

# On the role of feature and signal selection for terrain learning in planetary exploration robots

Angelo Ugenti<sup>1</sup>, Fabio Vulpi<sup>1,2</sup>, Raúl Dominguez<sup>3</sup>, Florian Cordes<sup>3</sup>, Annalisa Milella<sup>2</sup>, Giulio Reina<sup>1</sup>

<sup>1</sup>*Department of Mechanics, Mathematics and Management, Polytechnic of Bari, Via Orabona 4, 70125, Bari, Italy {angelo.ugenti, giulio.reina}@poliba.it*

<sup>2</sup>*Institute of Intelligent Industrial Technologies and Systems for Advanced Manufacturing, National Research Council, via G. Amendola 122/O, 70126 Bari, Italy, {fabio.vulpi, annalisa.milella}@stiima.cnr.it*

<sup>3</sup>*DFKI RIC, Robert-Hooke-Str. 1, Bremen, 28359 Germany, {raul.dominguez, florian.cordes}@dfki.de*

## Abstract

Increasing the terrain awareness of planetary exploration rovers is one key technology for future space robotics to successfully accomplish long-distance and long-duration missions. In contrast to most of the existing algorithms that use visual or depth data for terrain classification, the approach presented in this work tackles the problem using proprioceptive sensing, e.g., vibration or force measurements. The underlying assumption is that these signals, being directly modulated by the terrain properties, are well descriptive of a given surface. Therefore, terrain signature can be inferred via learning algorithms that are trained on either the signals directly or a signal-derived feature set.

Following the latter approach, first, a physics-based signal augmentation process is presented that aims at maximizing the information content. Then, a feature selection algorithm based on a scoring system and an iterative search is developed to decrease the computational cost while preserving high classification accuracy. The resulting most informative feature subspace can be used to train a Support Vector Machine (SVM) classifier. For comparison, the time histories of the selected proprioceptive signals are used to train a deep Convolutional Neural Network (CNN).

Results obtained from real experiments using the SherpaTT rover confirm that proprioceptive sensing is effective in predicting terrain type with an accuracy higher than 90% for both algorithms in generalization tasks. When the two learning approaches are contrasted in extrapolation problems, e.g. predicting observations acquired at previously unseen velocity or terrain, CNN outperforms the standard SVM. Furthermore, CNN holds the additional advantage of learning features automatically from signal spectrograms, reducing the need of a-priori knowledge at the expense of higher computational efforts.

**Keywords:** Planetary exploration robots, vehicle-terrain mechanics, proprioceptive sensing, feature selection, learning methods, deep learning, terrain classification.

## I. Introduction

This work has been developed as part of the research activity for the project ADE (Autonomous DEcision making in very long traverses) (Ocón et al., 2020), funded by the European Union’s Horizon 2020 research and innovation programme. The main goal of ADE is to develop and test a rover system capable to achieve autonomous long-range navigation in hostile environments, while guaranteeing consistent data collection. The mobility range of planetary exploration rovers has been up to date limited to few hundreds of meters per sol day (*ESA, Robotic Exploration of Mars*, 2021; *JPL, Mars Exploration Rovers*, 2021; *Nasa, Mars 2020*, 2020). From a purely technical point of view, this limitation has both hardware and software sources. The former and most important is the finite power storage of rover locomotion system, that is fixed given a robot design. The latter is reduced skills in terms of autonomous decision-making, that can be improved by artificial intelligence. Improving

49 these capabilities extends the autonomy of the rover across multiple geographical areas and therefore  
50 expands opportunities of data collection.

51 Directly related to long-range navigation is also the safety issue. The importance of sensing hazards  
52 was highlighted, for example, in April 2005, when the Mars exploration rover Opportunity became  
53 embedded in a dune of loosely packed drift material (Cowen, 2005). The terrain geometry as  
54 reconstructed from a distance via stereovision did not indicate any hazard. However, the high  
55 compressibility of the loose drift material caused the wheels to sink deeply into the surface. The  
56 combination of the drift's low internal friction and the motion resistance due to sinkage prevented the  
57 rover from producing sufficient thrust to travel up the slope. Opportunity's progress was delayed for  
58 more than a month while engineers worked to find a way out. A similar embedding event led to the  
59 end of operations for the twin rover Spirit in 2010.

60 Therefore, future generations of planetary exploration rovers will require key technologies suitable  
61 to overcome these limitations, performing long traverses while guaranteeing fast reaction, mission  
62 reliability and safety, and optimal exploitation of the robot's resources within reasonable costs.

63 In this context, the ability to sense and characterize the incoming terrain would represent an enabling  
64 technology towards long-term autonomy and potential hazard avoidance (Nampoothiri et al., 2021).

65 The objective of this paper is to demonstrate the potential of terrain classification via learning  
66 algorithms that are trained on proprioceptive features. Here, proprioceptive features refer to statistics  
67 that are extracted from the measurement of a physical variable pertaining to the robot-environment  
68 interaction, e.g., wheel velocity, forces, body linear and angular accelerations.

69 The hypothesis is that, being modulated by the terrain properties, these features are a rich source of  
70 information from which the specific terrain type can be inferred via learning approaches (Brooks &  
71 Iagnemma, 2005; Gonzalez et al., 2019).

72  
73 One of the contributions of this research refers to the selection of the most informative subset of  
74 proprioceptive features derived from the sensor suite integrated onboard of planetary exploration  
75 rovers. A range of aspects is addressed that includes feature extraction, feature ranking, multivariate  
76 feature selection and efficient feature space construction. While feature selection has been largely  
77 investigated in other domains e.g., image processing, text processing and gene expression analysis  
78 (Guyon & Elisseeff, 2003), it remains largely under investigated for the terrain classification problem  
79 of planetary exploration rovers, and rough-terrain robots, in general. In contrast to other areas of  
80 applications where datasets with tens or hundreds of thousands of variables are available forming a  
81 statistically significant population, data acquired by a rover driving over natural terrain present many  
82 challenges such as sparseness, presence of unknown and uncontrolled disturbances, dependence on  
83 the specific time and site of the acquisition.

84 The objectives pursued by feature selection include improvement in the prediction performance,  
85 reduction in training time, computational burden and memory usage of the algorithm and facilitation  
86 of understanding the underlying process that generated the data.

87  
88 The other contribution of this research is the adoption of a suitable learning algorithm to infer the  
89 type of terrain from the selected feature set. This algorithm will have to look for patterns in the data  
90 to construct the mapping from the proprioceptive measurements to the corresponding terrain type.

91 The well-known Support Vector Machine (SVM) is contrasted with a deep convolutional neural  
92 network (CNN). While SVM requires in input hand-crafted features that are selected during a pre-  
93 processing stage, CNN uses learned features that are extracted automatically from the signal time  
94 histories.

95 An important goal of the proposed approach is to improve the performance of terrain classifiers for  
96 two use cases: generalization and extrapolation. Generalization is defined as the performance of an  
97 algorithm on previously unseen observations (test set) that is extracted from the same distribution as  
98 the data in the training set, e.g., the same test run. The error measured on the test set corresponds to  
99 both the on-line performance of the model and the operating conditions included in the training set.

100 The second use case, extrapolation, is even more challenging since, in general, learning algorithms  
101 are known to perform poorly outside the training data population. We compare the performance of  
102 the two terrain classifiers (SVM and CNN) for both generalization and extrapolation.

103

104 After related research is surveyed in Section II, highlighting the novel aspects of this paper, Section  
105 III presents SherpaTT, the rover used as test bed, and the learning algorithms implemented in this  
106 work. Next, signal augmentation, feature extraction and selection problems are tackled in Section IV.  
107 The results obtained from the terrain classifiers are presented and discussed in Section V. Conclusions  
108 wrap up the paper.

## 109 II. Related Work

110 Solving terrain-related challenges such as soil identification is an important research area in  
111 autonomous robots, alongside trajectory planning, localization and obstacle avoidance (Nampoothiri  
112 et al., 2021). The latest developments in terrain classification strategies show that researchers have  
113 been focusing on two main categories: visual (or exteroceptive) and visual-independent (or  
114 proprioceptive) methods. In both approaches, data collected from sensors are used to train machine  
115 or deep learning-based classifiers that enable identification of the traversed terrain.

116 The sensors used for visual perception include RGB cameras (Tai et al., 2017; Wellhausen et al.,  
117 2019), RGB-D cameras (Manduchi et al., 2005), LiDARs (Tai et al., 2017), visual cameras (Otsu et  
118 al., 2016) and monocular cameras (Barnes et al., 2017). Although visual-based approaches are more  
119 common than proprioceptive-based ones, they have limitations as well. The performance of RGB  
120 cameras is limited by difficult environmental conditions (e.g., low, or direct lighting and surface  
121 reflectivity). LiDARs struggle to capture the fine texture of objects and terrains, and they also perform  
122 poorly in compromised environment conditions (e.g., in presence of dust, hail and smog).  
123 Furthermore, vision-based rovers are not able to navigate in unfamiliar surroundings because  
124 observing distant terrain patches does not provide information about the mechanical properties that  
125 directly impact on vehicle mobility.

126 Therefore, researchers have investigated methods that use proprioceptive sensing for terrain  
127 classification. In this case, the sensors used to perceive the incoming terrain include IMU (Inertial  
128 Measurement Unit), force-torque sensors, microphones, and wheel encoders. As an example,  
129 Hishikawa et al. (Ishikawa et al., 2021) used microphones to support an RGB camera in dark  
130 conditions. Brooks and Iagnemma (Brooks & Iagnemma, 2005) measured vibrations via  
131 accelerometers, analyzed them in the frequency domain and implemented an online classifier that  
132 relies on Principal Component Analysis (PCA) for feature reduction. DuPont et al. (DuPont et al.,  
133 2008) presented a method based on frequency response and vibration-based transfer function. Giguere  
134 and Dudek (Giguere & Dudek, 2011) used a tactile probe combined with accelerometers to account  
135 for inertial effects. Dutta and Dasgupta (Dutta & Dasgupta, 2017) pursued a low cost approach using  
136 a multi-sensor platform fitted with GPS, IMU and metal detector. A model-based observer grounded  
137 in the Cubature Kalman filter was also proposed in (Reina et al., 2020) to predict terrain deformability  
138 using vertical acceleration measurements.

139 The above works based on visual-independent approaches represent a step forward in the direction  
140 of providing a mobile robot with information about the mechanical properties of the terrain. Although  
141 they achieved high confidence levels, little effort was spent on feature selection as a mean to reduce  
142 the computational burden of the model without penalties in performance. Ultimately, the objective of  
143 researchers that work on robot-terrain interaction is to develop an accurate algorithm that runs online  
144 while the robot is moving. This algorithm must comply with the limited resources of an autonomous  
145 vehicle in terms of processing power and memory. A reduction in the number of features used to train  
146 and test a machine learning classifier would lead to a lighter computational burden in terms of feature  
147 extraction time, testing time and memory usage. One of the contributions of this paper is to develop

148 a feature selection algorithm demonstrating that these benefits can be achieved without compromising  
149 the accuracy of the model.

150 A body of research has been devoted to the feature extraction process, as the quality of the feature  
151 space directly affects the accuracy of the associated classifier. The feature extraction strategy depends  
152 on the machine learning approach chosen for terrain classification. Traditionally, for a supervised  
153 machine learning algorithm such as SVM, an extraction stage is required where features are hand-  
154 crafted by experts based on their knowledge in the specific application domain. Attempts have been  
155 made in various research fields to find effective features, for example in image-processing-related  
156 applications (Lu & Weng, 2007). However, this approach is not always possible for classifiers and it  
157 is often practically difficult, for instance when the relationship between input measurements and user-  
158 defined classes is extremely complex or even completely unknown beforehand. Additionally, features  
159 that are crafted manually may be not optimal. For this reason, finding more systematic ways to get  
160 good features has drawn an increasing research interest (Bengio et al., 2013).

161 Notable progress has been done recently to find learning techniques that allow models to learn  
162 features automatically from data with minimal manual input. Solutions using Deep Neural Networks  
163 (NNs) have especially attracted much attention. The effectiveness of deep NNs has been  
164 demonstrated in many fields other than image classification, such as audio and natural language  
165 processing or transfer learning. The adoption of Recurrent and Convolutional Neural Network was  
166 discussed in (Vulpi et al., 2021), in the context of terrain classification using an agricultural robot  
167 equipped only with inertial and electrical current sensors. However, although the promising results,  
168 it remains challenging to evaluate the effectiveness of learned features contrasted with expert-  
169 designed ones. The complexity of this comparison resides in the difficulty of determining the  
170 descriptive power of hand-crafted features. For this reason, this paper presents a fair comparison  
171 between hand-crafted and learned features through a rigid feature scoring and selection process.

172 In previous work by the authors (Dimastrogiovanni et al., 2020), a preliminary attempt was presented  
173 to select a subset of optimal proprioceptive features to train an SVM-based ground classifier then  
174 tested over only two terrain types, e.g. rock and sand.

175 In this work, several novel additions are made. First, a whole new signal engineering stage is  
176 introduced to improve the overall information content. The signal selection strategy is formalized and  
177 reflected in an explanatory block diagram. Improved robustness has been achieved by increasing the  
178 number of training repetition for each candidate feature set. Then, the importance of feature selection  
179 for terrain classification is shown by comparing a machine learning approach (SVM) with a deep  
180 convolutional neural network (CNN) in terms of model complexity, computational burden, and  
181 prediction accuracy over a larger terrain set (3 types of terrain against 2 of the previous work).  
182 Finally, the system is evaluated not only in a standard generalization problem but as well as in two  
183 more challenging extrapolation contexts that are seldom described in the Literature.

### 184 III. Materials and Methods

185 The first part of this section (III.A) briefly presents the experimental planetary rover used for data  
186 gathering, describing the onboard sensor suite and the datasets collected during the field trials for  
187 developing terrain classification models. Then, the learning algorithms for terrain classification are  
188 presented, providing insights into the theoretical background.

#### 190 III.A. The rover SherpaTT

191 The experimental test bed used in the ADE project is the SherpaTT rover (see Figure 1) built by DFKI  
192 (Cordes et al., 2018). SherpaTT is a hybrid four-wheeled-leg rover, where the wheel-on-leg design  
193 constitutes an actively articulated suspension system. Flexible metal wheels provide a passive ground  
194 adaption on a small scale, while the active suspension fits the wheel positions to larger ground  
195 irregularities (Cordes & Babu, 2016).

196 Each of the four legs of SherpaTT's suspension has five DOF: the rotation of the whole leg about the  
197 pan axis with respect to the robot body, the two rotations of the inner and outer leg parallelograms,  
198 the steer and drive angle of the wheel. A unique feature of Sherpa is a 6-axis Force-Torque Sensor  
199 (FTS) mounted on the flange of each wheel-drive actuator, providing direct measurement of the force  
200 system exchanged with the ground.

201 The rover also features a six degrees of freedom (DOF) manipulation arm. The arm is designed to  
202 withstand a good portion of the rover's weight to support it during locomotion. However, for the  
203 experiments described in this article, the arm was not involved in locomotion testing.



204  
205

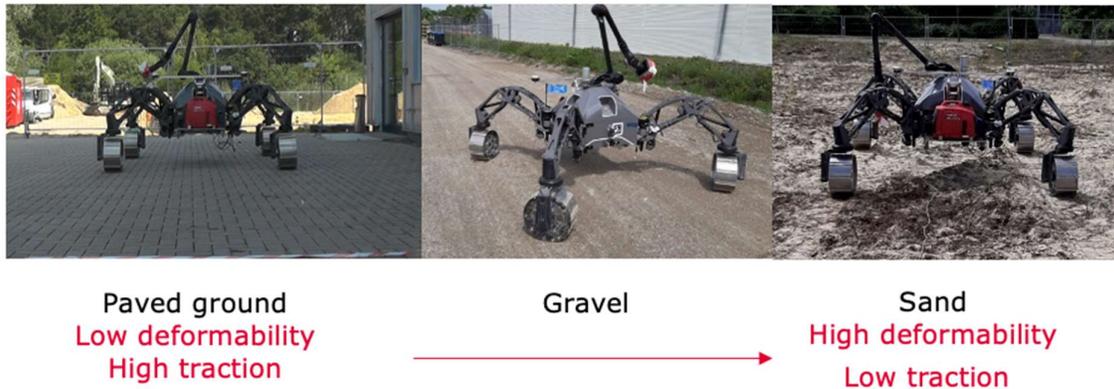
*Figure 1. SherpaTT in a sandy trench during the ADE final field tests in spring 2021.*

206 The logging system provides data at a rate of 100 Hz and comprises the following main proprioceptive  
207 blocks:

- 208 • Inertial Measurement Unit (IMU)
- 209 • Wheel-mounted 6-axis Load Cell (LC). In this study, we adopt solely the LC mounted on the  
210 front left wheel.
- 211 • Joint Telemetry (JT). Each of the 20 actuated joints of the suspension system delivers  
212 telemetry such as supply voltage, supply current, temperatures, PWM duty cycle, position  
213 (relative and absolute), and velocity.

214 The main data set used for this work was generated at the DFKI premises in Bremen, Germany.  
215 SherpaTT was remotely controlled to move for approximately 10 m in a straight line over three types  
216 of terrain: sand, gravel, and paved ground. This represents a varied dataset with a high traction, low  
217 deformability surface (paved ground) at one end, and a surface with low traction and high  
218 deformability (sand) on the other end, with gravel in the middle of the two (Figure 2). For each terrain,  
219 five runs were repeated in forward and reverse drive, except for gravel for which only four runs are  
220 available. Two different drive speeds of the rover were used, namely 0.1 m/s and 0.15 m/s.

221 A second data set was generated in a sand mine close to Bremen (please refer again to Figure 1, GPS  
222 coordinates (DMS format): 53° 18' 54.9" N, 8° 41' 17.3" E) during the ADE's final testing in April  
223 2021. This independent data set is used to predict terrain labels for observations outside the training  
224 data population. In this last environment, the surface traversed was somewhat like the sand case of  
225 the previous settings but the terrain was more compact and wetter. It can be directly observed in  
226 Figure 1 how humid sand got matted to the wheels while traversing, unlike in the previous  
227 environments (Figure 2).



228  
229 Figure 2. Types of surfaces traversed by SherpaTT during the test and development of the system

230

### 231 III.B. Learning Algorithms

#### 232 Support Vector Machine

233 Support Vector Machine is a well-established machine learning solution for soil classification  
234 problems (Bellone et al., 2018; Gonzalez et al., 2019; Reina et al., 2017). This section will present a  
235 summary of the theory behind SVM classification. For a detailed description of SVM algorithm  
236 please refer to (Hastie et al., 2009) and (Vapnik, 2013). An SVM problem is composed of two stages:  
237 training and testing. Given two classes  $A$  and  $B$  (binary classifier), an input training set  $S$  composed  
238 of  $p$  samples and  $n$  features can be defined as:

$$S = \{(\mathbf{x}_i, y_i) : \mathbf{x}_i \in \mathbb{R}^n, y_i \in \{-1, 1\}, i = 1, 2, \dots, p\}$$

$$\text{where } \begin{cases} y_i = 1 & \text{if } \mathbf{x}_i \in A \\ y_i = -1 & \text{if } \mathbf{x}_i \in B \end{cases} \quad (1)$$

239  $\mathbf{x}_i$  are referred as predictors,  $y_i$  represents the response variable. The purpose of the linear SVM  
240 algorithm is to find a decision function  $D$  that allows, in the testing phase, to classify any new sample  
241  $\mathbf{x} \in \mathbb{R}^n$  according to the sign of  $D(\mathbf{x})$ . This is done by finding the hyperplane that maximizes its  
242 distance to the support vectors (i.e., the predictors closest to the hyperplane), while minimizing the  
243 loss due to misclassification. The Lagrangian dual of this optimization problem can be formulated as:  
244

$$\max_{\alpha} \sum_{i=1}^p \alpha_i - \sum_{i=1}^p \sum_{j=1}^p \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$

$$\text{subject to } \sum_{i=1}^p y_i \alpha_i = 0 \quad \text{and} \quad 0 \leq \alpha_i \leq C \quad (2)$$

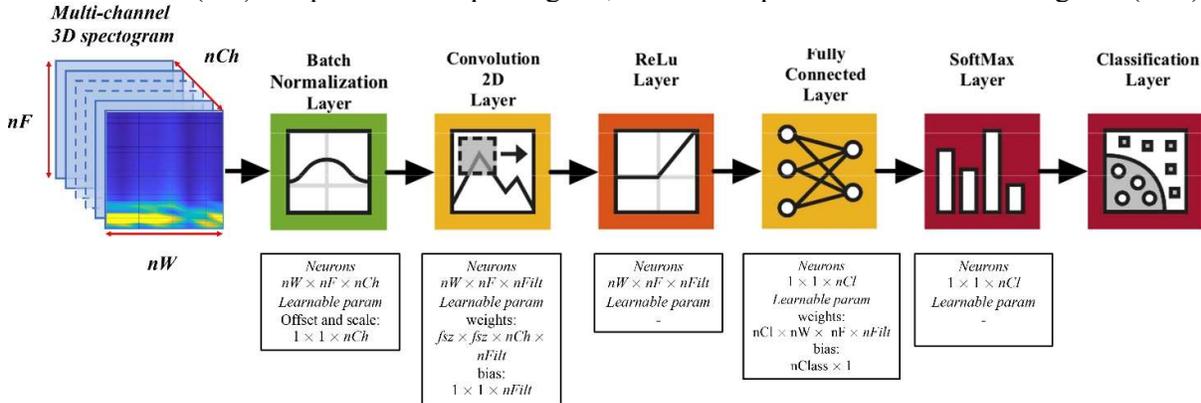
245 where  $\alpha_i$  are Lagrangian multipliers and  $C$  is a parameter called box constraint.

246 The dominant approach for multi-class applications is to reduce the single problem into multiple  
247 binary classification problems (Duan & Keerthi, 2005). One of the most common methods for such  
248 reduction is the Error-Correcting Output Codes (ECOC) model (Dietterich & Bakiri, 1994). The most  
249 important parameter for this method is the coding design, a matrix where elements indicate which  
250 classes are trained by each binary learner, reducing the multiclass problem to a series of binary  
251 problems.

252 In this research, SVM is considered as the benchmark approach that is compared against other  
253 alternatives as a deep Convolutional Neural Network (CNN).

254 *Convolutional Neural Network*

255 In contrast to SVM that uses handcrafted features manually engineered by data analysts, CNN derives  
 256 features automatically from inputs throughout a training process, searching for those that better  
 257 characterize each terrain. However, as input, CNN takes an image-like observation, therefore a first  
 258 practical issue to solve is how to derive a 3D object from several signals. One possible solution,  
 259 proposed in this research, is to resort to Fast Fourier Transform (FFT) to construct magnitude  
 260 spectrograms of the signals then appended into a multichannel object forming the input for the net.  
 261 So, sensory data can be assembled in 3D shape, namely height, width and depth. The height  
 262 corresponds to the frequencies ( $nF$ ) analyzed by the FFT, the width corresponds to the number of  
 263 time windows ( $nW$ ) adopted in the spectrogram, and the depth is the number of signals ( $nCh$ ).



264  
265

Figure 3. Architecture of the convolutional neural network

266 The architecture of the Convolutional Neural Network is shown in Figure 3, where the neural  
 267 dimensions and the learnable variables of each layer are indicated.

268 The first layer takes as input the multichannel spectrogram, next, the batch-normalization layer  
 269 normalizes inside the mini-batch the value kept by each input neuron. Normalization process follows  
 270 equation (3) where  $x_n$  and  $y_n$  are respectively the input and output values of neuron  $n$  of this layer,  
 271 batch mean  $\mu_B$  and standard deviation  $\sigma_B$  are computed during training, while learnable parameters  
 272 offset  $\gamma$  and bias  $\beta$  are searched through optimization across the whole training set. Computational  
 273 constant  $\varepsilon$  can improve numerical stability when variance  $\sigma_B^2$  is small.

$$y_n = \gamma \frac{x_n - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}} + \beta \quad (3)$$

$$\forall n = 1 \dots (nW \cdot nF \cdot nCh)$$

274 The following 2D convolution layer spans the output across time and frequency domain convoluting  
 275 the  $nW \times nF \times nCh$  batch-normalized spectrograms into  $nFilt$  objects of dimensions  $nW \times nF$ . A  
 276 user-specified number of square filters  $nFilt$  with size  $fsz$  are here used to perform convolution  
 277 process briefly described in equation (4)) where  $X$  is the zero-padded neural grid after batch-  
 278 normalization and  $Y$  the output of the convolution process. The learnable parameters of this layer are  
 279 the kernel of filter  $m$ , the weights of matrix  ${}_m\omega$ , and  ${}_m\beta$  the corresponding bias.

$$Y_{w,f,m} = \sum_{c=1}^{nCh} \sum_{i,j=-fsz/2}^{fsz/2} {}_m\omega_{i,j,c} * X_{w+i,f+j,c} + {}_m\beta \quad (4)$$

$$\forall w = 1 \dots nW, \forall f = 1 \dots nF, \forall m = 1 \dots nFilt$$

280 The output of the convolution process is passed to the REcified Linear Unit (ReLU) activated neurons  
 281 in a grid  $nW \times nF \times nFilt$ , fully connected to  $nCl$  neurons where  $nCl$  is the number of terrain classes  
 282 considered. Compared to other activation functions such as the sigmoidal function, ReLu helps in  
 283 preventing the exponential growth in the neural network computation and the “vanishing gradient”

284 problem that is the tendency for the gradient of a neuron to approach zero for high values of the input  
 285 (Kingma & Ba, 2015). The two following layers SoftMax and Classification are standard as output  
 286 layers for classification networks. The function SoftMax is defined in equation (5) where  $x_n$  is the  $n$ -  
 287  $th$  input neuron and  $y_n$  is the corresponding output of this layer.

$$y_n = \frac{\exp(x_n)}{\sum_{i=1}^{nCl} \exp(x_i)}, \forall n = 1 \dots nCl \quad (5)$$

288 The output layer of this network is the classification layer that computes the cross-entropy loss for  
 289 classification among terrains.

### 290 III.C. Parameters of the learning algorithms

291 In this section, the values assigned to the parameters of the learning algorithms are highlighted.  
 292 The parameter set of the SVM-based classifier is indicated in Table 1. It was found empirically to  
 293 give the best balance of sensitivity and specificity (Lin et al., 2002).

294

295 *Table 1. Parameters of the SVM classifier*

PARAMETER	VALUE
C (Box Constraint)	1
Standardize	True
Coding design	one-versus-one

296

297 As for CNN, during the training stage the learnable parameters are updated at each iteration, whereas  
 298 the hyper-parameters are defined by the user to govern the training process.

299 In one iteration, the network analyses the samples contained in the mini-batch. One epoch consists in  
 300 the number of iterations necessary to review the entire training dataset. The training stage stops after  
 301 the network has passed through the entire dataset the number of times specified as the maximum  
 302 number of epochs. It is usually preferred to stop the training before this number has been reached,  
 303 not only because it shortens the time required for training, but also because it prevents overfitting on  
 304 the training set. Therefore, a percentage of the training data is kept apart as validation set, and the  
 305 network evaluates its loss after the number of iterations specified as validation frequency. The  
 306 validation patience is the number of times that this loss can be smaller or equal to the previously  
 307 smallest loss before the training stage stops. The initial learning rate drops by a factor (learn drop  
 308 factor) after a given number of iterations (learn drop period). Part of the hyperparameters is set  
 309 according to the Literature, e.g. the solver and the gradient threshold follows the value suggested in  
 310 (Kingma & Ba, 2015). The remaining parameters have been selected empirically through grid-search  
 311 and they are reported in Table 2.

312 Note that for a fair comparison with SVM, the magnitude spectrograms of the signals used as input  
 313 to CNN are obtained from a time window  $w_s = 2$  s (please refer to Section IV.B).

314

315 *Table 2. Hyper-parameters of the CNN classifier*

PARAMETER	VALUE
Filter size ( $fsz$ )	[5, 5]
Number of filters ( $nFilt$ )	9
Mini-batch size	160
Maximum number of epochs	150
Validation percentage	15%
Validation frequency	20

Validation patience	15
Initial learning rate	0.005
Learn drop factor	0.2
Learn drop period	10

316

317 

## IV. Signal engineering

318 A list of measurements available from the SherpaTT's sensor suite is shown in Table 3, with  
 319 corresponding sensorial group and Signal ID. From a first analysis of Table 3, some of the signals  
 320 may appear seemingly correlated. However, if we consider, for example, body acceleration and wheel  
 321 force, these signals are actually uncorrelated through the flexibility of the suspension system, and  
 322 therefore they are both relevant for the proposed analysis.

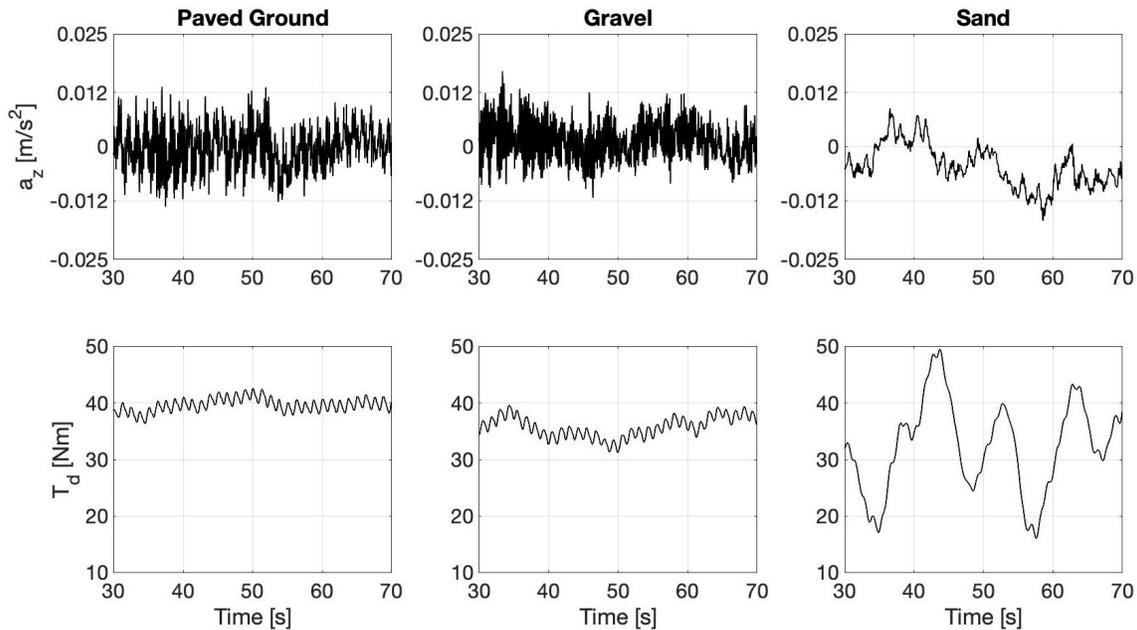
323 Signals that are directly derived from measurements are referred to as direct signals. Conversely,  
 324 signals engineered with expert knowledge combining direct signals are referred to as indirect, as  
 325 explained in the next section.

326 Figure 4 shows a sample time history of the vertical acceleration (gravity-compensated) and drive  
 327 torque experienced by SherpaTT on different terrains. As seen from this figure, signals show a  
 328 signature that seems to change according to the specific surface. The goal of this research is to learn  
 329 this signature to gain terrain awareness. To this aim, it is necessary to select the most relevant signals  
 330 for building an accurate predictor.

331

332 *Table 3. List of available proprioceptive signals*

SIGNAL	SYMBOL	SENSORS	Signal ID
Longitudinal Force	$F_x$	LC	S1
Vertical Force	$F_z$	LC	S2
Drive Torque	$T_d$	LC	S3
Drive electrical current	$C_d$	JT	S4
Drive PWM duty cycle	$PWM_d$	JT	S5
Longitudinal acceleration	$a_x$	IMU	S6
Lateral acceleration	$a_y$	IMU	S7
Vertical acceleration	$a_z$	IMU	S8
Gyro roll rate	$gyro_x$	IMU	S9
Gyro pitch rate	$gyro_y$	IMU	S10
Gyro yaw rate	$gyro_z$	IMU	S11



333  
334 *Figure 4. Vertical acceleration and drive torque (wheel front left) measured while SherpaTT*  
335 *driving straight on different terrains*

#### 336 IV.A. Signal augmentation

337 To improve the information content, an augmentation engine combines multiple direct measurements  
338 based on our understanding of the physical mechanisms underlying the wheel-terrain interaction.  
339 These are few of the many possible signal combinations that can be implemented, and they are chosen  
340 following a trial-and-error approach to provide the best performance over other alternatives. In this  
341 way, nine more indirect signals can be obtained (Table 4). The derivation of these signals is detailed  
342 in this section, and the rationale behind the choice of these entities is also explained.

343 Two main motivations support the proposed augmentation stage. First, two or more signals that are  
344 useless (not relevant) for themselves can be useful when combined. Then, noise reduction and  
345 consequently better class separation may be achieved by adding variables that are seemingly  
346 redundant (Guyon & Elisseff, 2003). This explains why we resort to indirect or combined signals  
347 and include redundant measurements of the same physical quantity.

348 The first indirect signal is the power loss due to the wheel traction on given terrain. It can be derived  
349 from a “mechanical” or “electrical” analysis. The mechanical power can be estimated as follows:

$$P_M = T_d \cdot \omega \quad (6)$$

350 where  $\omega$  is the rotational speed of the wheel. Conversely, the electrical power consumption can be  
351 obtained as:

$$P_E = \eta \cdot V_d \cdot PWM_d \cdot C_d \quad (7)$$

352 where  $V_d$  is the drive voltage,  $C_d$  is the wheel drive current,  $PWM_d$  is the duty cycle of the wheel drive  
353 Pulse Width Modulation, and  $\eta$  is the efficiency of the electric motor, assumed to be constant and  
354 approximately equal to 0.85.

355 Due to the rolling resistance, the direction of the resultant vertical force  $F_z$  might not pass through the  
356 centre of the wheel, with an offset in the direction of the movement (Figure 5). This is especially true  
357 for soft terrain where the impact of rolling resistance is larger.

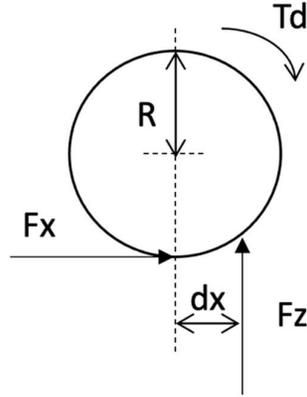


Figure 5. Definition of vertical force offset ( $dx$ )

358  
359

360 Therefore, we can define the vertical force offset  $dx$  from the equilibrium of moments around the  
361 centre of the wheel, neglecting the contribution of rotational inertia:

$$dx = \frac{T_d - F_x \cdot R}{F_z} \quad (8)$$

362 where  $R$  is the loaded wheel radius defined as:

$$R = R_N - \frac{F_z}{k_z} \quad (9)$$

363 being  $R_N$  (=200 mm) the nominal wheel radius, and  $k_z$  the vertical stiffness of the SherpaTT wheel  
364 that was experimentally estimated as 69 N/mm.

365 The friction coefficient is an important entity related with the traction ability over the traversed  
366 surface. In this work, it is estimated in three different ways:

$$\begin{aligned} \mu_1 &= \frac{F_x}{F_z} \\ \mu_2 &= \frac{T_d}{F_z \cdot R} \\ \mu_3 &= \frac{C_d k_T}{F_z \cdot R} \end{aligned} \quad (10)$$

367 where  $k_T$  (17.4 Nm/A in our case) is the scale factor taking into account the torque constant of the  
368 electric motor and the transmission ratio of the motor reducer.

369 Speed deviation is the difference between the angular speed of each wheel  $\omega$  and the average angular  
370 speed of the four wheels  $\bar{\omega}$ . In this work, speed deviation was estimated in two ways:

$$\begin{aligned} SD &= |\omega - \bar{\omega}| \\ SD_{normalised} &= \frac{\omega - \bar{\omega}}{\bar{\omega}} \end{aligned} \quad (11)$$

371 Wheel sinkage is another critical parameter related to rough terrain mobility that can be approximated  
372 as suggested in (Guo et al., 2020):

$$z = R \cdot \left( 1 - \cos \left( 2 \cdot \frac{dx}{R} \right) \right) \quad (12)$$

373  
374

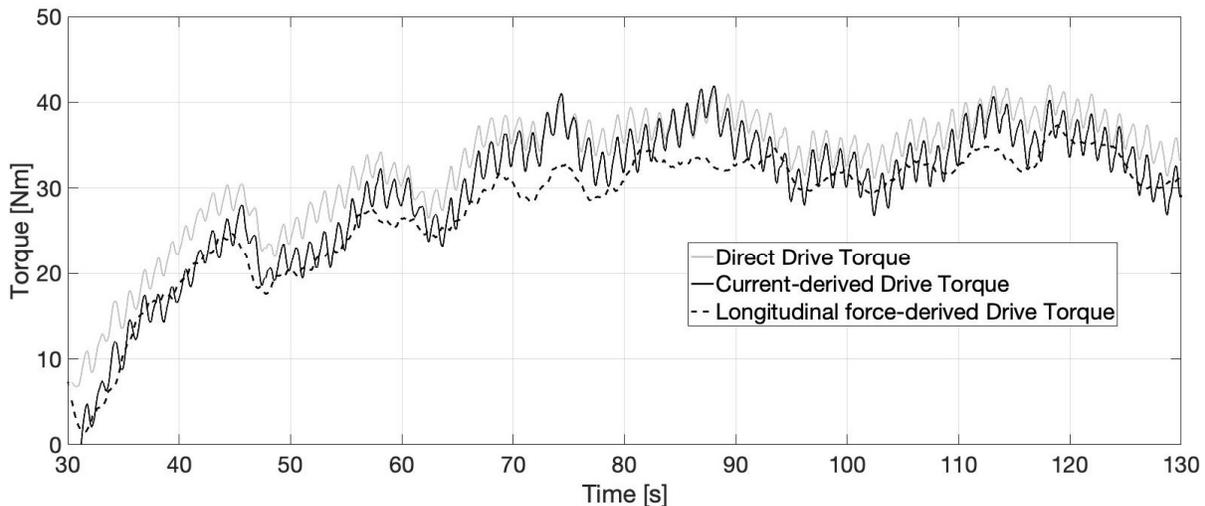
Table 4. List of indirect signals

Signal	SYMBOL	SENSORS	Signal ID
Mechanical Power	$P_M$	LC, JT	S12
Electrical Power	$P_E$	MC	S13
Vertical force offset	$dx$	LC	S14
Friction coefficient 1	$\mu_1$	LC	S15

Friction coefficient 2	$\mu_2$	LC	S16
Friction coefficient 3	$\mu_3$	LC, JT	S17
Speed deviation	$SD$	JT	S18
Normalised speed deviation	$SD_n$	JT	S19
Sinkage	$z$	LC	S20

375  
376  
377  
378  
379  
380  
381  
382

One important aspect is the general data consistency. As an example, Figure 6 shows the drive torque delivered by the left wheel drive motor, measured by three different sensors. Direct torque measurement from the wheel-mounted LC is denoted by a solid grey line, whereas indirect estimation via the associated electric current drawn by the motor is marked by a black solid line. Finally, an alternative indirect measurement via the LC-derived longitudinal force is also plotted using a dashed black line. As seen in this figure, all three measurements show a good agreement. Similar results were observed on different surfaces.



383  
384  
385  
386  
387

Figure 6. Torque applied by the left front wheel of SherpaTT as obtained from: direct measurement of the load cell (solid grey line), indirect measurement via the electric current drawn by the motor (solid black line), or alternatively via the longitudinal force provided by the load cell (dashed black line).

#### 388 IV.B. Feature extraction

389 First, each sensory signal is divided in time windows, and then, for each window features are extracted  
390 as the four main statistical moments. The size of the window,  $w_s$  is a design parameter. It is set as  $w_s=$   
391 2 s corresponding to a traversed terrain patch of about 20 cm (comparable with the wheel radius) at  
392 an average travel speed of 0.1 m/s. In previous works by the authors (Vulpi et al., 2021), it was found  
393 that this value of window size represents a good trade-off between informative content and spatial  
394 resolution.

395 The four statistical moments are mean  $E$ , variance  $\sigma$ , skewness  $Sk$  and kurtosis  $Ku$  and are defined as  
396 follows:

$$E_i = \frac{1}{N} \sum_{n=1}^N x_n$$

$$\sigma_i^2 = \frac{1}{N} \sum_{n=1}^N (x_n - E_i)^2 \quad (13)$$

$$Sk_i = \frac{1}{N} \frac{\sum_{n=1}^N (x_n - E_i)^3}{\left(\sqrt{\sigma_i^2}\right)^3}$$

$$Ku_i = \frac{1}{N} \frac{\sum_{n=1}^N (x_n - E_i)^4}{\left(\sqrt{\sigma_i^2}\right)^4}$$

397 where  $x_n$  is the value of the signal at the  $n^{\text{th}}$  time step and  $N$  is the total number of time-steps for the  
398  $i^{\text{th}}$  window.

399 The extraction of the statistical features brings the size of the SVM-feature space to 80 (20 signals  
400 multiplied by their 4 statistical moments). The generic feature will be indicated as  $SiMj$ , where  $i$   
401 ( $i=1, \dots, 20$ ) represents the signal ID, whereas  $j$  represents the statistical moment ( $j=1, \dots, 4$ ).  
402

#### 403 IV.C. Feature selection

404 Retaining only the features with the highest information content reduces the computational cost while  
405 preserving the accuracy of the model. The selection process can be performed via feature scoring  
406 using appropriate validity indices. Then, an iterative search algorithm can be followed to select a  
407 reduced best feature space.

##### 408 *Validity indexing*

409 A validity index can be assigned to each feature. This index represents a measure of the information  
410 content of the feature. In this work, two validity indices are considered: the Pearson Coefficient (PC)  
411 (Hastie et al., 2009), and the WB index (Zhao & Fränti, 2014).

412 The PC index can be computed through linear regression of a feature against the 3 classes of terrain,  
413 e.g., sand, gravel, and paved ground. The higher the PC, the larger the information content of the  
414 feature. Although this index can be successfully used for 2-class classification problems  
415 (Dimastrogiovanni et al., 2020), it might be difficult to implement it for multi-class cases like the  
416 one presented in this work, because the number assigned to each type of terrain is arbitrary. To  
417 overcome this issue, first, the PC index is computed for each terrain pair (e.g., sand-gravel, gravel-  
418 paved ground, and sand-paved ground), and then averaged. For example, the PC index of the feature  
419  $SiMj$  against the classes 1 and 2 (sand and gravel) can be calculated as (Guyon & Elisseeff, 2003):

$$\frac{1}{2}PC_{SiMj} = \frac{cov(\frac{1}{2}F_{SiMj}, \frac{1}{2}y)}{\sqrt{var(\frac{1}{2}F_{SiMj})var(\frac{1}{2}y)}} \quad (14)$$

420 where  $\frac{1}{2}F_{SiMj}$  is a vector containing all values of the feature  $SiMj$  for terrains 1 and 2, whereas  $\frac{1}{2}y$   
421 contains class values (1 or 2) for each element of  $\frac{1}{2}F_{SiMj}$ . Similarly,  $\frac{2}{3}PC_{SiMj}$  (PC index of feature  
422  $SiMj$  against the classes gravel and paved ground) and  $\frac{1}{3}PC_{SiMj}$  (PC index of feature  $SiMj$  against the  
423 classes sand and paved ground) follow the same principle.

424 The overall PC index for feature  $SiMj$  can be now computed as follows:

$$PC_{SiMj} = \frac{\frac{1}{2}PC_{SiMj} + \frac{2}{3}PC_{SiMj} + \frac{1}{3}PC_{SiMj}}{3} \quad (15)$$

425 In addition, the WB index can be computed for feature  $SiMj$ :

$$WB_{SiMj} = m \cdot \frac{SSW_{SiMj}}{SSB_{SiMj}} \quad (16)$$

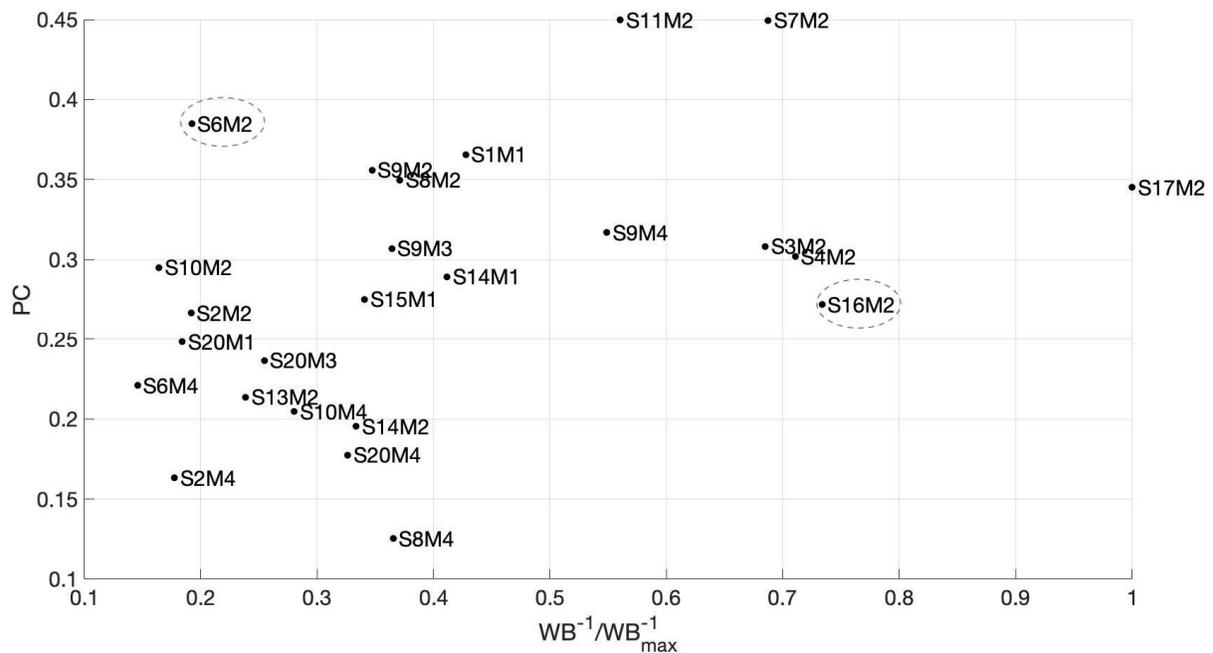
426 where  $SSW$  is the sum of square within classes and  $SSB$  is the sum of squares between classes,  
427 computed as follows:

$$SSW_{SiMj} = \sum_{k=1}^{nCl} \sum_{s=1}^{n_k} (x_s - \mu_k)^2 \quad (17)$$

$$SSB_{SiMj} = \sum_{k=1}^{nCl} n_k (\mu_k - \mu)^2$$

428 where  $x_s$  is the  $s^{\text{th}}$  sample of feature  $SiMj$ ,  $\mu_k$  is the class  $k$  centroid value,  $\mu$  is the overall dataset  
 429 centroid value,  $n_k$  is the number of samples in class  $k$  and  $nCl$  ( $=3$ ) is the number of classes. A low  
 430 value of  $WB_{SiMj}$  indicates that classes form compact and distant clusters relatively to feature  $SiMj$ .  
 431 Therefore, the score assigned to each feature will be  $WB^{-1}$ : the higher the  $WB^{-1}$ , the better the feature  
 432 for classification purposes.

433 The rationale behind using two validity indices is that the WB and PC have two different statistical  
 434 meanings: the former describes the compactness of classes, the latter shows the correlation between  
 435 a given feature and the type of terrain. One may think that a feature with a low value of PC index will  
 436 also have a relatively low value of  $WB^{-1}$  index. However, this is not always true, and exceptions do  
 437 occur. For example, Figure 7 shows the distribution of PC and WB indices for the 25 features with the highest scores.  
 438 S6M2 is the feature with the third highest value of PC index, but it is only the 21<sup>st</sup>  
 439 feature in terms of  $WB^{-1}$ . Similarly, S16M2 is the feature with the second highest value of  $WB^{-1}$   
 440 index, but it is only the 14<sup>th</sup> in terms of PC. This shows that the two indices rank the features in  
 441 different ways, therefore they complement each other very well.



442

443 *Figure 7. PC and WB indices distribution for the most relevant features*

#### 444 *Selection algorithm*

445 The proposed selection approach is based on the iterative search scheme presented in the block  
 446 diagram of Figure 8. The input to the algorithm is the full set of  $n_{feat}$  ( $=80$ ) features. These features  
 447 are then ranked using the output of one of the two validity indices (PC or WB) as a score. The best  
 448 feature set is initialized with the first  $n_{min} - 1$  ( $=2$ ) features of the ranking. At this point, the objective  
 449 is to iterate on all the remaining features to find those which provide better classification performance.  
 450 In each iteration, identified with the index  $i$  that varies from  $n_{min}$  to  $n_{feat}$ , the  $i^{\text{th}}$  feature in the ranking  
 451 is added provisionally to the best feature set. Then, an SVM-based classifier is trained and evaluated  
 452 in terms of F1 score via 5-fold cross validation. The  $k$ -fold cross validation process partitions data  
 453 into  $k$  randomly chosen subsets (or folds) of roughly equal size. Therefore, to improve the robustness  
 454 of the feature selection algorithm, the training phase is repeated  $n_{train}$  ( $=10$ ) times and the final F1  
 455 score is computed as the average of the scores obtained at each training phase. If the final F1 score is

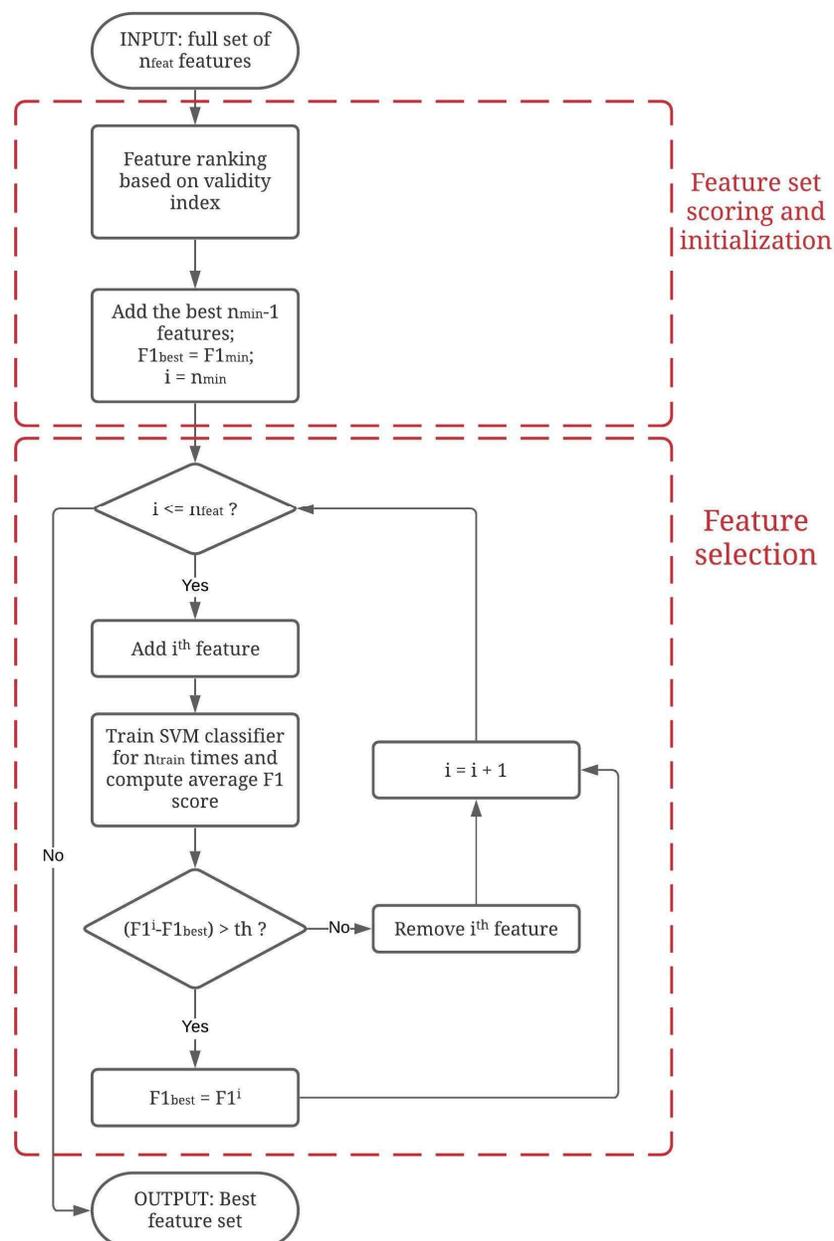
456 sufficiently higher than the best F1 score obtained so far, the  $i^{\text{th}}$  feature is kept in the best feature set,  
 457 and the best F1 score is updated. Otherwise, the  $i^{\text{th}}$  feature is discarded from the best set and not  
 458 considered for training purposes.

459 In order to facilitate the reading of the block diagram in Figure 8, the meaning and the numerical  
 460 values of the parameters involved in the selection process are collected in Table 5.

461 Table 5. List of parameters involved in the feature selection approach

Parameter	Description	Value
$n_{min}$	Minimum number of features	3
$F1_{min}$	Minimum F1 score	60%
$th$	Accepted improvement (threshold) in the F1 score	5%
$n_{train}$	Number of trainings for each new best feature set	5
$n_{feat}$	Number of features in the initial full feature set	80

462



463

464

Figure 8. Block diagram of the proposed feature selection algorithm

465 The selection process discussed in Figure 8 can be repeated for each one of the two validity indices.  
 466 Eventually, two best reduced feature spaces will be obtained: one associated with the PC and the  
 467 other with the WB index. To further improve the robustness of the selection algorithm, the union of  
 468 these two sets is chosen as the best for SVM training purposes. The 18 selected features are listed in  
 469 Table 6. It is worth noting that three features extracted from indirect signals are included as well,  
 470 thus, proving the utility of the signal augmentation phase.

471 A 3D plot of the three most relevant features in terms of WB index is shown in Figure 9 to help the  
 472 reader to easily visualize the result of the whole selection process. As shown in this figure, the sand  
 473 data form a quite compact cluster, with relatively low values of all three features. Conversely, gravel  
 474 and paved ground data show higher values of S7M2 (variance of  $a_y$ ) than sand and differentiate  
 475 prevalently for values of S17M2 (variance of  $\mu_3$ ).

476 *Table 6. Best feature set*

Signal	Statistical moment	Direct or Indirect	Feature ID	$WB^{-1}/WB^{-1}_{MAX}$	PC
$\mu_3$	Variance	Indirect	S17M2	1.00	0.345
$a_y$	Variance	Direct	S7M2	0.691	0.449
$T_d$	Variance	Direct	S3M2	0.685	0.308
$gyro_z$	Variance	Direct	S11M2	0.561	0.450
$gyro_x$	Kurtosis	Direct	S9M4	0.549	0.317
$F_x$	Mean	Direct	S1M1	0.428	0.366
$a_z$	Variance	Direct	S8M2	0.371	0.350
$gyro_x$	Skewness	Direct	S9M3	0.364	0.307
$gyro_x$	Variance	Direct	S9M2	0.348	0.356
$\mu_1$	Mean	Indirect	S15M1	0.342	0.275
$z$	Kurtosis	Indirect	S20M4	0.327	0.177
$a_x$	Variance	Direct	S6M2	0.192	0.385
$F_z$	Variance	Direct	S2M2	0.192	0.267
$gyro_y$	Variance	Direct	S10M2	0.164	0.295
$F_x$	Variance	Direct	S1M2	0.143	0.253
$a_y$	Kurtosis	Direct	S7M4	0.048	0.115
$PWM_d$	Mean	Direct	S5M1	0.013	0.062
$gyro_z$	Mean	Direct	S11M1	0.011	0.064

477

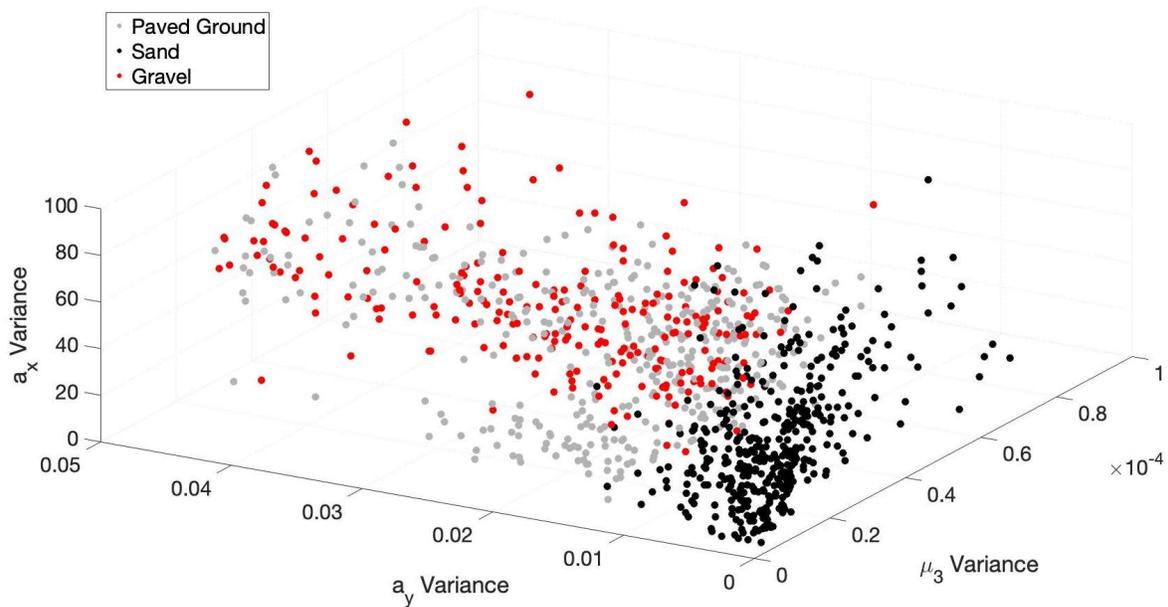


Figure 9. 3D plot of the first three features with the highest score of  $WB^{-1}$

478  
479

## 480 V. Results and discussion

481 In this section, the results of the generalization problem are shown on the main dataset. Next, results  
482 for two extrapolation cases are presented.

### 483 V.A. Generalization

484 In the generalization problem, only the main data set is used (e.g., experiments on paved ground,  
485 gravel, and sand). The algorithms are tested via 5-fold cross validation. The data set comprises of  
486 1204 samples, where a sample corresponds to a 2-second time window. Of these 1204 samples, 443  
487 are collected on paved ground, 338 on gravel and 423 on sand.

488 One of the objectives of this paper is to demonstrate how a proper feature selection algorithm can  
489 reduce the computational and memory cost of the model, while maintaining a similar accuracy in  
490 prediction. Table 7 shows comparison between the two machine learning algorithms in terms of  
491 accuracy and computational burden. Moreover, SVM is tested with three different feature sets:

- 492 • Direct feature set (44 features)
- 493 • Full feature set (80 features)
- 494 • Best feature set (18 features)

495 while CNN is tested with three different signal sets:

- 496 • Direct signal set (11 signals)
- 497 • Full signal set (20 signals)
- 498 • Best signal set (13 signals)

499 The signals used for training CNN correspond to those used to compute SVM features. In fact, the  
500 44 direct features are the 4 statistical moments of the 11 direct signals and the full 80-feature set is  
501 composed by the 4 statistical moments of the full 20-signal set. Furthermore, the training set for CNN  
502 includes the signals used to derive the features in the best feature set. Namely, the 13 best signals are:  
503 friction coefficients 1 and 3, longitudinal, lateral and vertical accelerations, drive torque, yaw, pitch  
504 and roll rates, longitudinal and vertical forces, sinkage, drive PWM.

505 The accuracy of the SVM model trained with the direct and full feature sets is 89.8% and 90.8%,  
506 respectively. With the full feature set, more samples are correctly classified by SVM, but memory  
507 usage has increased by 82%, training time by 32%, testing time by 71% and feature extraction time  
508 by 50%. This proves the effectiveness of the signal augmentation in terms of accuracy and shows the

509 drawbacks in terms of computational burden. The purpose of feature selection is to reduce the  
 510 computational cost, without losing classification accuracy. The results presented for SVM trained  
 511 with the best feature set, prove that the feature selection algorithm proposed in this work is effective.  
 512 In fact, the accuracy reaches 90.9% and when compared to the SVM trained on the full feature set,  
 513 while the model memory usage is reduced by 77%, training time by 6%, testing time by 29%, feature  
 514 extraction time by 33%.

515 The effectiveness of both input signal augmentation and feature selection is also confirmed by the  
 516 results presented for CNN. This deep learning algorithm gains in terms of accuracy from signal  
 517 augmentation reaching 96.4%. Using the full signal set still results for CNN in the same drawbacks  
 518 presented for SVM: model memory usage increased by 18%, training time by 36%, feature extraction  
 519 time by 77%. In contrast with SVM, testing time for CNN with full signal set is reduced by 22%.  
 520 Training CNN with the best signals resulting from feature selection leads to an accuracy of 96.2%  
 521 and when compared to the full-signal CNN, the model memory usage is reduced by 14%, training  
 522 time by 27%, testing time by 4%, feature extraction time by 41%.

523 Feature extraction times presented in the last row of Table 7 are suitable for online application for  
 524 both SVM and CNN, even if construction of multichannel spectrograms from best signals for CNN  
 525 takes about 2.1 ms more than construction of best features for SVM. It should also be noted that  
 526 feature extraction time for both SVM and CNN can be further improved by optimizing the current  
 527 MatLab code using vectorization or processing the data directly with a C++ code. Note that at the  
 528 time of writing of the paper, the algorithms and the dataset are under revision in a private Github  
 529 repository that will be made available to the interested readers upon paper publication.

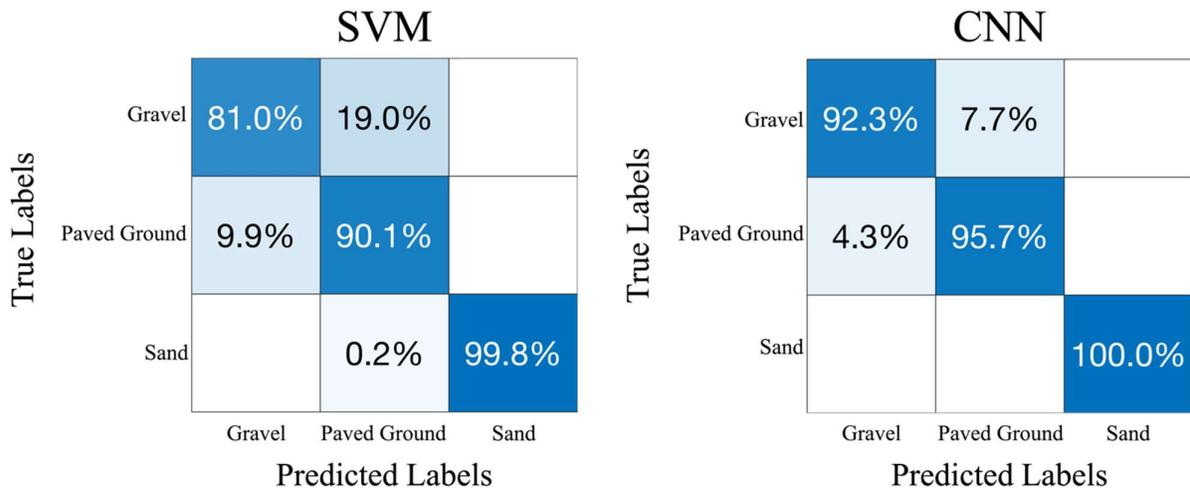
530 Confusion matrixes for both SVM and CNN are shown in Figure 10 only for best feature and best  
 531 signal sets. Sensitivity results for each class are contained in the diagonal elements of each confusion  
 532 matrix. The performance of both models in terms of precision, recall and F1 score are shown in Table  
 533 8. Both models perform good in generalization of data, with CNN being slower but significantly more  
 534 accurate. This increase in classification accuracy is not the main advantage for CNN classification  
 535 model with respect to SVM. Where the two models show the greatest difference in classification  
 536 performance is indeed extrapolation, as shown in the next section.

537

538 *Table 7. Performance comparison between terrain classifiers trained on different feature sets: direct,*  
 539 *full, best feature set*

Feature and signal sets	SVM			CNN		
	Direct	Full	Best	Direct	Full	Best
Accuracy [%]	89.8	90.8	<b>90.9</b>	95.6	<b>96.4</b>	96.2
Model memory usage [kB]	547.6	996.9	<b>228.0</b>	44.9	53.2	45.8
Training time [ms]	118.9	157.7	148.0	1.07 e4	1.46 e4	1.06 e4
Testing time [ms]	17.4	29.8	21.2	153.0	119.4	114.7
Feature extraction time [ms]	0.6	0.9	0.6	2.6	4.6	2.7

540



541

542 *Figure 10. Generalization Results for best features SVM and best signals CNN*543 *Table 8. Accuracy, Precision, Recall and F1 score for SVM and CNN in generalization*

Class	SVM			CNN		
	Gravel	Paved Ground	Sand	Gravel	Paved Ground	Sand
<b>Precision [%]</b>	89.1	82.4	100	80.3	82.2	100
<b>Recall [%]</b>	81.0	90.1	99.8	92.3	95.7	100
<b>F1 score [%]</b>	84.9	86.1	99.9	85.9	88.4	100

544

545 **V.B. Extrapolation**

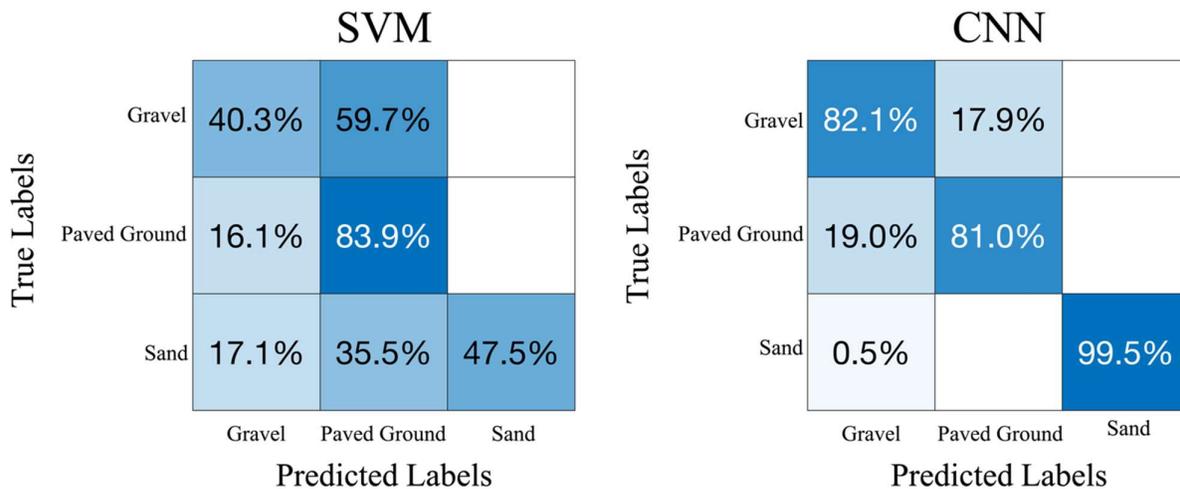
546 In the extrapolation problem, the operating conditions of training and testing sets are different,  
 547 therefore these sets do not come from the same population. In this work two extrapolation cases are  
 548 presented. The first one deals with varying rover speed, whereas the second one assesses the  
 549 performance of the algorithms on a terrain unseen in the training phase.

550 *Testing on a new vehicle velocity*

551 During the experiments with SherpaTT, the rover was controlled at two different speeds: 0.1 m/s and  
 552 0.15 m/s. Of the 14 runs, 7 were conducted at low speed (0.1 m/s) and 7 at high speed (0.15 m/s).  
 553 Data collected at low-speed form the low-speed distribution, whereas data collected at high-speed  
 554 belong to the high-speed distribution. In the extrapolation problem presented here, low-speed data  
 555 are used as training set, while high-speed data are used as testing set. Both sets belong to the main  
 556 dataset (paved ground, gravel, and sand).

557 Proprioceptive sensorial data are very useful for terrain classification but also show a strong  
 558 dependency from traversing speed (Bai et al., 2019). Most terrain classification algorithms analyse  
 559 and classify proprioceptive data acquired at constant traversing velocity on different terrains. Studies  
 560 have been also conducted to show dependency of terrain classification performances from rover's  
 561 traversing speed, searching for the velocity that maximizes classification performance. For being able  
 562 to classify the traversed terrain at any travelling speed a rover should be equipped with a model trained  
 563 on a vast variety of possible traversing speeds or could only use speed independent features that are

564 difficult to construct and may not be well suited for terrain classification. Another way of achieving  
 565 the goal of sensing and classifying the terrain at any travelling speed is using a model that shows  
 566 good results when tested on data acquired at a traversing velocity different from the one used for  
 567 training. Figure 11 contains the confusion matrixes for both SVM and CNN when trained on low-  
 568 speed data and tested on high-speed ones. As can be seen, despite both models showed good results  
 569 in generalization only CNN is also capable of extrapolating the information of the traversed terrain  
 570 from data acquired at a different speed. The two models were still trained and tested using only best  
 571 feature set for SVM and corresponding signal set for CNN. While CNN keeps classification accuracy  
 572 as high as 89.5%, SVM becomes unreliable achieving only 55.7% of correctly classified data samples.  
 573 The performances of both models in terms of precision, recall and F1 score are shown in Table 9.  
 574 It should also be pointed out that high-speed data used as testing constitute 50% of available data,  
 575 representing therefore testing set larger than the one usually used (20-30%). The robustness of CNN's  
 576 classification performance on a large testing set composed by data acquired at a different speed  
 577 suggests that this model is well suited for terrain classification purposes. Moreover, the features  
 578 automatically learned from signal spectrograms appear to be more reliable than statistic ones and  
 579 represent a better choice to be able to classify the traversed terrain at various travelling velocities.  
 580 Similar results are obtained when trained on high-speed data and tested on low-speed data, and they  
 581 are omitted for brevity sake.



582

583 *Figure 11. Extrapolation results for best features SVM and corresponding signals CNN*

584

584 *Table 9. Precision, Recall and F1 score for SVM and CNN in extrapolation using varying velocity*

Class	SVM			CNN		
	Gravel	Paved Ground	Sand	Gravel	Paved Ground	Sand
<b>Precision [%]</b>	54.8	46.8	100	80.3	82.2	100
<b>Recall [%]</b>	40.3	83.9	47.5	82.1	81.0	99.5
<b>F1 score [%]</b>	46.4	60.1	64.4	81.2	81.6	99.7

585

586 *Testing on an independent dataset*

587 The second extrapolation use case aims to evaluate the system response when labeling observations  
 588 collected on a terrain different from those used in training (independent dataset). To this aim, the

589 ground classifier previously trained on the main data set (formed by paved ground, gravel, and sand)  
 590 is further validated on a representative dataset gathered from a second field test campaign run in a  
 591 planetary analogue terrain in a sand mine near Bremen (see Figure 12).

592 For this extrapolation challenge, we have tried to generalize the classification problem at hand by  
 593 referring to terrain difficulty labels rather than specific terrain classes, as explained in Table 10.  
 594 Adopting the proposed terrain difficulty scale, paved ground and sand can be seen as the opposite  
 595 extremes. Firm ground offers better traction and less compressibility, therefore a low difficulty label  
 596 can be assigned to it. Conversely, soft ground poses more challenges, and it is scored as a highly  
 597 difficult surface. Then, the difficulty degree associated with an unknown observation can be  
 598 considered as inversely proportional to the distance from the class sand. One should note that such a  
 599 generalization effort can be useful or necessary for the practical implementation of planetary  
 600 exploration terrain classifiers that can be only trained on Earth using representative analogue surfaces,  
 601 and then applied to unknown planetary surfaces via extrapolation.

602  
 603 The sand mine independent dataset consists of 302 samples, where, again, a sample corresponds to a  
 604 2-second window. It should be also underlined that, although ground-truth data is not available for  
 605 this extrapolation problem, the terrain in the sand mine can be expected as a surface with medium-  
 606 high difficulty, like the sand type of the main dataset (Figure 2) but somewhat more compact and  
 607 humid. As an indicative measure, sample tracks left by the wheels on the sand mine terrain are shown  
 608 in Figure 12(b).

609



610

611

612 *Figure 12. (a) Sherpa TT during the sand mine testing; (b) a close up of the tracks left by the wheels*

613

614 *Table 10. Category of difficulty assigned to each terrain type of the training set.*

Terrain type	Equivalent Category of Terrain difficulty
Sand	High
Gravel	Medium
Paved Ground	Low

615

616 The classification results obtained from SVM and CNN are collected in Table 11 showing predicted  
 617 labels of terrain difficulty. Out of the 302 samples, the SVM-based algorithm classifies 71.2% as high  
 618 difficult terrain, 17.2% as medium and 11.6% as low. CNN performs similarly, classifying 69.9% of  
 619 the new terrain samples as highly difficult, 24.2% as medium and 5.9% as low. A relatively low  
 620 percentage of the test samples (about 12 % for SVM and 6% for CNN) is classified as hard soil.

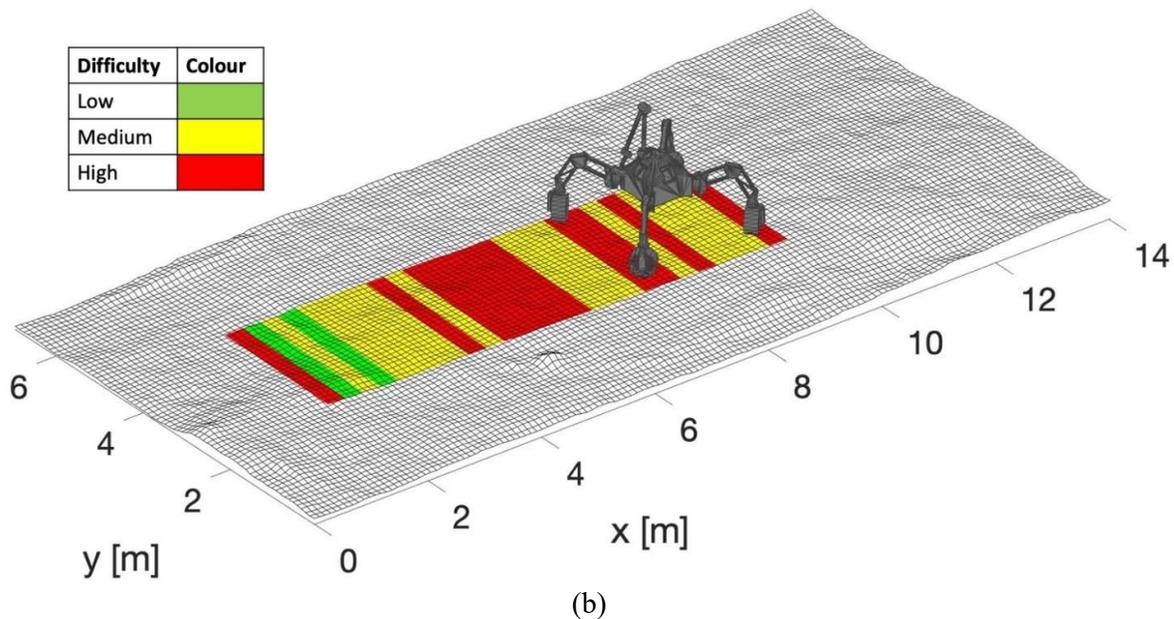
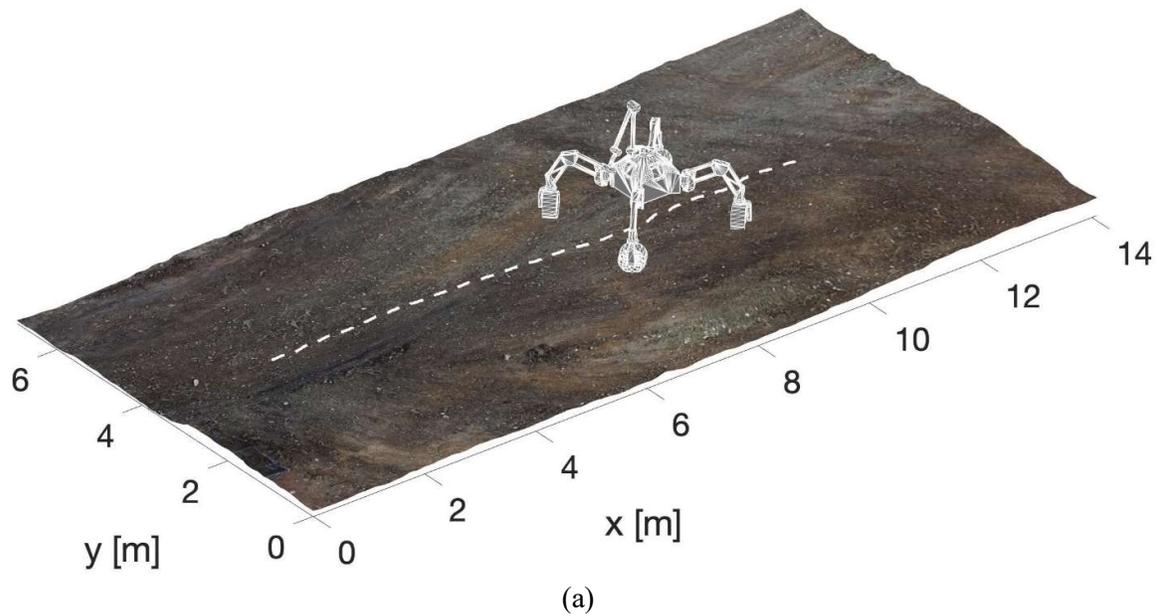
621 *Table 11. Terrain difficulty predictions as obtained from SVM and CNN in the sand mine test*

Terrain Difficulty labels	SVM	CNN
High	215	211
Medium	52	73
Low	35	18

622

623 For an easier visualization, the results obtained from the CNN-based classifier are presented in Figure  
624 13 during a sample straight run using a semantic labelling where the successive terrain patches  
625 traversed by the rover are marked according to a color map that reflects the terrain difficulty scale of  
626 Table 10 (see also to the inset of Figure 13b). We recall that three discrete levels of terrain difficulty  
627 are considered: low, medium, and high.

628 Figure 13a shows the 3D stereo-generated map of the environment with overlaid a CAD model of  
629 SherpaTT and the path followed by the rover denoted with a dashed white line, whereas in Figure  
630 13b the corresponding terrain labeling is reported with terrain patches marked respectively in red,  
631 yellow, and green, for high, medium, and low difficulty. In this test that was performed on fairly  
632 homogeneous terrain, the system mostly classifies the sand mine surface as of medium-high difficulty  
633 with two erroneous predictions (low difficulty) between 2 and 3 m.



638 *Figure 13. Semantic labeling using discrete terrain difficulty categories: (a) 3D stereo-generated*  
 639 *map of the environment with overlaid the path (dashed white line) followed by the rover, (b)*  
 640 *corresponding terrain difficulty visualization. Terrain patches are marked respectively in red, yellow,*  
 641 *and green, for high, medium, and low difficulty.*

## 642 VI. Conclusions

643 This work presented an approach to soil classification that relies on proprioceptive sensing only, e.g.  
 644 accelerations, forces, torques, and electrical currents. The algorithms developed are validated on data  
 645 collected during tests performed with the hybrid wheeled-legged rover SherpaTT. The physics-based  
 646 signal augmentation process presented in this paper uses 11 proprioceptive measurements to produce  
 647 a large set of 80 features for SVM and 20 signals for CNN. This improved the information content as  
 648 proved by the high classification accuracy obtained in generalization (90.8% for SVM and 96.4 %  
 649 for CNN). The proposed feature selection algorithm allows SVM to retain a high classification  
 650 accuracy with only a portion of the full set (18 features), with successful reductions in memory usage

651 (-77%) and required time for training (-6%), testing (-29%) and feature extraction (-33%). The same  
652 benefits also apply for CNN when using a reduced set of 13 signals related to the 18 best SVM  
653 features, improving memory usage (-14%), training time (-27%), testing time (-4%) and feature  
654 extraction time (-41%). The comparison between SVM and CNN shows good capabilities of both  
655 models in generalization, with accuracy higher than 90%. More challenging extrapolation problems  
656 have been tackled as well to evaluate the impact of varying operating conditions and site of the  
657 acquisition. In these tests, CNN outperformed the SVM counterpart. When tested on a new vehicle  
658 velocity, CNN reached an accuracy of 89.5%, against 55.7% held by SVM. When tested on a new  
659 terrain, CNN recognized its deformability class more frequently than SVM, correctly classifying 6%  
660 more of the available samples. Based on these results, the proposed CNN qualifies as a good  
661 algorithm for soil classification even in the presence of disturbances and unknown conditions.

662 This work proved that is possible to use only proprioceptive features to infer the signature of a  
663 particular surface via learning algorithms. Moreover, the presented promising results suggest the  
664 possibility to extend rover travelling distance thanks to on-board integration of the developed learning  
665 algorithms.

666 Future developments of this research refer to: i) continuous training of the system by incorporating  
667 instances of “new terrain” classes during normal operations, therefore making the system adaptive,  
668 ii) augmenting the classifier with new special classes; for example, instances of excessive wheel  
669 slippage (close to 100%) can be used to train a hazard class to inform the rover of impending  
670 immobilization conditions, iii) combining the proposed framework using proprioceptive signals with  
671 exteroceptive signals. The latter would enable the vehicle to predict hazards or trapping conditions  
672 before driving through the ground, e.g., based on non-contact information coming from vision  
673 sensors.

#### 674 *Software repository*

675 The codes and data used for this research will be made publicly available at a Github repository.

#### 676 *Acknowledgements*

677 The financial support of the projects: Autonomous Decision making in very long traverses (ADE),  
678 H2020 (Grant No. 821988), Agricultural inTeroperabiLity and Analysis System (ATLAS), H2020  
679 (Grant No. 857125), and multimodal sensing for individual pLANT phenOtypiNg in agriculture  
680 rObotics (ANTONIO), ICT-AGRI-FOOD COFUND (Grant No. 41946) is gratefully acknowledged.

## 681 VII. Bibliography

- 682 Bai, C., Guo, J., Guo, L., & Song, J. (2019). Deep multi-layer perception based terrain classification  
683 for planetary exploration rovers. *Sensors (Switzerland)*, *19*(14), 3102.  
684 <https://doi.org/10.3390/s19143102>
- 685 Barnes, D., Maddern, W., & Posner, I. (2017). Find your own way: Weakly-supervised segmentation  
686 of path proposals for urban autonomy. *Proceedings - IEEE International Conference on*  
687 *Robotics and Automation*, 203–210. <https://doi.org/10.1109/ICRA.2017.7989025>
- 688 Bellone, M., Reina, G., Caltagirone, L., & Wahde, M. (2018). Learning Traversability from Point  
689 Clouds in Challenging Scenarios. *IEEE Transactions on Intelligent Transportation Systems*,  
690 *19*(1), 296–305. <https://doi.org/10.1109/TITS.2017.2769218>
- 691 Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new  
692 perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(8), 1798–  
693 1828. <https://doi.org/10.1109/TPAMI.2013.50>
- 694 Brooks, C. A., & Iagnemma, K. (2005). Vibration-based terrain classification for planetary  
695 exploration rovers. *IEEE Transactions on Robotics*, *21*(6), 1185–1191.  
696 <https://doi.org/10.1109/TRO.2005.855994>
- 697 Cordes, F., & Babu, A. (2016). SherpaTT: A Versatile Hybrid Wheeled-Leg Rover. *Proceedings of*  
698 *the 13th International Symposium on Artificial Intelligence, Robotics and Automation In Space,*  
699 *(ISAIRAS-16)*.
- 700 Cordes, F., Kirchner, F., & Babu, A. (2018). Design and field testing of a rover with an actively  
701 articulated suspension system in a Mars analog terrain. *Journal of Field Robotics*, *35*(7), 1149–  
702 1181. <https://doi.org/10.1002/rob.21808>
- 703 Cowen, R. (2005). Opportunity rolls out of Purgatory. *Science News*, *167*(26), 413.
- 704 Dietterich, T. G., & Bakiri, G. (1994). Solving Multiclass Learning Problems via Error-Correcting  
705 Output Codes. *Journal of Artificial Intelligence Research*, *2*, 263–286.  
706 <https://doi.org/10.1613/jair.105>
- 707 Dimastrogiovanni, M., Cordes, F., & Reina, G. (2020). Terrain estimation for planetary exploration  
708 robots. *Applied Sciences*, *10*(17), 6044. <https://doi.org/10.3390/app10176044>
- 709 Duan, K. B., & Keerthi, S. S. (2005). Which is the best multiclass SVM method? An empirical study.  
710 *International Workshop on Multiple Classifier Systems*, 278–285.  
711 [https://doi.org/10.1007/11494683\\_28](https://doi.org/10.1007/11494683_28)
- 712 DuPont, E. M., Moore, C. A., Collins, E. G., & Coyle, E. (2008). Frequency response method for  
713 terrain classification in autonomous ground vehicles. *Autonomous Robots*, *24*(4), 337–347.  
714 <https://doi.org/10.1007/s10514-007-9077-0>
- 715 Dutta, A., & Dasgupta, P. (2017). Ensemble Learning with Weak Classifiers for Fast and Reliable  
716 Unknown Terrain Classification Using Mobile Robots. *IEEE Transactions on Systems, Man,*  
717 *and Cybernetics: Systems*, *47*(11), 2933–2944. <https://doi.org/10.1109/TSMC.2016.2531700>
- 718 *ESA, Robotic exploration of Mars*. (2021). <http://exploration.esa.int/mars/>

- 719 Giguere, P., & Dudek, G. (2011). A simple tactile probe for surface identification by mobile robots.  
720 *IEEE Transactions on Robotics*, 27(4), 534–544. <https://doi.org/10.1109/TRO.2011.2119910>
- 721 Gonzalez, R., Chandler, S., & Apostolopoulos, D. (2019). Characterization of machine learning  
722 algorithms for slippage estimation in planetary exploration rovers. *Journal of Terramechanics*,  
723 82, 23–34. <https://doi.org/10.1016/j.jterra.2018.12.001>
- 724 Guo, J., Guo, T., Zhong, M., Gao, H., Huang, B., Ding, L., Li, W., & Deng, Z. (2020). In-situ  
725 evaluation of terrain mechanical parameters and wheel-terrain interactions using wheel-terrain  
726 contact mechanics for wheeled planetary rovers. *Mechanism and Machine Theory*, 145, 103696.  
727 <https://doi.org/10.1016/j.mechmachtheory.2019.103696>
- 728 Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of*  
729 *Machine Learning Research*, 3, 1157–1182. <https://doi.org/10.1162/153244303322753616>
- 730 Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: data mining,  
731 inference, and prediction. In *Springer Science & Business Media* (Second).
- 732 Ishikawa, R., Hachiuma, R., & Saito, H. (2021). Self-Supervised Audio-Visual Feature Learning for  
733 Single-Modal Incremental Terrain Type Clustering. *IEEE Access*, 9, 64346–64357.  
734 <https://doi.org/10.1109/ACCESS.2021.3075582>
- 735 *JPL, Mars Exploration Rovers*. (2021). <http://marsrovers.jpl.nasa.gov/home/index.html>
- 736 Kingma, D. P., & Ba, J. L. (2015). Adam: A method for stochastic optimization. *3rd International*  
737 *Conference on Learning Representations (ICLR)*.
- 738 Lin, Y., Lee, Y., & Wahba, G. (2002). Support vector machines for classification in nonstandard  
739 situations. *Machine Learning*, 46, 191–202. <https://doi.org/10.1023/A:1012406528296>
- 740 Lu, D., & Weng, Q. (2007). A survey of image classification methods and techniques for improving  
741 classification performance. *International Journal of Remote Sensing*, 28(5), 823–870.  
742 <https://doi.org/10.1080/01431160600746456>
- 743 Manduchi, R., Castano, A., Talukder, A., & Matthies, L. (2005). Obstacle detection and terrain  
744 classification for autonomous off-road navigation. *Autonomous Robots*, 18(1), 81–102.  
745 <https://doi.org/10.1023/B:AURO.0000047286.62481.1d>
- 746 Nampoothiri, M. G. H., Vinayakumar, B., Sunny, Y., & Antony, R. (2021). Recent developments in  
747 terrain identification, classification, parameter estimation for the navigation of autonomous  
748 robots. *SN Applied Sciences*, 3, 480. <https://doi.org/10.1007/s42452-021-04453-3>
- 749 *Nasa, Mars 2020*. (2020). <https://mars.nasa.gov/mars2020/>
- 750 Ocón, J., Dragomir, I., Coles, A., Green, A., Kunze, L., Marc, R., Perez, C. J., Germa, T., Bissonnette,  
751 V., Scalise, G., Foughali, M., Kapellos, K., Dominguez, R., Cordes, F., Paar, G., Reina, G., &  
752 Kisdi, A. (2020). Ade: Autonomous Decision Making in Very Long Traverses. *15th*  
753 *International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-*  
754 *SAIRAS '20)*.
- 755 Otsu, K., Ono, M., Fuchs, T. J., Baldwin, I., & Kubota, T. (2016). Autonomous Terrain Classification  
756 with Co-and Self-Training Approach. *IEEE Robotics and Automation Letters*, 1(2), 814–819.  
757 <https://doi.org/10.1109/LRA.2016.2525040>

- 758 Reina, G., Leanza, A., & Messina, A. (2020). Terrain estimation via vehicle vibration measurement  
759 and cubature Kalman filtering. *Journal of Vibration and Control*, 26(11–12), 885–898.  
760 <https://doi.org/10.1177/1077546319890011>
- 761 Reina, G., Milella, A., & Galati, R. (2017). Terrain assessment for precision agriculture using vehicle  
762 dynamic modelling. *Biosystems Engineering*, 162, 124–139.  
763 <https://doi.org/10.1016/j.biosystemseng.2017.06.025>
- 764 Tai, L., Li, S., & Liu, M. (2017). Autonomous exploration of mobile robots through deep neural  
765 networks. *International Journal of Advanced Robotic Systems*, 14(4).  
766 <https://doi.org/10.1177/1729881417703571>
- 767 Vapnik, V. N. (2013). The Nature of Statistical Learning Theory. In *Springer science & business*  
768 *media* (Second). <https://doi.org/10.1007/978-1-4757-2440-0>
- 769 Vulpi, F., Milella, A., Marani, R., & Reina, G. (2021). Recurrent and convolutional neural networks  
770 for deep terrain classification by autonomous robots. *Journal of Terramechanics*.  
771 <https://doi.org/10.1016/j.jterra.2020.12.002>
- 772 Wellhausen, L., Dosovitskiy, A., Ranftl, R., Walas, K., Cadena, C., & Hutter, M. (2019). Where  
773 should i walk(Predicting terrain properties from images via self-supervised learning. *IEEE*  
774 *Robotics and Automation Letters*, 4(2), 1509–1516. <https://doi.org/10.1109/LRA.2019.2895390>
- 775 Zhao, Q., & Fränti, P. (2014). WB-index: A sum-of-squares based index for cluster validity. *Data &*  
776 *Knowledge Engineering*, 92, 77–89. <https://doi.org/10.1016/j.datak.2014.07.008>
- 777