3 consecutive slots needed. The problem is modeled as finding optimal positions in a 32×100 grid to minimize gaps while accommodating longer appointments. The GP environment has a $2 \times 32 \times 100$ state space with two layers: a booking layer showing occupied slots (‘1’ for occupied, ‘0’ for empty) and a position layer indicating the agent’s location. The agent moves in a finite action space (up, down, left, right), with rewards based on the agent’s position and the upcoming reservation status. Empty slots yield $+0.5$ rewards, while occupied ones give -0.1 . Extreme cases, like boundary movements or repeated jumps, result in a -0.5 reward. The DQN with Early Stopping (EDQN) pseudo-code is shown in Algorithm 1.

$$reward = \begin{cases} +0.5, & \text{if the position is empty} \\ -0.1, & \text{if the position is not empty} \\ -0.5, & \text{if the agent reaches the} \\ & \text{boundary or goes back} \end{cases} \quad (1)$$

2.3 Hospital Patient Appointments Scheduling

Unlike the UK, patients in China can book web-based appointments with specialists based on their requests. The hospital includes various specialized departments, such as internal medicine and surgery, allowing patients to seek treatment according to their conditions. While these systems aim to reduce wait times and improve efficiency, high demand limits access to specialists, potentially delaying treatment. An optimized booking system can help reduce these delays. Table 2 shows appointment data from the First Affiliated Hospital of Anhui Medical University, indicating that Internal Medicine and Surgery have the highest patient numbers, forming the basis for our experiment.

Task Description. Hospital patient appointment scheduling is similar to the General Practitioner (GP) Scheduling Problem, aiming to optimize hospital resource allocation. Based on data from The First Affiliated Hospital of Anhui Medical University, we created two environments for the Internal Medicine and Surgery departments. Each doctor works 8-h shifts, divided into 96 five-minute slots. Patient needs are categorized into three types based on duration: 1, 2, or 3 slots. The goal is to find continuous sequences of slots to meet these requirements, modeled as positions in a grid representing the number of clinics multiplied by 96.

Table 2. Summary of data from the First Affiliated Hospital of An Medical University.

Department	Internal Medicine	Surgical	Paediatrics	Dermatology
Number of appointments	3220	3097	1694	1986
Proportion of appointments	0.2439	0.2346	0.1283	0.1504
Number of departments	16	13	7	3
Department	Ophthalmology	Otolaryngology	Stomatology	DTCM
Number of appointments	715	843	1019	228
Proportion of appointments	0.0542	0.0638	0.0772	0.0173
Number of departments	2	2	4	4
Department	OG	AD	Haematology	
Number of appointments	228	79	92	
Proportion of appointments	0.0173	0.0060	0.0070	
Number of departments	8	1	1	

2.4 Results and Evaluation

We compared the performance of Deep Q-Network (DQN) and DQN with Early Stopping (EDQN) on GP scheduling and hospital patient appointment data. In our experiments, the neural network structure of the policy comprises three convolutional layers and two linear layers which use the Kaiming initialisation method to initialise parameters. The hyperparameter settings include a learning rate $lr = 1e - 4$, $\epsilon = 0.3$, $\epsilon_{decay} = 0.995$ and dynamic γ . At the beginning of exploration, the agent does not fully understand the environment. As the agent further explores the environment, it is more in line with the learning process to take the long-term future benefits into account in the value generated by the current behaviour. Therefore, it is more appropriate to use a dynamic γ . The dynamic γ is formulated as

$$\gamma = 1 - 0.9 \times (1 - \gamma) \quad (2)$$

Here, we set the initial $\gamma = 0.1$, and the maximum value of γ does not exceed 0.99.

Before the experiments, we need to establish the basic parameters of the environment. These include the number of pre-booking slots, the constant C representing the agent’s position in the position layer, and the patient demands. For the GP scheduling problem, we set the number of pre-booking slots to 750, and $C = 100$. The early stopping rule is that the agent obtains 10 negative rewards of -0.5 in total. For hospital patient appointments, we set the number of pre-booking slot to 100, and $C = 100$. The early stopping rule specifies that the agent receives 10 consecutive negative rewards of -0.5 .

The experimental results are presented in Fig. 2. Due to Early Stopping, the rewards obtained by EDQN were higher than those obtained by DQN from the beginning. Because the reward we set were relatively small, the curve of EDQN does not show an obvious upward trend (Fig. 2(a)). Since there is no significant

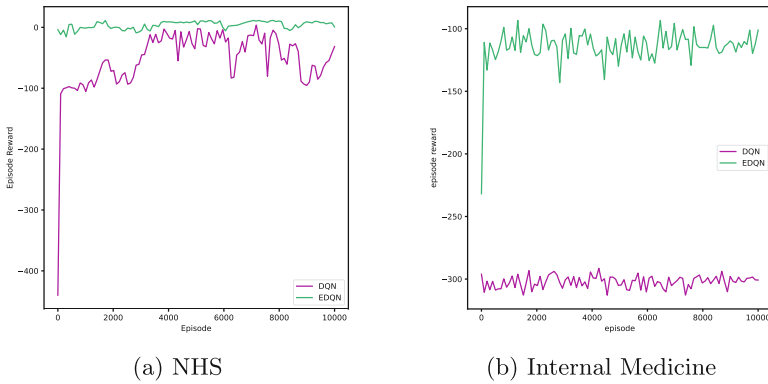


Fig. 2. Hospital patient appointment

difference in appointment data between Internal Medicine and Surgical department in hospitals, the experimental results are similar. The performance in Internal Medicine is shown in Fig. 2(b), and the performance in Surgical department shows the same trend. It is obvious that EDQN is better than DQN in the performance of GP appointment and hospital appointment data. This is because we eliminate redundant data and increase the agent's learning of sparse experience.

3 Conclusion and Future Work



Patient scheduling is a critical aspect of healthcare management, necessitating the efficient allocation of resources such as physicians, nursing staff, equipment, and facilities. This paper introduces a novel approach, conceptualizing resource allocation as a combinatorial optimization problem. We propose a reinforcement learning methodology, incorporating an early stopping mechanism, to optimize resource utilization and improve adaptability. By employing the Deep Q-network (DQN) algorithm, we address practical scheduling challenges, including the General Practitioner (GP) problem and hospital appointment scheduling. A key innovation of this approach is the integration of early stopping to enhance efficiency and accuracy. While our method may not achieve theoretical optimality, it demonstrates high efficacy in large-scale applications. Future research could extend this work by exploring continuous action spaces to address dynamic resource allocation and real-time scheduling problems.

References

1. Grinsztajn, N., Furelos-Blanco, D., Barrett, T.D.: Population-based reinforcement learning for combinatorial optimization. arXiv preprint [arXiv:2210.03475](https://arxiv.org/abs/2210.03475) (2022)
2. Hester, T., et al.: Deep q-learning from demonstrations. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32 (2018)
3. Kool, W., Van Hoof, H., Welling, M.: Attention, learn to solve routing problems! arXiv preprint [arXiv:1803.08475](https://arxiv.org/abs/1803.08475) (2018)



Nursing Activity Recognition for Automated Care Documentation in Clinical Settings

Frank Wallhoff^(✉) and Fenja T. Hesselmann

Institute for Assistive Technologies, Jade University of Applied Sciences,
Oldenburg/Wilhelmshaven/Elsfleth, 26121 Oldenburg (Oldb), Lower Saxony,
Germany

{frank.wallhoff, fenja.hesselmann}@jade-hs.de

<http://www.jade-hs.de>

Abstract. This paper introduces a concept for an assistance system that enables nursing staff in real-life clinical settings to reduce the time-consuming nursing documentation effort, e.g. from the morning routine. The overall goal is an AI-based documentation assistant that pre-fills the documentation record. An essential constraint in collecting and processing data is the use of sensors and features that preserve people's privacy and the acceptance of being observed.

The selected sensors are known body-worn acceleration sensors as well as far-infrared based thermal scans. During the training and evaluation phase, the use of an Azure Kinect is foreseen as well. A crucial intermediate step within the AI based concept is the autonomous identification and learning of actions that form an activity that is recognised as a part of the entire routine added to the documentation. Both open data sets and self-recorded material tailored to the use case are to be used for this purpose. By checking the automatically generated documentation results after each treatment by the nursing staff, additional training material is to be constantly generated during the application operation in order to improve the pattern recognition systems in the processing layer in the long term.

Keywords: Human Activity Recognition · Data Privacy · Assisted Documentation · Intelligent Agents

1 Introduction

In nursing care, it is important to document the performed actions. This documentation serves both as proof and for quality assurance. However, studies have shown that the effort required for documentation amounts to approx. 20–30% of working time [7]. It has also been shown that around 40 % of nursing staff suffer from burnout, partly due to the high workload [3]. There is consequently a high demand for the reduction of documentation work in care settings.

Therefore, an AI based concept shall be developed to establish an assistance system that supports nurses in clinical facilities in documenting their actions and activities, e.g. during the morning routine. Typically, these individually planned nursing activities include actions such as making beds, helping with personal hygiene, dressing, medical prescriptions (taking blood sample, applying wound dressings, changing bandages, measuring blood pressure, pulse and temperature), serving breakfast, optionally helping to prepare the breakfast, bringing morning medication and administering if necessary. A user study has shown, that assistance systems in care settings will be accepted if they preserve the users' privacy and do not entail any additional workload [2]. This essential requirement restricts the choice of deploy-able sensors, which do not have the best recognition results.

The rest of the paper is organised as follows. In the next section, applicable approaches for systems to recognise human activity in care scenarios with different sensor modalities are briefly recapitulated. This is followed by a consideration of the selection of suitable sensors and existing classification approaches. The proposed concept for a multimodal recognition framework is then presented. The paper concludes with a summary and the next steps.

2 Related Work

Research in the field of Human Activity Recognition (HAR) has grown in recent years. In particular, the research on nursing activity recognition has been identified as research field and four Nurse Care Recognition Challenges (NCRC) have been carried out as part of the Activity and Behavior Computing (ABC) series [6]. Although there are similar approaches in this domain, a holistic approach for the real-world task of documenting a care routine – with the restriction of non-optimal sensors – has not been addressed to our knowledge.

An interesting approach using a multi-modal transformer for nursing activity recognition, uses the modalities of video-based skeleton data and acceleration data [5]. It has been evaluated on the publicly available database of the Nurse Care Recognition Challenge from 2021 [6]. However, one limitation is the small amount of activities and the use of image-based motion capture data. An attempt was made to identify and correct missing entries in the care documentation based on smartphone data. In addition, a prediction was implemented to determine which upcoming activities are to be expected based on the missing entries. A neural network was used for this purpose. However, the use of smartphone data is a limitation [9].

Although transformer approaches and hierarchical or deep neural networks have generally turned out to be a central trend in research, there is not enough training material available for the recognition of care activities in a single-pass approach, at least at the moment. Therefore, the proposed concept foresees the use of sensors with machine learners that output their results to a knowledge-based intermediate layer. This intermediate layer can be re-trained during the application phase in a semi-supervised manner by incorporating user feedback when inspecting the generated care protocol.

3 Considerations on Sensor Selection

The acceptance of available sensors within a patient room in a clinical environment has to be ensured, which typically goes hand in hand with preserving the privacy of the patient and the nurse. Therefore, video-based systems are not an option due to the lack of acceptance. Sensor data must not be measured, stored or analysed externally except for the desired documentation use-case. Additionally, no complicated and expensive additional installation on-site should be required. On the other hand, the installed sensors should cover the field of interest without too many occlusions. In the best case, the hardware should be equipped with passive sensors to be applicable in clinical settings aiming to avoid problems with interference with other medical devices.

The use of acceleration or inertial measurement units (IMUs), image and depth-image based sensors (Azure Kinect) and far-infrared or thermal imaging sensors have received a lot of attention in the HAR research. It is important to note that the Kinect sensor must only be used during the collection of training data since it does not fulfill two requirements: it violates privacy by capturing pictures and it uses active near-infrared light. However, it will be useful to allow for a trans-modal learning approach to improve the performance of the two remaining sensors.

4 Self-supervised and Reinforcement Learning

For each of the above mentioned sensor modalities, many different competing recognition approaches exist, each with a specific strength. There are also machine learning approaches and specialized databases for each of the proposed recognition techniques that can be used to train sophisticated, pre-built classifiers for this use case.

However, besides the transformer approach mentioned above, there are currently only few databases that provide corresponding sensor data from two or more modalities at the same time. This is especially the case for activities that are less frequently classified and do not belong to the group of activities “standing, sitting, lying down”. Thus, there is a need to collect different data streams. For this, the Azure Kinect data can be used as gold standard for motion recognition, since its accuracy has proven to be reliable [1]. With the help of the Kinect, the other sensor modalities can be annotated using self-supervised learning.

In addition, reinforcement learning with user feedback can be used to further optimise the model. Users provide continuous feedback on the model’s predictions, for example by confirming or correcting the recognised activities. This feedback is used to adapt the model’s decision-making strategies and increase recognition accuracy [8]. By combining these approaches, the self-supervised learning component of the system can pre-train models on large data sets, ensuring robust initial performance. The reinforcement learning component then adapts these models to the user’s behaviour and preferences in real time.

5 Proposed Concept

The concept for modeling care activities is outlined by an example: The morning routine consists of a varying sequence of many observable and (by the chosen sensor) non-observable activities, whereby each activity consists of several actions.

At this point, a comparison is suggested: the complex activities to be recognised during an entire morning routine can be interpreted analogously as a sequence of individual words in a sentence. While words are further modeled from a combination of phonemes, activities can be composed of a sequence of individual actions. Therefore, semi- or fully automated annotation techniques such as proposed by Florenc et al. [4] can be applied.

The level of self-care of a patient can gradually change for the same person within a short period of time. It is assumed that a typical care routine is accompanied by the use of a tablet. By entering the room, the nurse starts the documentation. When leaving the room, the documentation will be completed. With a low latency period, the monitored routine will be recapitulated and the documentation form will be compiled, which is verified by the nurse on the tablet. The result can be used later for re-training the system since the nurse can check all entries at the end of the routine and correct them if necessary.

To ensure data privacy, sensors are used that cannot draw any conclusions about the person, e.g. acceleration sensors or thermal imaging cameras. However, as information is missing depending on the sensor, e.g. the position of the person in the room, AI methods must be used to reconstruct the current situation and the actions performed using several sensors.

From a functional point of view, the recognition framework has to process all available inputs, i.e. acceleration data, thermal images, vital data (temperature etc.) and direct input from tablet (touch gestures) as depicted in Fig. 1. The skeleton data (from Azure Kinect) is only used during the initial training phase and therefore plotted with a dashed line.

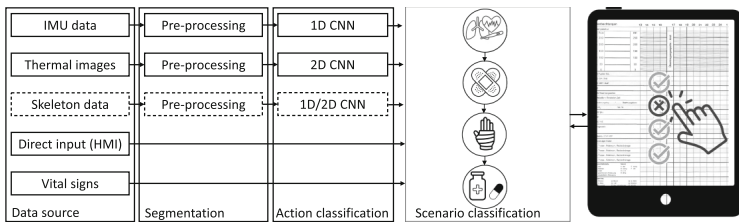


Fig. 1. Overall holistic functional scheme of the proposed approach.

The proposed training of the documentation system will be carried out in three phases. In the first phase, the classification modules for each of the modalities (acceleration, thermal images and skeleton data) will be trained separately on several publicly available databases (e.g. from KAGGLE) on the

pre-processing and segmentation unit. As training material, sample material of isolated actions will be used. Supervised learning algorithms that classify the activities will be used for training.

In the second phase, an embedded training using more complex data with activities (consisting of sequences with actions) will be used. In this phase, it is important that the activities are modelled autonomously by the pre-trained actions. It is expected that the skeleton data from the Azure Kinect are the most reliable and thus control the learning process and help to boot-strap the classifiers during the supervised training. The parameters, i.e. the type of actions as well as their temporal order for the activity classifier, are also learned in this phase. The activities will then be modelled in a next step similar to a finite state machine. The advantage with this approach is that the action sequences can be hand-crafted and initialised manually at the beginning. Furthermore, an inspection of the action frequency and sequence order can provide information regarding the plausibility of a recognition result. This would be possible with hierarchical networks.

The following example illustrates the idea: In the first phase, dynamic and static models for the actions A are trained along with the segmentation unit to find the action boundaries: $A \in [\textit{grasping}, \textit{walking}, \textit{standing away from the patient}, \textit{standing close to the patient}]$. The activity S **Giving medication** can be represented by variants of the sequence $S = \{\textit{standing away from the patient grasping walking}, \textit{standing close to the patient}\}$.

In the third phase, the remaining classifiers are re-trained using sample material that is tailored to this use case.

During the application phase of the concept, the recognised elements are summarised after each routine in the form of the care documentation, that has to be verified by the nurse. By quickly correcting possibly wrong recognised activities, additional training material is continuously created.

In order to train and test the machine learning models in the proposed approach, sophisticated training material is needed. Therefore, twelve qualified nurses have performed examples of morning routines on a high-fidelity manikin instead of a human patient in a laboratory setting. Currently, we are focusing on the following set of 13 activities: Addressing the patient, Checking vital signs (measuring blood pressure, pulse, temperature), Brush teeth, Washing the face, Dressing and undressing in bed, Removing and applying the infusion, Washing the upper/lower, Changing bandages/plasters, Giving medication, Changing a patient's position.

The implementation and training of the proposed concept is ongoing.

6 Summary

This paper presents a concept for recognising nursing activities in the clinical setting. To this end, the considerations for sensor selection were explained and, based on this, three sensor modalities were presented for closer selection. A scenario with the activities to be recognised and the overall scheme were

then presented. In addition to the pre-processing and segmentation of the data streams, this includes an action classification and then an activity classification. An important part of the concept is the re-training of the model, in which the carer checks the recognised activities. To evaluate the concept, data sets from twelve nurses were recorded with different sensors.

The next step is to implement the concept presented. This involves implementing the three phases of the HAR recognition pipeline, in particular the segmentation unit and the reasoning layer. The final step of the implementation is to ensure the ability to re-train the AI through user feedback. This provides a semi-supervised self-improvement.

Acknowledgements. This study was supported by the Lower Saxony Ministry for Science and Culture with funds from the governmental funding initiative zukunft.niedersachsen of the Volkswagen Foundation, project “Data-driven health (DEAL)”.

The experiment complied with the Declaration of Helsinki and was approved by the ethics committee of the University of Oldenburg with approval identifier Drs.EK/2024/027.

Furthermore the authors would like to express their thanks to the undergraduate student Fabian Kessener for building the training database.

References

1. Bertram, J., et al.: Accuracy and repeatability of the microsoft azure Kinect for clinical measurement of motor function. *PloS One* **18**, e0279697 (2023). <https://doi.org/10.1371/journal.pone.0279697>
2. Bruns, F.T., Pauls, A., Koppelin, F., Wallhoff, F.: Activity recognition of nursing tasks in a hospital: requirements and challenges. In: Salvi, D., Van Gorp, P., Shah, S.A. (eds.) *PH 2023. LNICS, Social Informatics and Telecommunications Engineering*, vol. 572, pp. 235–243. Springer, Cham (2024). https://doi.org/10.1007/978-3-031-59717-6_16
3. Dall’Ora, C., Ball, J., Reinius, M., Griffiths, P.: Burnout in nursing: a theoretical review. *Hum. Resour. Health* **18**(1), 41 (2020). <https://doi.org/10.1186/s12960-020-00469-9>
4. Demrozi, F., Turetta, C., Machot, F.A., Pravadelli, G., Kindt, P.H.: A comprehensive review of automated data annotation techniques in human activity recognition (2023). <https://doi.org/10.48550/ARXIV.2307.05988>
5. Ijaz, M., Diaz, R., Chen, C.: Multimodal transformer for nursing activity recognition. In: *Proceedings: IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2064–2073 (2022). <https://doi.org/10.1109/CVPRW56347.2022.00224>
6. Inoue, S., Alia, S.S., Lago, P., Goto, H., Takeda, S.: Nurse care activities datasets: in laboratory and in real field (2020). <https://doi.org/10.21227/jem3-ap07>
7. Joukes, E., Abu-Hanna, A., Cornet, R., de Keizer, N.F.: Time spent on dedicated patient care and documentation tasks before and after the introduction of a structured and standardized electronic health record. *Appl. Clin. Inform.* **09**(01), 046–053 (2018). <https://doi.org/10.1055/s-0037-1615747>

8. Kaufmann, T., Weng, P., Bengs, V., Hüllermeier, E.: A survey of reinforcement learning from human feedback (2024). <https://arxiv.org/abs/2312.14925>
9. Okuda, R., Xia, Q., Maekawa, T., Hara, T., Inoue, S.: Activity prediction method for nursing care records with missing entries. *Int. J. Act. Behav. Comput.* (2024)



Exploring Efficient Job Shop Scheduling Using Deep Reinforcement Learning

Reshma Maharjan^(✉), Per-Arne Andersen, and Lei Jiao

Department of ICT, University of Agder, Kristiansand, Norway
{reshma.maharjan,per.andersen,lei.jiao}@uia.no

Abstract. This paper evaluates four Reinforcement Learning (RL) algorithms, namely, Proximal Policy Optimization (PPO), Policy Gradient (PG), Advantage Actor-Critic (A2C), and Asynchronous Advantage Actor-Critic (A3C), for solving the Job Shop Scheduling Problem (JSSP) using Lawrence, Dermikol, and Taillard datasets. Experiments show that PPO consistently outperforms traditional dispatching rules and state-of-the-art methods, achieving 6–9 times lower optimality gaps than traditional algorithms and 2–3 times lower than state-of-the-art approaches across all datasets. These results demonstrate the potential of RL, particularly PPO, in enhancing scheduling optimization for the JSSP.

Keywords: Job Shop Scheduling · Reinforcement Learning · Proximal Policy Optimization · Optimality Gap · Scheduling Optimization

1 Introduction

The Job Shop Scheduling Problem (JSSP) involves sequencing multiple operations across various machines to minimize the total completion time, or makespan, in fields like manufacturing, healthcare, and beyond. Many studies in [10–12, 19] have addressed the JSSP. Traditional mathematical optimization techniques like mixed integer programming [12] and integer linear programming [10] become impractical for large-scale or dynamic scheduling due to the curse of dimensionality and lack of real-time modification. Instead, heuristic methods such as simulated annealing [19], and genetic algorithms [11] provide high-quality solutions quickly but are unsuitable for dynamic problems due to their need for reapplication.

Reinforcement learning (RL) has recently gained significant attention in addressing JSSPs due to its ability to learn effective scheduling policies through environmental interaction. RL approaches excel in adapting to dynamic and uncertain scenarios, making them particularly well-suited for real-world manufacturing settings. Initially applied to the Traveling Salesman Problem using Pointer Networks [1], RL has since evolved to encompass a wide range of techniques, including heuristic learning [4], attention-based models [7], and effective production scheduling using Deep Q-Network (DQN) agents [17]. Recent

advancements in Deep RL have further expanded its capabilities, incorporating graph neural networks [20], disjunctive graphs, double DQN, and prioritized experience replay [5]. These innovations have shown considerable promise in job-shop scheduling, demonstrating improved performance and adaptability. However, despite these advances, tackling large-scale, complex JSSP instances remains a significant challenge, driving ongoing research in the field.

This study, supported by the EU Horizon 2020 ‘‘Rhinceros’’ project [16] on automating car battery dismantling, models the process as a JSSP. It validates the approach by replicating prior experiments from [15] and compares different RL algorithms to identify the most effective method for large-scale scheduling. The goal is to improve scheduling efficiency and reduce makespan by training a dispatcher agent to choose jobs sequentially using DRL techniques. In this paper, we make the following contributions:

1. This study evaluates the applicability of several RL algorithms by conducting a comparative study to solve a JSSP.
2. We conduct numerous experiments on hyperparameter sensitivity to optimize the solution, and we identify a suitable model, namely Proximal Policy Optimization (PPO), to solve the JSSP for a single agent.

2 Environment and Model Configurations

Each JSSP instance in its traditional form consists of two sets of constants, jobs \mathcal{J} and machines \mathcal{M} . Every job $J_i \in \mathcal{J}$ is composed of n_i operations ($O_{i,1} \rightarrow O_{i,2} \rightarrow \dots \rightarrow O_{i,n_i}$) that need to be completed in a specific order, where each element $O_{i,j}$ ($1 \leq j \leq n_i$) is called an operation of J_i . Every machine $M_k \in \mathcal{M}$ can operate on an operation $O_{i,j}$ with the processing time $p_{i,j,k} \in \mathbb{N}$. Each job has a number of operations and the time taken to complete that operation by different machines may have distinct processing durations. The objective is to ascertain which processes should be scheduled in what order to minimize the makespan or overall execution time. Several constraints govern the problem: machines can only handle one task at a time, operations within a job must follow a specific order, and once started, tasks must run to completion without interruption. These constraints ensure efficient and orderly production scheduling, crucial for optimizing manufacturing and operational processes.

Environment: The JSSP RL environment uses a single-agent dispatcher to assign jobs to machines, with actions constrained by machine availability and job completion. The agent either schedules jobs or uses *No-Op* to advance time. Efficient scheduling is achieved by prioritizing non-final jobs and minimizing delays through careful use of *No-Op*. In the JSSP, the reward function focuses on active job processing time rather than minimum makespan to provide more frequent feedback and accelerate learning. The reward is calculated as $R(s, a) = p_{i,j,k} - \sum_{M_k \in \mathcal{M}} \text{empty}_{M_k}(s, s')$, where s represents the current state and s' the next state after action a . a is the j th operation of job J_i with processing time

$p_{i,j,k}$ for machine M_k , and $\text{empty}_{M_k}(s, s')$ is the amount of time the machine M_k being idle while transitioning from the state s to s' .

State Space: The state representation for the JSSP is represented by a matrix. Each row in this matrix represents a job that includes seven attributes: whether the job can be assigned to a machine, the remaining time for the current operation, the percentage of completed operations, the remaining time until job completion, the time until the next required machine is available, the idle duration since the last operation, and the cumulative idle time throughout the job’s schedule. These attributes collectively provide a comprehensive overview of each job’s status, progress, and machine availability, helping the RL agent to understand the job shop scheduling environment and make informed decisions, thus adhering to the Markov property for effective scheduling optimization.

Action Selection: State representations are converted from tabular matrices to vectors for simpler action distribution and state-value estimation via the agent’s Multi-Layer Perceptron (MLP). A mask applied to neural network outputs assigns minimal negative values to invalid actions before the softmax function, guiding the agent towards feasible actions. This approach, highlighted by Huang et al. [6], enhances performance in JSSPs by focusing the agent on optimal, legal actions.

3 Experimentation

This section outlines the experimentation procedures used to evaluate the performance of various RL algorithms and baseline methods on JSSP benchmark datasets, namely Taillard [14], Demirkol [3], and Lawrence [8].

Baselines: The agent is evaluated against three dispatching rules: *First In First Out* (FIFO), which processes jobs in the order they arrive; *Most Work Remaining* (MWKR), which prioritizes jobs with the highest remaining processing time; and *Shortest Processing Time* (SPT), which favors jobs with the shortest processing time. The benchmarks, namely, Zhang et al. [20], Han et al. [5], and Wu et al. [18], help assess the agent’s performance. The Google OR-Tools CP-SAT solver is used to obtain the optimal solution.

Implementation: The paper uses the open-source RL library RLlib and TensorFlow to implement RL on the JSSP environment. WandB [2] is utilized for hyperparameter optimization and data logging. Four RL algorithms-PG, PPO, A2C, and A3C-are trained on an NVIDIA H100 GPU server.

Evaluation Configuration: The model is trained on JSSP instances from Taillard’s, Demirkol’s, and Lawrence’s datasets using distinct MLP architectures for state-value prediction and action selection, each with two hidden layers of 256 neurons and ReLU activation. Optimized with PPO-specific parameters, these networks feature clipping at 0.5, ten epochs for updates, and policy and value function coefficients of 0.5 and 0.8. A linear decay scheduler adjusts the learning rate from 6.6×10^{-4} to 7.8×10^{-5} and the entropy coefficient from 2.0×10^{-3}

to 2.5×10^{-4} . The Adam optimizer is used with a discount factor γ of 1. The experimental setup for PG, A2C, and A3C is similar, with MLP architectures, linear decay scheduling, and the Adam optimizer. However, unlike PPO, these algorithms do not require specific parameters such as clipping or loss coefficients.

Performance Metrics: To evaluate considered algorithms, two main metrics are adopted: *Makespan* and *optimality gap*. The average optimality gap (Opt_{gap}) [9] is given by $Opt_{gap} = \frac{Makespan - Makespan^*}{Makespan^*} \times 100$, where *Makespan* is the makespan obtained from different algorithms, and *Makespan** is either optimal or the best-known solution. The optimal solutions are derived using Or-Tools [13]. This gap assesses how close the solution is to the optimal one.

Table 1. Makespan Comparison of considered RL Algorithms on Standard Benchmark Instances.

Size	Instance	PPO	PG	A2C	A3C
20 × 10	la30	1356	1567	1630	1759
30 × 10	la35	1895	2043	2113	2216
15 × 15	la40	1323	1457	1519	1614
40 × 15	dmu21	4863	5598	5600	6363
40 × 20	dmu26	5835	6316	6235	7153
50 × 15	dmu31	6221	7189	7143	7729
50 × 20	dmu36	6693	7487	7482	8250
30 × 20	ta42	2146	2625	2631	3046
50 × 15	ta52	2957	3402	3425	3724
50 × 20	ta62	3485	3776	3821	4211
100 × 50	ta72	5860	6385	6082	6263

Results: First, the agent is trained using four RL algorithms across three benchmark datasets, addressing diverse problem instances. Results in Table 1 reveal that as instance size grows, so does the makespan, indicating increased difficulty. PPO outperforms other algorithms, with PG second, and A2C and A3C lagging behind. The clipped surrogate objective function, which stabilizes learning, helps PPO explore the action space and learn a precise policy, contributing to its effectiveness.

We also present comparisons in Table 2 for different approaches, including the baselines described above, in terms of makespan and optimality gap. Across this tables, PPO consistently demonstrates superior performance compared with the baseline heuristics, except “dmu26” and “ta72” where MWKR and FIFO is outperforming respectively. The comparison with the other state-of-art approaches is challenging because they often have different goals, settings, and algorithms. These variations can make it difficult to assess their performance accurately and

Table 2. Comparison of RL Algorithm and Baselines on Standard Benchmark Instances: Makespan and Optimality Gap Analysis.

Instance		PPO	FIFO	MWKR	SPT	Zhang et al. (2020)	Han et al. (2020)	Wu et al. (2024)	Or-Tool
la30	Makespan	1356	1648	1533	1775	-	1417	1395	1355
	<i>Opt_{gap}</i>	0.07	21.62	13.14	31	-	4.58	2.95	-
la35	Makespan	1865	2138	2073	2464	-	1941	1896	1800
	<i>Opt_{gap}</i>	3.61	18.78	15.17	36.78	-	7.83	5.33	-
la40	Makespan	1323	1435	1450	1481	-	1336	1314	1222
	<i>Opt_{gap}</i>	8.27	17.43	18.66	21.19	-	9.33	8.26	-
Average <i>Opt_{gap}</i>		3.47	25.95	17.42	31.35	-	7.65	5.92	-
dmu21	Makespan	4863	5674	5325	6378	5314	5255	-	4280
	<i>Opt_{gap}</i>	11.03	29.54	21.58	45.62	21.32	19.98	-	-
dmu26	Makespan	5835	6125	5567	6725	6241	5695	-	4986
	<i>Opt_{gap}</i>	17.03	22.84	11.65	34.88	25.17	14.22	-	-
dmu31	Makespan	6221	6817	6523	7666	6639	6588	-	5642
	<i>Opt_{gap}</i>	10.26	20.83	15.62	35.87	17.67	16.77	-	-
dmu36	Makespan	6693	7422	6837	7577	7328	6859	-	5973
	<i>Opt_{gap}</i>	12.05	24.26	14.47	26.85	46.52	14.83	-	-
Average <i>Opt_{gap}</i>		12.13	25.39	16.54	34.83	32.97	16.32	-	-
ta42	Makespan	2146	2578	2401	2783	2664	2351	2305	2020
	<i>Opt_{gap}</i>	6.23	27.62	18.86	37.77	31.88	16.39	14.11	-
ta52	Makespan	3066	3549	3585	3457	3599	3263	3155	2756
	<i>Opt_{gap}</i>	7.29	21.15	25.22	25.47	21.23	17.16	10.89	-
ta62	Makespan	3485	3652	3489	4075	3722	3489	-	3042
	<i>Opt_{gap}</i>	14.56	20.05	14.96	33.96	22.35	14.69	-	-
ta72	Makespan	5860	5610	5625	6506	5695	5746	-	5531
	<i>Opt_{gap}</i>	5.95	1.43	1.70	17.63	2.97	3.89	-	-
Average <i>Opt_{gap}</i>		8.23	17.56	15.18	28.70	19.61	13.03	12.5	-

meaningfully. But stated in paper [5], the agent was trained using the same benchmark instances, allowing for a more direct comparison, their comparison provides an improved evaluation. Analyzing the table with respect to optimality gap, it is evident that the PPO-based approach achieves a 6–9 times lower optimality gap compared with traditional scheduling algorithms and a 2–3 times lower optimality gap than the state-of-the-art approaches in all three datasets, underscoring its superiority in addressing the JSSP.

4 Conclusions

This study evaluated RL algorithms—PPO, PG, A2C, and A3C—in solving JSSPs using instances from the Lawrence, Demirkol, and Taillard datasets.

Results demonstrate that the PPO algorithm consistently outperforms other methods across diverse instance sizes and datasets. PPO exhibited remarkable robustness and adaptability, achieving 6–9 times lower optimality gaps than traditional algorithms and 2–3 times lower than the state-of-the-art approaches. The study concludes that PPO offers a promising solution for dynamic and complex scheduling tasks, with significant potential for industrial applications through enhanced optimization and planning capabilities.

References

1. Bello, I., Pham, H., Le, Q.V., Norouzi, M., Bengio, S.: Neural combinatorial optimization with reinforcement learning. arXiv preprint [arXiv:1611.09940](https://arxiv.org/abs/1611.09940) (2016)
2. Biewald, L., et al.: Experiment tracking with weights and biases. Software available from wandb.com **2**(5) (2020)
3. Demirkol, E., Mehta, S., Uzsoy, R.: Benchmarks for shop scheduling problems. *Eur. J. Oper. Res.* **109**(1), 137–141 (1998)
4. Deudon, M., Cournut, P., Lacoste, A., Adulyasak, Y., Rousseau, L.-M.: Learning heuristics for the TSP by policy gradient. In: van Hoeve, W.-J. (ed.) CPAIOR 2018. LNCS, vol. 10848, pp. 170–181. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-93031-2_12
5. Han, B.A., Yang, J.J.: Research on adaptive job shop scheduling problems based on dueling double DQN. *IEEE Access* **8**, 186474–186495 (2020)
6. Huang, S., Ontañón, S.: A closer look at invalid action masking in policy gradient algorithms. arXiv preprint [arXiv:2006.14171](https://arxiv.org/abs/2006.14171) (2020)
7. Kool, W., Van Hoof, H., Welling, M.: Attention, learn to solve routing problems! arXiv preprint [arXiv:1803.08475](https://arxiv.org/abs/1803.08475) (2018)
8. Lawrence, S.: Resource constrained project scheduling: an experimental investigation of heuristic scheduling techniques (supplement). Carnegie-Mellon University, Graduate School of Industrial Administration (1984)
9. Lee, J., Kee, S., Janakiram, M., Runger, G.: Attention-based reinforcement learning for combinatorial optimization: application to job shop scheduling problem. arXiv preprint [arXiv:2401.16580](https://arxiv.org/abs/2401.16580) (2024)
10. Manne, A.S.: On the job-shop scheduling problem. *Oper. Res.* **8**(2), 219–223 (1960)
11. Mattfeld, D.C., Bierwirth, C.: An efficient genetic algorithm for job shop scheduling with tardiness objectives. *Eur. J. Oper. Res.* **155**(3), 616–630 (2004)
12. Morinaga, E., Tang, X., Iwamura, K., Hirabayashi, N.: An improved method of job shop scheduling using machine learning and mathematical optimization. *Procedia Comput. Sci.* **217**, 1479–1486 (2023)
13. Perron, L., Furnon, V.: Or-tools. <https://developers.google.com/optimization> (2019). Accessed 28 June 2024
14. Taillard, E.: Benchmarks for basic scheduling problems. *Eur. J. Oper. Res.* **64**(2), 278–285 (1993)
15. Tassel, P., Gebser, M., Schekotihin, K.: A reinforcement learning environment for job-shop scheduling. arXiv preprint [arXiv:2104.03760](https://arxiv.org/abs/2104.03760) (2021)
16. Union, E.: Rhinoceros (2022). <https://www.rhinoceros-project.eu/>. Accessed 15 May 2024
17. Waschneck, B., et al.: Optimization of global production scheduling with deep reinforcement learning. *Procedia CIRP* **72**, 1264–1269 (2018)

18. Wu, X., Yan, X., Guan, D., Wei, M.: A deep reinforcement learning model for dynamic job-shop scheduling problem with uncertain processing time. *Eng. Appl. Artif. Intell.* **131**, 107790 (2024)
19. Yim, S.J., Lee, D.Y.: Scheduling cluster tools in wafer fabrication using candidate list and simulated annealing. *J. Intell. Manuf.* **10**, 531–540 (1999)
20. Zhang, C., Song, W., Cao, Z., Zhang, J., Tan, P.S., Chi, X.: Learning to dispatch for job shop scheduling via deep reinforcement learning. In: *Advances in Neural Information Processing Systems*, vol. 33, pp. 1621–1632 (2020)



Respiratory Disease Detection Using Deep Convolutional Transformer Models

Holly Burrows^(✉), Mahdi Maktabdar Oghaz, and Lakshmi Babu Saheer

Anglia Ruskin University Cambridge, CB1 1PT, Cambridge, UK
hb643@pgr.aru.ac.uk

Abstract. Respiratory diseases are the leading cause of many hospital admissions and account for a significant portion of fatalities each year in Europe. Long-term respiratory conditions such as asthma affect millions of people globally. A crucial element in diagnosing and monitoring respiratory disease is assessing lung sounds known as respiratory auscultation. Nevertheless, this process can be automated using Deep Learning (DL) techniques to alleviate the strain on healthcare services. This work offers a comparison of various State-Of-The-Art DL models', namely ConvNeXt, and Vision and Swin transformers for predicting respiratory diseases asthma and COPD and healthy controls from a novel dataset of lung sound recordings represented by melspectrograms. The research concludes that using ConvNeXt in its' Base configuration outperforms other networks with metrics including accuracy, sensitivity, precision, specificity and F1 score.

Keywords: Lung sounds · Respiratory diseases · Transformer models

1 Introduction

The World Health Organisation reports the third most significant cause of death in Europe is respiratory disease [23]. The highest number of hospital admissions are due to Chronic Obstructive Pulmonary Disease (COPD), Asthma, Pneumonia, and Influenza [1]. The European Respiratory Society estimates that COPD affects approximately 44 million people in Europe, and the economic burden of the disease is around €141 billion per year, considering direct and indirect medical costs, and intangible expenses such as reduced quality of life [14]. Novel solutions using Artificial Intelligence and deep learning could automate the process of diagnosis and help to reduce clinician workload. Predictive models can analyse respiratory recordings to distinguish respiratory diseases and identify abnormalities. In this vein, this research investigates State-of-the-Art (SOTA) deep learning techniques to recognise respiratory diseases COPD and asthma from a novel dataset of lung sound recordings.

2 Related Work

Much of the literature investigating respiratory sound classification makes use of the Respiratory Sound Database [21], a dataset of lung sounds with disease labels. It can be used for Adventitious Lung Sound Classification (ALSC) or Respiratory Disease Classification (RDC) with class labels healthy, chronic, non-chronic. ALSC using this dataset is well explored within the literature, demonstrating success using melspectrograms [10], and features extracted with algorithms such as constant-Q transform (CQT), STFT, Mel-STFT, and empirical mode decomposition [4]. Deep learning techniques such as CNN have been explored [15], compared to machine learning methods such as decision trees [3], where using audio features achieves 85% accuracy, however this suffers significantly when the same method is applied to RDC. CNNs have also been used to predict spectrograms for RDC [19], and variations of Recurrent Neural Networks such as LSTM, GRU, BiLSTM, and BiGRU [17]. The impact of multiple spectrograms was investigated by [20], generating Morse and Amor scalograms to capture high-resolution frequencies and variance over time. An inception-based architecture achieved 87% and 85% specificity and sensitivity respectively for RDC multi-class task. Transfer learning has been investigated for RDC multi-class, demonstrating 92.57% average sensitivity and specificity with ResNet and melspectrograms [16]. Combining samples to reframe as a binary task, [18] use a Mixture-of-Experts method classifying gammatone spectrograms to reach average sensitivity and specificity of 92%, making it one of the best in the domain for this task. This dataset provides diagnosis for each subject, facilitating disease classification i.e. COPD, asthma, pneumonia. However, compared to aforementioned tasks, there is a significantly smaller body of work investigating this problem. Due to the huge imbalance between classes, SOTA research for this task remove classes with the fewest instances, namely asthma and LRTI, resulting in a far more balanced dataset: in work by [2] an F1 score of 0.96 was achieved modelling a GRU and temporal features. A benchmark study by [8] combines this data and a proprietary dataset of lung sounds, increasing the number of asthma samples. Robust evaluation of CNN and long short-term memory methods showed an average precision of 98.70% across six diseases. Emulating self-attention mechanisms from NLP transformers, the Vision Transformer (ViT) [7] was developed to approach image classification as sequence prediction using patching to create regions. ALSC using transformer-based methods is in its infancy but has proven to be more effective than hybrid CNN-RNN models for COVID-19 detection from melspectrograms [5]. Audio spectrogram transformer models have been used to classify different types of cough [11] with good performance (F1 score 0.80). Branch attention methods for feature enhancement of time and frequency information have been exploited with a transformer, showing good performance to identify healthy, asthma, COPD, and pneumonia [22]. In the literature, fewer works are using this dataset to classify respiratory diseases. Moreover, there are even fewer works investigating transformer networks for this task. Research using this dataset demonstrate predictive capabilities tend to suffer due to the significant class imbalance. Therefore, we propose respiratory

disease classification using a novel lung sounds dataset, combined with limited open access datasets and SOTA convolutional and transformer networks.

3 Experimental Setup

3.1 Data

This research creates a combined dataset from publicly available data and a novel dataset of lung sounds. We use the Respiratory Sound Database [21], containing lung sounds from healthy people, and those with various respiratory diseases, including asthma and COPD. There are several issues with this dataset. Firstly it is heavily imbalanced towards COPD class, and there are very few instances of asthma and LRTI; it also uses different types of equipment to record lung sounds, lacking consistency. With a focus on COPD and asthma diseases, this work uses a subset of this data, using only samples from healthy, asthma, and COPD classes that were recorded using stethoscopes. Secondly, asthma samples from [9] are used to increase the size of this class. We combine these data with our samples collected from healthy and asthmatic subjects. Recordings were obtained using digital stethoscopes from chest and back auscultation points. The combined dataset results in 147, 123, 104 samples for COPD, asthma, and healthy classes respectively (total 374). We employ a 70:30 split for training and testing sets respectively, reserving 10% for validation.

3.2 Feature Extraction

Librosa library was used to generate melspectrograms from time-series lung sound recordings as feature representation. We use an FFT window length of 2048, and a hop length between successive frames of 512. Figure 1 shows a sample melspectrogram for each class.

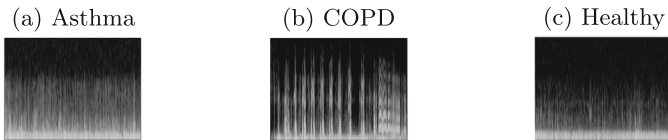


Fig. 1. Samples of Asthma, COPD, Healthy Lung Sounds

3.3 Predictive Models

We offer a comparison of a custom CNN architecture and various pre-trained predictive models for lung disease classification. The CNN model contained four convolutional layers, each followed by max pooling and a layer of dropout, resulting in a comparatively small network of 43,827 parameters. Then, we employed

various previously SOTA pre-trained networks for image classification, namely Xception [6] and ConvNeXtBase [13], which has shown competitive performance with transformers in object detection and segmentation tasks. These models are then compared with transformer networks, namely Vision transformer [24] and Swin [12]. In all training, Adam optimiser was used with weight decay of 0.0, and early stopping was configured to cease training when validation loss had not improved for eight epochs, using a batch size of 32.

3.4 Evaluation Metrics

Given the intended application domain of disease diagnosis, we use numerous evaluation metrics to provide a thorough picture of model performance. Loss and accuracy are used, alongside sensitivity/recall and precision, specificity, and F1 score.

4 Results and Discussion

The results for initial experiments are shown in Table 1. Using the training configurations described in Sect. 3.3, the results show that our custom built CNN architecture is inferior across all metrics for this task. It has a very modest ability to identify positive cases of disease and healthy samples. It showed the largest loss at 0.81 and a low 0.67 F1 score. Xception performed markedly better, reducing loss to 0.54, and significantly improving in accuracy, sensitivity, precision, specificity, and F1 score. The performance demonstrated by Xception is on a par with that of Swin transformer, achieving the same accuracy, sensitivity, and specificity of 0.77, 0.78, and 0.89 respectively, although this network reduced loss further to 0.52. The vision transformer model achieves the lowest loss value for this task at 0.44, however ConvNeXtBase (CvN) outperforms other architectures significantly with an F1 score of 0.87, meaning it is the most suitable network for identifying positive cases of asthma and COPD, and accurately predicting those without disease.

Table 1. Results of all models: Accuracy (A), Loss (L), Sensitivity (Se), Precision (Pr), Specificity (Sp), F1 score (F1) expressed in decimals

	A	L	Se	Pr	Sp	F1
CNN	0.67	0.81	0.68	0.68	0.83	0.67
Xcep	0.77	0.54	0.78	0.78	0.89	0.78
CvN	0.86	0.47	0.86	0.87	0.93	0.87
ViT	0.84	0.44	0.84	0.84	0.92	0.84
Swin	0.77	0.52	0.78	0.80	0.89	0.77

5 Conclusion and Further Work

This work offers a comparative performance in lung disease classification with a custom built CNN and various SOTA image classification networks, including Xception, ConvNeXtBase, and vision and swin transformer networks using novel data combined with limited publicly available samples. We have demonstrated superior performance for this task with ConvNeXtBase, where an F1 score of 0.87 is obtained. We recognise a limitation of this work to be the relatively small dataset of 374 samples. However, the data is well distributed between classes; this tackles a notorious problem in the field, in that lung sounds from asthmatics are not readily available compared to recordings from people with COPD. Secondly, this research fails to address data variety. Future efforts will look to improve variety by exploring methods such as audio augmentation, namely frequency and time masking in attempt to improve the robustness of the classifier, particularly for classes with fewer samples. Finally, ablation studies will be carried out to further understand the performance of the classifier when components of the network are removed.

References

1. Åström, C., Orru, H., Rocklöv, J., Strandberg, G., Ebi, K.L., Forsberg, B.: Heat-related respiratory hospital admissions in Europe in a changing climate: a health impact assessment. *BMJ Open* **3**(1), e001842 (2013)
2. Basu, V., Rana, S.: Respiratory diseases recognition through respiratory sound with the help of deep neural network. In: 2020 4th International Conference on Computational Intelligence and Networks (CINE), pp. 1–6 (2020)
3. Chambres, G., Hanna, P., Desainte-Catherine, M.: Automatic detection of patient with respiratory diseases using lung sound analysis. In: 2018 International Conference on Content-Based Multimedia Indexing (CBMI), pp. 1–6 (2018)
4. Chanane, H., Bahoura, M.: Convolutional neural network-based model for lung sounds classification. In: 2021 IEEE International Midwest Symposium on Circuits and Systems (MWSCAS), pp. 555–558 (2021)
5. Chang, Y., Ren, Z., Schuller, B.W.: Transformer-based CNNs: mining temporal context information for multi-sound COVID-19 diagnosis. In: 2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), pp. 2335–2338 (2021)
6. Chollet, F.: Xception: deep learning with depthwise separable convolution, pp. 1251–1258 (2017). [arXiv: arXiv:1610.02357](https://arxiv.org/abs/1610.02357)
7. Dosovitskiy, A., et al.: An image is worth 16 × 16 words: transformers for image recognition at scale. arXiv preprint [arXiv:2010.11929](https://arxiv.org/abs/2010.11929) (2020)
8. Fraiwan, M., Fraiwan, L., Alkhodari, M., Hassanin, O.: Recognition of pulmonary diseases from lung sounds using convolutional neural networks and long short-term memory. *J. Ambient Intell. Humaniz. Comput.* 1–13 (2022)
9. Fraiwan, M., Fraiwan, L., Khassawneh, B., Ibrnian, A.: A dataset of lung sounds recorded from the chest wall using an electronic stethoscope. *Data Brief* **35**, 106913 (2021)

10. Gairola, S., Tom, F., Kwatra, N., Jain, M.: Respirenet: a deep neural network for accurately detecting abnormal lung sounds in limited data setting. In: 2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), pp. 527–530 (2021)
11. Habashy, K., et al.: Cough classification using audio spectrogram transformer. In: 2022 IEEE Sensors Applications Symposium (SAS), pp. 1–6 (2022)
12. Liu, Z., et al.: Swin transformer: hierarchical vision transformer using shifted windows. *CoRR* **abs/2103.14030** (2021), <https://arxiv.org/abs/2103.14030>
13. Liu, Z., Mao, H., Wu, C.Y., Feichtenhofer, C., Darrell, T., Xie, S.: A convnet for the 2020s. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 11966–11976 (2022)
14. Müller, V., et al.: Characteristics of reversible and nonreversible copd and asthma and copd overlap syndrome patients: an analysis of salbutamol easyhaler data. *Int. J. Chronic Obstr. Pulm. Dis.* 93–101 (2016)
15. Nguyen, T., Pernkopf, F.: Lung sound classification using snapshot ensemble of convolutional neural networks. In: 2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), pp. 760–763 (2020)
16. Nguyen, T., Pernkopf, F.: Lung sound classification using co-tuning and stochastic normalization. *IEEE Trans. Biomed. Eng.* **69**(9), 2872–2882 (2022)
17. Perna, D., Tagarelli, A.: Deep auscultation: predicting respiratory anomalies and diseases via recurrent neural networks. In: 2019 IEEE 32nd International Symposium on Computer-Based Medical Systems (CBMS), pp. 50–55 (2019)
18. Pham, L., McLoughlin, I., Phan, H., Tran, M., Nguyen, T., Palaniappan, R.: Robust deep learning framework for predicting respiratory anomalies and diseases. In: 2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), pp. 164–167 (2020)
19. Pham, L., Phan, H., Palaniappan, R., Mertins, A., McLoughlin, I.: CNN-MoE based framework for classification of respiratory anomalies and lung disease detection. *IEEE J. Biomed. Health Inform.* **25**(8), 2938–2947 (2021)
20. Pham, L., Phan, H., Schindler, A., King, R., Mertins, A., McLoughlin, I.: Inception-based network and multi-spectrogram ensemble applied to predict respiratory anomalies and lung diseases. In: 2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), pp. 253–256 (2021)
21. Rocha, B.M., et al.: An open access database for the evaluation of respiratory sound classification algorithms. *Physiol. Meas.* **40**(3), 035001 (2019)
22. Shi, L., Zhang, J., Yang, B., Gao, Y.: Lung sound recognition method based on multi-resolution interleaved net and time-frequency feature enhancement. *IEEE J. Biomed. Health Inform.* **27**(10), 4768–4779 (2023)
23. WHO: The top 10 causes of death (2020). <https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death>. Accessed 04 Mar 2024
24. Wu, B., et al.: Visual transformers: token-based image representation and processing for computer vision (2020)



Evaluating the Performance of LLMs When Translating Saudi Arabic as Low Resource Language

Salwa Alahmari^{1,2} , Eric Atwell¹ , Mohammad Alsalka¹ ,
and Hadeel Saadany³ 

¹ University of Leeds, Leeds, UK

{scssala, e.s.atwell, m.a.alsalka}@leeds.ac.uk

² University of Hafr Albatin, Hafr Albatin, Saudi Arabia

ssalahmari@uhb.edu.sa

³ Birmingham City University, Birmingham, UK

hadeel.saadany@bcu.ac.uk

Abstract. This paper evaluates the performance of different large language models (LLMs) in translating textual data from Saudi Arabic, a low-resource language, into English. In this investigation we employ the state-of-the-art language models namely; ChatGPT-4, Claude-3 and Palm-2. We assess the capabilities of these LLMs on the Arabic Semantic Textual Similarity (STS) dataset. The evaluation covers different aspects, including the standard evaluation metrics, prompt design, and comparison with baselines systems namely; Google Translator, QuillBot Translator and Systran Translator. We conducted human evaluation on the generated translation and analysis the most frequent translation error using our sample dataset and different models. Our findings reveal significant insights into the strengths of ChatGPT (GPT-4) model in handling and translating dialectal Arabic with the highest Bilingual Evaluation Understudy (BLEU) score among all participated models (46.56).

Keywords: ChatGPT · Claude · Palm · Large Language Models · Evaluation of AI Systems · Machine Translation · Saudi Arabic Dialect

1 Introduction

Arabic presents unique challenges in the field of Machine Translation (MT) due to its status as a low-resource language [3]. Unlike languages such as English, French, or Spanish, Arabic lacks the extensive digital resources and large-scale parallel corpora necessary for training effective machine translation models. This scarcity of high-quality data significantly impacts the performance of machine translation systems for Arabic, leading to inaccuracies, inconsistencies, and limited coverage in translated texts. The limited availability of standardized and

annotated corpora further compounds these challenges, hindering the development of robust machine translation systems for Arabic. Addressing the low-resource nature of Arabic in the context of machine translation requires concerted efforts to collect, curate, and annotate large-scale datasets, as well as the development of innovative techniques tailored to the unique characteristics of the Arabic language.

Recent strides in the realm of natural language processing (NLP) have ushered in a new era, marked notably by the emergence of large language models (LLMs) [7,9]. These groundbreaking models show their efficiency and strength in solving challenges related to various NLP tasks including machine translation [4].

However, to the best of our knowledge, no study has evaluated the performance of LLMs when translating Saudi Arabic to English. Importantly, this study represents the initial effort to assess the effectiveness of Claude AI in Arabic machine translation in general and Saudi Arabic in particular. The main objective of this work is to evaluate three state-of-the-art LLMs, namely Generative Pretrained Transformer (GPT-4) through ChatGPT by OpenAI, the Pathways Language Model (PaLM) using Vertex AI¹ by Google and Claude AI for machine translation task of Saudi Arabic dialect. For this task we used small sub-set of the Arabic (STS) Corpus created by [2].

2 Related Work

The majority of studies concerning machine translation using LLMs have primarily concentrated on English datasets, with very limited research conducted in the for Arabic language. Consequently, our research aims to fill this gap in machine translation studies for Arabic overall, and specifically for Saudi Arabic.

[10] demonstrate that ChatGPT can surpass Google Translate on many translation pairs. On another hand, [14] show that ChatGPT outperformed by No Language Left Behind (NLLB) [12] on high percentage. In document-level translation [13] prove that ChatGPT can match the performance of fully supervised models. Moreover, [6] found that Claude-3 outperforms NLLB-54B and Google Translate on a significant number of language pairs in the FLORES-200 benchmark. Additionally, Claude demonstrates resource efficiency comparable to NLLB-54B, offering promising prospects for cost-effective machine translation models.

For Arabic language using LLMs, [8] present a comprehensive evaluation of machine translation performance of ChatGPT and Bard AI² on ten Arabic Varieties.

3 Methodology

In this study, we conducted an evaluation of three LLMs - ChatGPT, Claude 3, and PaLM 2 - to assess their effectiveness in translating Saudi Arabic into

¹ <https://cloud.google.com/vertex-ai>.

² <https://gemini.google.com/app>.

English. The goal of the methodology was to provide a structured and consistent approach for comparing the translation capabilities of these models using a standardized dataset. We employed a set of 200 sentences from the Arabic STS dataset, representative of dialectal Saudi Arabic. We do not apply any pre-processing or post-processing on the selected sentences. Each model was tasked with translating the same set of sentences, allowing for a direct comparison of their performance. During the translation process, we used the console interface for all the LLMs. Both standard evaluation metrics and human evaluation were applied to measure the quality of the translations and to identify the specific strengths and limitations of each model in translating textual data from Saudi Arabic dialects to English. We used the console interface for all the LLMs to ensure consistency in inputs and outputs across the models during the evaluation process.

Additionally, we selected three commercial translation systems as baselines: Google Translate³, Systran⁴, and QuillBot⁵. We then compared the performance of the LLMs with these three commercial translation systems using the same dataset and evaluation metrics.

3.1 Datasets Description

The dataset utilized in this research originates from a subset of the Arabic STS dataset. Initially introduced by [5] in the Shared Task: Semantic Textual Similarity (*SEM 2013), the Arabic STS dataset was expanded by [2] to include an additional 250 pairs of sentences in Modern Standard Arabic (MSA). Furthermore, translations of 1379 sentence pairs from the English STS dataset, initially compiled by [1], were incorporated into the Arabic STS dataset, encompassing translations into Modern Standard Arabic, Egyptian Arabic, and Saudi Arabic.

We have meticulously curated a subset comprising 200 pairs of sentences in Saudi Arabic and English for the purpose of evaluating three selected LLMs. This subset was intentionally selected to encapsulate the most representative vocabulary from the Saudi Arabic dialect.

Examples of these words include: **تكد**، **بطاطسة**، **يخم** and **جذع**

which mean in English “brushing”, “potato”, “hugging” and “boy” respectively. Table 1 shows examples from our sample dataset.

3.2 Prompt Design

Based on the results of MT using Language LLMs obtained by [8], among the three prompts used, the most effective design was the concise English prompt. Therefore, in this study, we adopted the same concise English prompt for all three LLMs utilized. The prompt used was: “As a Professional Translator, Can you please translate this sentence from Saudi Arabic to English?”.

³ <https://translate.google.com>.

⁴ <https://www.systransoft.com>.

⁵ <https://quillbot.com/translate>.

Table 1. Examples of Saudi Arabic and English pairs in our sample dataset.

Saudi Arabic	English
بنت تكّد شعرها	A girl is brushing her hair.
حرمة تقطع لها بطاطسة	A woman is chopping a potato.
رجال يحم له حرمة ويحبها	A man is hugging and kissing a woman.
ولد جذع يدق على آلة موسيقية	A boy is playing an instrument.

3.3 Evaluation Metrics

Our evaluation framework involves a comprehensive assessment of translation quality using standard metrics. In this study we use the following metrics: BLEU, Translation Edit Rate (TER) and Character n-gram F-score (ChrF). We refrain from employing model-based automatic evaluation metrics, like the Cross-lingual Optimized Metric (COMET) for Evaluation of Translation [11], for two reasons. First: the default COMET model⁶ trained expensively in Arabic MSA as a result, the model may fail to capture dialect-level nuances in the source text when computing the scores. Second: as Saudi Arabic consider low-resource language, precise evaluating is essential to our approach.

4 Results and Discussion

The results in Table 2 of our experiments reveal that the translation quality of Google translator is the highest among other commercial MT systems. Followed by QuillBot translator with BLEU scores 10.42 and 9.34 respectively.

Table 2. MT Evaluation results using commercial translation systems.

MT System	BLEU	TER	CHRF
Google translator	10.42	79.2	34.24
QuillBot	9.34	0.14	29.41
Systran	7.82	77.33	29.4

From the results in Table 3, we can see that GPT-4 and Claude-3 both demonstrate strong capabilities in translating Saudi Arabic text to English, with only slight differences in BLEU scores. Claude-3 follows closely behind GPT-4. The reason for this is that both GPT-4 and Claude-3 were trained on large datasets that included dialectal Arabic. Although PaLM-2 shows lower performance in the translation of dialectal Arabic in terms of BLEU score, it still performs much better than any commercial translation system.

⁶ <https://huggingface.co/Unbabel/wmt22-comet-da>.

Table 3. MT Evaluation results using LLMs.

LLM Name	BLEU	TER	CHRF
GPT-4	46.56	36.59	62.87
Claude-3	46.4	37.76	60.94
PaLM-2	18.59	61.3	47.87

4.1 Error Analysis

An in-depth error analysis was conducted to identify common translation issues across all models. We categorized errors into several types, including lexical errors, grammatical errors, and cultural/contextual misunderstandings. Our analysis revealed that while LLMs generally handled syntax well, they often struggled with idiomatic expressions and cultural references unique to Saudi Arabic. Commercial systems, on the other hand, frequently produced literal translations that missed the underlying meaning of the proverbs. For example, one of the most frequent error in the human evaluation of the translation quality is the translation of the word **حرمه** in Saudi Arabic which means “woman” or “lady” in English translated mistakenly into “hurmah” which considered as proper noun instead of adjective. Another frequent error is the translation of the word **بزر** which means “little boy” in English mistranslated as “seed” which means **بذرة** in Arabic.

5 Conclusion and Future Work

We evaluate ChatGPT, Palm-2 and Claude-3 on MT of Saudi Arabic textual data into English. The evaluation involved comparison between the performance of the three LLMs and to three commercial systems. Overall, the performance of the LLMs outperforms those of commercial MT systems. In addition, ChatGPT surpassed other models with a slightly difference in BLEU score with Claude-3. Our study highlights the potential and limitations of current LLMs in translating low-resource dialects like Saudi Arabic. While these models show promise, there is still room for improvement, particularly in handling cultural nuances and rare dialectal expressions. Future work will focus on enhancing model training with more diverse and representative datasets, exploring advanced techniques for better prompt design and fine-tuning.

References

1. Agirre, E., Cer, D., Diab, M., Gonzalez-Agirre, A., Guo, W.: *SEM 2013 shared task: semantic textual similarity. In: Diab, M., Baldwin, T., Baroni, M. (eds.) Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity, pp. 32–43. Association for Computational Linguistics, Atlanta, Georgia, USA (June 2013)

2. Al Sulaiman, M., Moussa, A.M., Abdou, S., Elgibreen, H., Faisal, M., Rashwan, M.: Semantic textual similarity for modern standard and dialectal Arabic using transfer learning. *PLOS ONE* **17**(8), 1–14 (2022)
3. Almansor, E.H., Al-Ani, A., Hussain, F.K.: Transferring informal text in Arabic as low resource languages: state-of-the-art and future research directions. In: Barolli, L., Hussain, F.K., Ikeda, M. (eds.) *CISIS 2019. AISC*, vol. 993, pp. 176–187. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-22354-0_17
4. Brown, T.B., et al.: Language models are few-shot learners (2020)
5. Cer, D., Diab, M., Agirre, E., Lopez-Gazpio, I., Specia, L.: Semeval-2017 task 1: semantic textual similarity multilingual and crosslingual focused evaluation. In: *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Association for Computational Linguistics (2017)
6. Enis, M., Hopkins, M.: From llm to nmt: advancing low-resource machine translation with claude (2024)
7. Hoffmann, J., et al.: Training compute-optimal large language models (2022)
8. Kadaoui, K., et al.: Tarjamat: evaluation of bard and chatgpt on machine translation of ten Arabic varieties (2023)
9. Kaplan, J., et al.: Scaling laws for neural language models (2020)
10. Peng, K., et al.: Towards making the most of ChatGPT for machine translation. In: Bouamor, H., Pino, J., Bali, K. (eds.) *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 5622–5633. Association for Computational Linguistics, Singapore (Dec 2023)
11. Rei, R., Stewart, C., Farinha, A.C., Lavie, A.: COMET: a neural framework for MT evaluation. In: Webber, B., Cohn, T., He, Y., Liu, Y. (eds.) *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 2685–2702. Association for Computational Linguistics, Online (Nov 2020)
12. Team, N., et al.: No language left behind: scaling human-centered machine translation (2022)
13. Wang, Z., Xie, Q., Feng, Y., Ding, Z., Yang, Z., Xia, R.: Is chatgpt a good sentiment analyzer? A preliminary study (2024)
14. Zhu, W., et al.: Multilingual machine translation with large language models: empirical results and analysis (2023)



Bi-directional LSTM Applied to the Maritime Target Motion Analysis Problem

Lars Nolle^{1,2}(✉), Nils Meinardus¹, Martin Kumm¹, and Christoph Tholen²

¹ Department of Engineering Sciences, Jade University of Applied Sciences,
Friedrich-Paffrath-Str. 101, 26384 Wilhelmshaven, Germany
{lars.nolle,martin.kumm}@jade-hs.de, lars.nolle@dfki.de

² German Research Center for Artificial Intelligence GmbH, RG Marine Perception,
Marie-Curie-Str. 1, 26129 Oldenburg, Germany
christoph.tholen@dfki.de

Abstract. In this work, a bi-directional LSTM has been employed to predict the future positions of a maritime vessel, known as the target, based on noisy estimates of its previous and current positions. In a first set of experiments, plain trajectories generated by a simulation were used for training and noisy trajectories were used for testing. In a second set of experiments, noisy trajectories were used for training and testing. The accuracy and the loss achieved in both sets of experiments have demonstrated that this type of network is capable of solving the Target Motion Analysis problem.

Keywords: Bi-directional LSTM · Target Motion Analysis

1 Introduction

The aim of Target Motion Analysis (TMA) is to estimate the current state, i.e. location, bearing, and velocity of an object, known as the target, in order to predict its position at a later point in time. This is of particular interest to military users, who, in the maritime context, want to predict the future positions of unknown objects, such as submerged submarines, which in turn try to avoid observations at all costs. In such a scenario, the observer platform, also known as ownship, uses hydrophones, which are mounted at some distance D on a cable towed by the observer, to detect signals that are emitted from the target [1]. Here, the time of emission is not known. The range R and the bearing θ of the target can be determined by measuring differences in arrival time of short-duration signals along the paths R_1 , R_2 and R_3 , as shown in Fig. 1. In the maritime domain, the time delay measurements are disturbed by noise, caused, for example, by the environment or the cross-correlation function used for finding a common signal in a pair of sensors [2]. Some errors might be caused by false readings or clutter, which is assumed to be uniformly distributed over an area A . Location estimates are collected over time and are subsequently used for trajectory prediction, for example by applying M-Estimators [3], Kalman filters [4, 5] or Particle filters [6]. In previous work, artificial intelligence methods, like LSTMs [7], Ant Colony Optimisation [8], or feed-forward artificial neural

networks [9] were successfully applied to the TMA problem, achieving better accuracy than traditional methods [7].

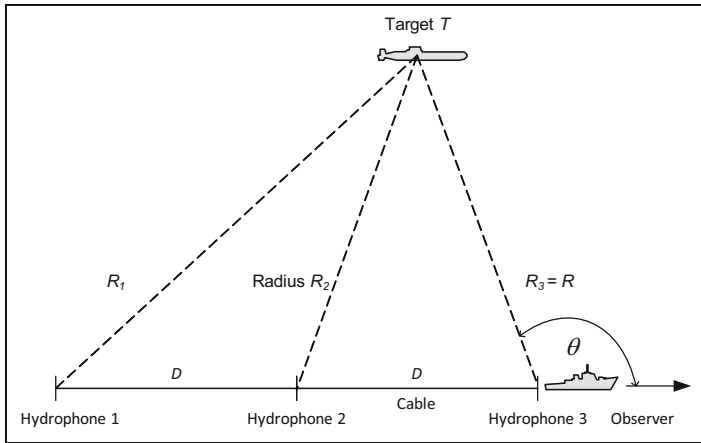


Fig. 1. Target Motion Analysis problem.

Zhang et al. [10] used a bi-directional LSTM for trajectory prediction using data provided by the Automatic Identification System (AIS). However, this data is actively generated by vessels using global navigation satellite systems (GNSS). Hence, it is not affected by noise and clutter as in the TMA scenario described above. In this work, a bi-directional LSTM is trained to predict trajectories from noisiness data, not only for straight trajectories but also for curved courses. For training and evaluation, a simulation, which is described below, was developed, that generated target trajectories in a marine environment.

2 Simulations

A simulation was developed, which generates target trajectories in relation to the ownships position. The parameters of the simulated targets are based on the limitations of real-world vessels.

The speed of the target was the main parameter used for track generation. The maximum speed is based on generally known speeds of ships, boats, or similar vessels. These can reach up to 25 knots. Hence, for each target a constant speed v was chosen randomly from the interval:

$$\{v \in \mathbb{R} | 5 \text{ kn} \leq v \leq 20 \text{ kn}\}. \quad (1)$$

Due to the problem at hand, it cannot be guaranteed that observations are equidistant in the time domain. Therefore, the time between two observations t was selected randomly from the interval:

$$\{t \in \mathbb{R} | 1 \text{ s} \leq t \leq 10 \text{ s}\}. \quad (2)$$

Due to physical limitations of passive sonar systems, measurements up to 6 km are assumed to be sufficiently accurate under most conditions [11]. The initial distance r of the targets is chosen randomly from the interval:

$$\{r \in \mathbb{R} | 0 \text{ m} \leq r \leq 6000 \text{ m}\}. \tag{3}$$

The initial position of the targets are determined by the radius r and the bearing φ . The bearing was also chosen randomly from the interval:

$$\{\varphi \in \mathbb{R} | -180^\circ \leq \varphi \leq 180^\circ\}. \tag{4}$$

The track data generated by the simulation was validated by human experts. Figure 2 shows a plot of 1,000 simulated target tracks. For training the artificial feed-forward network, 10,000 of such tracks were generated and used, as discussed in the next section.

In order to test the reliability of the network, the simulation offers the possibility of adding noise to the track data. In order to generate realistic error curves, the added noise is randomly chosen from a Gaussian distribution with mean of zero and a distance depending standard deviation. Figure 3 shows the trajectories from Fig. 2. Examples of generated track data. After noise added.

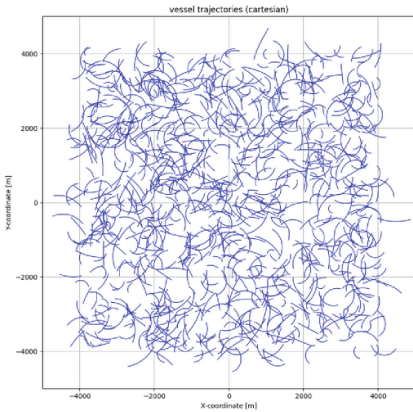


Fig. 2. Examples of generated track data.

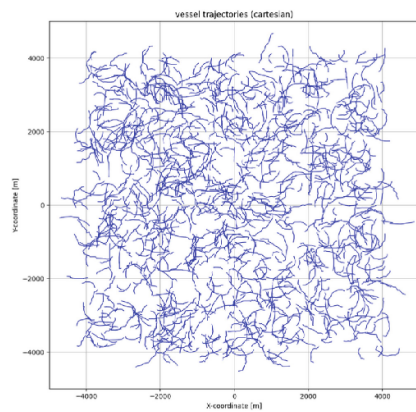


Fig. 3. Trajectories with added noise used for testing.

This data set was used for training and testing the effectiveness of the proposed method for noisy trajectory estimates.

3 Network Type and Architecture

In this work, a bi-directional LSTM was used for trajectory prediction. Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) designed to efficiently learn long-term dependencies in sequence data by using gates to control the flow of information in a feedforward manner [12], whereas a bi-directional LSTM processes

the input sequence forwards and backwards to capture information from past and future states in the data, i.e. training it simultaneously in positive and negative time direction [13]. The same bi-directional LSTM network was used as in [10], shown in Fig. 4. In order to determine the optimum number of nodes of the network, experiments have been carried out with 2, 4, 8, 16, and 32 nodes. Each experiment has been repeated 30 times and the average accuracy and the average loss have been calculated. The results can be found in Fig. 5.

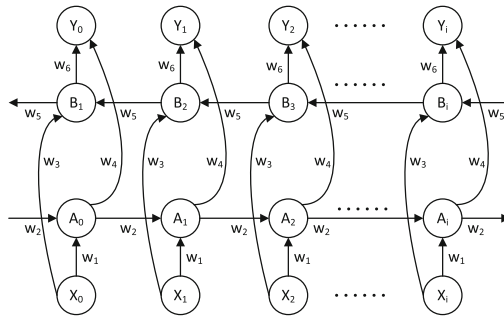


Fig. 4. Bi-directional LSTM network, adopted from [10].

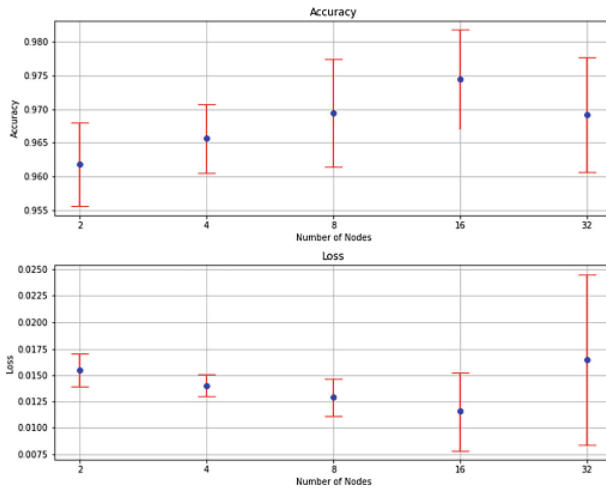


Fig. 5. Accuracy and loss for different number of nodes.

It can be seen from the graphs that the average accuracy peaked when 16 nodes were used. This corresponds to the average loss, which also has its minimum using 16 nodes. Therefore, this network topology was used in the subsequent experiments described in the next section.

4 Experimental Results and Discussion

Two sets of experiments were carried out. In the first set, the plain trajectories were used for training and the noisy trajectories were used for testing. In the second set, the noisy trajectories were used for training and testing. Each experiment was repeated 30 times in order to calculate the average accuracy and the average loss. The number of epochs was chosen to be 50 with early stopping detection. The results for the first set of experiments can be seen in Fig. 6. The results for the second set of experiments can be found in Fig. 7.

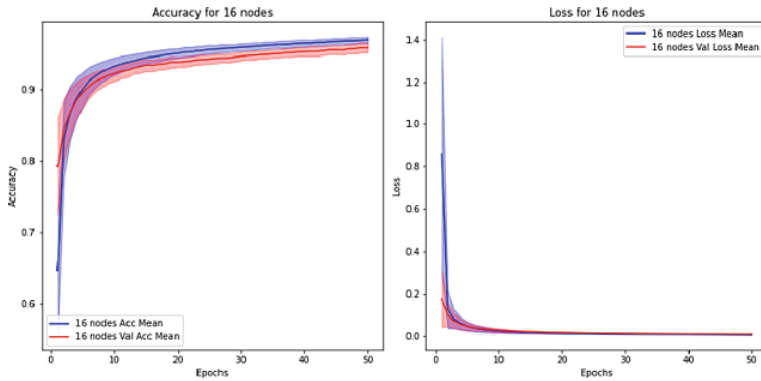


Fig. 6. Average accuracy and standard deviation (left) and average loss and standard deviation (right) over time, i.e. epochs, for the first set of experiments.

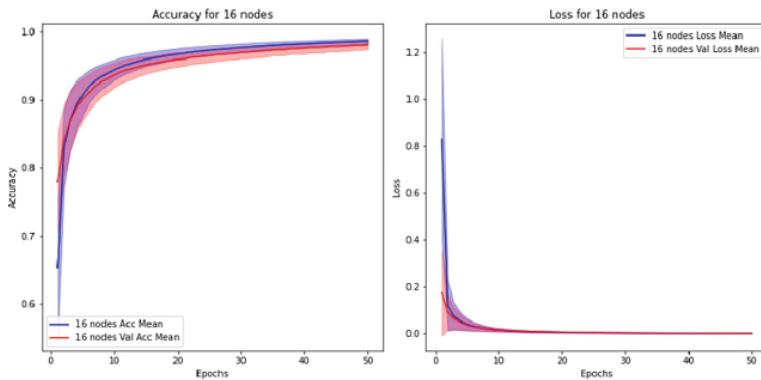


Fig. 7. Average accuracy and standard deviation (left) and average loss and standard deviation (right) over time, i.e. epochs, for the second set of experiments.

5 Conclusions and Future Work

It can be seen from the graphs in the section above that the average accuracy achieved in the first set of experiments was 0.968 and the average loss was 0.003. The standard deviation for the accuracy was 0.007 and the standard deviation for the loss was 0.003.

In the second set of experiments, an average accuracy achieved was 0.983 with a standard deviation of 0.005 and the average loss achieved was 0.003 with a standard deviation of 0.001. Hence, it was shown that that bi-directional LSTM was capable of approximating the trajectories of unknown targets by using noisy position estimates.

However, the experiments presented did not include clutter, i.e. observations that do not originate from the target. This will be included into future work.

References

1. Hassab, J.C., Guimond, B.W., Nardone, S.C.: Estimation of location and motion parameters of moving source observed from a linear array. *J. Acoust. Soc. Am.* **70**(4), 1054–1061 (1981)
2. Carevic, D.: Robust estimation techniques for target motion analysis using passively sensed transient signals. *IEEE J. Oceanic Eng.* **28**(2), 262–270 (2003)
3. Huber, P.J.: *Robust Statistics*. Wiley, Hoboken (1981)
4. Aidala, V.J.: Kalman filter behavior in bearings-only tracking applications. *IEEE Trans. Aerosp. Electron. Syst.* **1**, 29–39 (1979)
5. Babu, G., Jayaprakash, V., Mamatha, B., Annapurna, P.: A neural network target tracking using kalman filter. *Int. J. Eng. Res. Technol.* **1**(9) (2012)
6. Lin, X., Kirubarajan, T., Bar-Shalom, Y., Maskell, S.: Comparison of EKF, pseudomeasurement, and particle filters for a bearing-only target tracking problem. In: *Proceedings of the SPIE 4728, Signal and Data Processing of Small Targets* (2002)
7. Gao, C., Liu, H., Zhou, S., Su, H., Chen, B., Yan, J., Yin, K.: Maneuvering target tracking with recurrent neural networks for radar application. In: *2018 International Conference on Radar (RADAR)*, Brisbane, QLD, Australia, pp. 1–5 (2018)
8. Nolle, L.: On a novel ACO-estimator and its application to the target motion analysis problem. In: Ellis, R., Allen, T., Petridis, M. (eds.) *Applications and Innovations in Intelligent Systems XV. SGAI 2007*. Springer, London (2008). https://doi.org/10.1007/978-1-84800-086-5_1
9. Schlüsselburg, T., Tholen, C., Nolle, L.: On the application of feed-forward artificial neural networks to the maritime target motion analysis problem. In: Bramer, M., Stahl, F. (eds.) *Artificial Intelligence XL. SGAI 2023. LNCS*, vol. 14381, pp. 481–486. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-47994-6_41
10. Zhang, S., Wang, L., Zhu, M., Chen, S., Zhang, H., Zeng, Z.: A Bi-directional LSTM Ship Trajectory Prediction Method based on Attention Mechanism. In: *2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, Chongqing, China, pp. 1987–1993 (2021)
11. Han, J., Zhang, X.G., Meng, C.X., Cao, F.: Simulated Research on passive sonar range using different hydrographic conditions. In: *MATEC Web of Conferences*, vol. 35, p. 04003 (2015)
12. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural Comput.* **9**(8), 1735–1780 (1997)
13. Schuster, M., Paliwal, K.K.: Bidirectional recurrent neural networks. *IEEE Trans. Signal Process.* **45**(11), 2673–2681 (1997)

Author Index

A

Abbas, Noorhan I-301, I-307, I-313, I-333,
II-105

Abdelmoty, Alia I. I-285

Agarwal, Ritika II-105

Agater, Jérôme II-208

Ainslie, Russell II-179

Alahmari, Salwa II-264

Alsalka, Mohammad II-264

Anders, Royce II-232

Andersen, Per-Arne I-3, I-116, I-175, II-251

Ani, Uchenna Daniel II-225

Asowo, Patricia II-225

Atwell, Eric I-307, II-264

B

Bader-El-Den, Mohamed II-3

Banda, Tiwonge Msulira I-268

Bando, Sio II-232

Barndon, Halvor Helland II-147

Basson, Anton Herman II-75

Bergmann, Ralph I-189

Biermann, Daniel I-61

Bin Shiha, Rawan I-307

Blanc, Nathalie II-232

Borgersen, Karl Audun I-3

Bouomar, Alaa I-333

Brådland, Henrik I-175

Brigaud, Emmanuelle II-232

Brownlee, Alexander E. I. II-179

Bundy, Alan II-194

Burrows, Holly II-258

C

Cairns, David II-179

Caldwell, Nicholas H. M. II-119

Catalano, GianCarlo A. P. I. II-179

Caulfield, Brian II-88

Cernekova, Zuzana I-158

Chakraborti, Sutanu I-235

Chan, Pak Yin II-194

Chen, Ren II-238

Chen, Zhen II-238

Croituru, Madalina II-232

D

Day, Matthew I-346

Debich, Tarek II-59

Dickson, Nathan R. II-119

Dippel, Oliver I-207

Dominguez, Alejandro Rodriguez I-88

E

El-Mihoub, Tarek A. I-144

Elo Dean, Sara II-147

Elsayed, Ahmed H. I-144

F

Feely, Ciara II-88

Frederick-Preece, Nicholas Arthur I-301

Fyvie, Martin II-179

G

Ganesh, Gowrishankar II-232

Garbagna, Lorenzo I-327

Gilles, Eric II-232

Giske, Lars Adrian II-147

Goodwin, Morten I-3, I-61, I-116, I-175,
II-21

Granmo, Ole-Christoffer I-61

Grobler, Jacomine II-75

Grundetjern, Morten I-3

H

Helian, Na II-46

Hesselmann, Fenja T. II-244

Holen, Martin II-21

Hong, Xia I-88

Hopgood, Adrian A. II-3

Huyck, Christian I-33

J

- Jacobson, Lars E. O. II-3
 Jafari, Fatana I-158
 Jiao, Lei II-251
 Jyhne, Sander Riisøen I-116

K

- Kapetanakis, Stylianos I-105
 Knausgård, Kristian M. II-21
 Kumm, Martin II-270

L

- Lal, Sangeeta II-225
 Laviron, Pablo II-232
 Lawlor, Aonghus II-88
 Ławryńczuk, Maciej I-74
 Lenz, Mirko I-189
 Li, Xue II-194
 Liao, Yong II-238
 Lin, Yuhui II-194
 Lisitsa, Alexei I-207
 Liu, Xi II-238
 Lopedoto, Enrico I-130
 Lu, Yiwei II-194
 Luhmann, Thomas I-352

M

- Maharjan, Reshma II-251
 Manss, Christoph I-144
 Martens, Tyrell II-133
 Masum, Shamsul II-3
 McCall, John A. W. II-179
 Meinardus, Nils II-270
 Melethil, Praseed I-327
 Memari, Ammar II-59, II-208
 Miedtank, Andre I-144
 Miled, Amine II-232
 Minku, Leandro L. I-253
 Mostein, Herman Jangsett II-147

N

- Neiss-Theuerkauff, Tobias I-352
 Nolle, Lars I-47, I-144, II-270
 Nordberg, Henrik II-147
 Nossun, Alexander S. I-175

O

- Oghaz, Mahdi Maktabdar I-320, I-327,
 II-35, II-258

- Omlin, Christian I-221, II-21
 Osborn, Peter II-3
 Oveland, Ivar I-116

P

- P., Deepak I-235
 Palumbo, Fabrizio I-61
 Parsodkar, Adwait P. I-235
 Pasipamire, Kudiwa II-46
 Pasipamire, Tony II-46
 Peng, Bei I-207
 Podestà, Silvia II-147
 Prendergast, David II-3

R

- Rettig, Robert I-47
 Rimal, Kshitiz I-313
 Rodenbäck, Eike I-47

S

- Saadany, Hadeel II-264
 Saheer, Lakshmi Babu I-320, I-327, II-35,
 II-258
 Salako, Kizito I-130
 Satoti, Abdurauf I-285
 Schierbaum, Arne I-352
 Shahzad, Muhammad I-88
 Sheni, Dauda Nanman II-75
 Shi, Xinming I-253
 Sieberth, Till I-352
 Singh, Jayant II-21
 Sisik, Viktor I-158
 Smítková Janků, Ladislava I-19
 Smyth, Barry II-88
 Stahl, Frederic I-47, I-144
 Stegane, Sara II-147
 Sun, Yi II-46
 Syed, Hasnain Murtaza II-35

T

- Tamma, Vincenzo II-3
 Tang, Wei Liang Russell II-162
 Tholen, Christoph I-47, II-270
 Toh, Say Meng II-46
 Tolegenov, Mukhambet I-320

V

- Vishwanath, Ajay I-221

WWallhoff, Frank [I-352](#), [II-244](#)Weyde, Tillman [I-130](#)Wheatman, M. J. [I-339](#)**Y**Yang, Shufan [II-238](#)Yao, Xin [I-253](#)**Z**Zafar, Huma [I-105](#)Zarzycki, Krzysztof [I-74](#)Zăvoianu, Alexandru-Ciprian [I-268](#)Zeman, Jakub [I-19](#)Zhang, John Z. [II-133](#)Zheng, Changgang [II-238](#)Zhou, Jing [II-21](#)