Towards Intrusion Detection of Previously Unknown Network Attacks

Saif Alzubi[1], Frederic Stahl[1,2], Mohamed Medhat Gaber[3,4]
[1]Department of Computer Science, University of Reading, Reading, UK
[2]Marine Perception Research Department,German Research Center for Artificial Intelligence (DFKI), Oldenburg, Germany
[3]School of Computing and Digital Technology, Birmingham City University, UK
[4]Faculty of Computer Science and Engineering, Galala University, Galala City 43511, Egypt

KEYWORDS

Anomaly Detection; Network Intrusion Detection; Unsupervised algorithms; Emsemble Learning

ABSTRACT

Advances in telecommunication network technologies have led to an ever more interconnected world. Accordingly, the types of threats and attacks to intrude or disable such networks or portions of it are continuing to develop likewise. Thus, there is a need to detect previously unknown attack types. Supervised techniques are not suitable to detect previously not encountered attack types. This paper presents a new ensemble-based Unknown Network Attack Detector (UNAD) system. UNAD proposes a training workflow composed of heterogeneous and unsupervised anomaly detection techniques, trains on attack-free data and can distinguish normal network flow from (previously unknown) attacks. This scenario is more realistic for detecting previously unknown attacks than supervised approaches and is evaluated on telecommunication network data with known ground truth. Empirical results reveal that UNAD can detect attacks on which the workflows have not been trained on with a precision of 75% and a recall of 80%. The benefit of UNAD with existing network attack detectors is, that it can detect completely new attack types that have never been encountered before.

INTRODUCTION

The Internet has become a part of our daily lives with billions of active users. New types of network attacks keep emerging, and there is a need to detect novel attacks without prior knowledge. Yet many Data Mining approaches to detect network attacks are supervised and are only suitable for detecting previously known attack types. There is a need for more exploration of unsupervised approaches as these approaches typically suffer from many false positives [1], a low precision or recall in detecting attacks, and some works are based on older attack types. Hence, the motivation of the paper is to fill this gap by developing the unsupervised ensemble-based Unknown Network Attack Detector (UNAD).

This paper first explores several unsupervised algorithms with respect to precision, recall, F1-Score for their suitability to be included as part of UNAD, namely the Local Outlier Factor (LOF) [2], Isolation Forest (iForest) [3] and Elliptic Envelope [4]. For this exploration, the CICIDS2017 [5] dataset is used. CICIDS2017 comprises 14 attack types, some of which emerged in recent years. Next, the paper proposes UNAD as a composition of some of the evaluated anomaly detecting methods as base learners (LOF and iForest). The reason for choosing an ensemble approach here is that ensemble approaches tend to improve the average accuracy over any member of the ensemble and reduce overfitting [6].

The contributions of the paper are (1) an experimental evaluation of the suitability of unsupervised anomaly detection methods for unknown attack detection; (2) a new heterogeneous unsupervised ensemble technique termed UNAD capable of detecting new previously unseen attack types; and (3) an experimental evaluation showing that UNAD it is capable of achieving high accuracy, precision and recall for detecting unknown attack types, and outperforms its standalone base learners.

Lastly the paper provides an outlook on ongoing and future research with respect to UNAD and also provides concluding remarks.

RELATED WORK

Supervised data mining approaches for Intrusion Detection Systems tend to achieve high accuracy, recall and precision such as [7], [8], [9]. However, they are not suitable for detecting unknown attacks types. Hence, unsupervised techniques have been explored, such as the Intrusion Detection System proposed in [10] based on One-class SVM. However, One-class SVMs tend to have a high computational overhead [11] and thus are not suitable for high-speed network traffic flow. The authors of [1] proposed an unsupervised ensemble model based Intrusion Detection System which achieved relatively high recall and precision. Yet both aforementioned unsupervised approaches have been trained and evaluated on relatively old datasets comprising none of the attack types that emerged over the last 10 years. More recently, iForest [3] was used in [12] to detect abnormal user behaviour on payroll access logs. Ensemble methods have also been used for insider threat detection such as in [13]. The authors of [14] used LOF to detect network attacks as anomalies. However, their study was conducted almost 10 years ago, and it is not clear if it still holds on recent attack types. Elliptic Envelopes [4], another unsupervised anomaly detection method, it has been used by the authors of [15] to detect Injection Attacks in Smart Grid Control Systems.

The research presented in this paper develops a new ensemble learner and workflow termed UNAD for unknown attack detection. Unlike previous ensemble learners for network intrusion/attack detection, UNAD integrates a heterogeneous set of standalone anomaly detection methods and improves upon their accuracy, precision and recall. Furthermore, UNAD is applied on recent data which contains more recent attack types. A problem with training models for unknown attacks is privacy issues. Therefore, publicly available synthetic benchmark datasets such as KDD Cup 99 [16], NSL-KDD[17], Kyoto 2006+[18], UNSW-NB15[19] and CICIDS2017 [5] can be used to mitigate privacy issues. The work presented in this paper uses CICIDS2017 as it contains more recent attack types.

UNAD BASE ANOMALY DETECTION METHOD SELECTION

In order to build the UNAD ensemble learner anomaly detection methods need to be selected as base learners. In total, 4 different kinds of anomaly detection methods that have previously been applied for similar applications to network attack detection (see RELATED WORK section) were considered. The considered techniques are One-Class SVM [20], iForest [3], LOF [2] and Elliptic Envelope [4]. The One-Class SVM method was ruled out early in the selection process since it is unsuitable for fast network flows due to its high computational demand [11]. The remaining three algorithms were experimentally optimised on the CICIDS2017 dataset and subsequently evaluated for their inclusion in the UNAD ensemble.

Experimental Setup

Evaluation Metrics

The metrics used to evaluate base learners are precision, recall and F1-Score. In UNAD, precision is equivalent to the portion of true positive attacks of all detections and recall is equivalent to the portion of attacks detected from all attacks present in the network flow. A high precision is equally important as detecting the majority of attacks. This is because false positive attack detections may trigger expensive actions to counter a non-existing threat. Since precision and recall are both equally important in this application, the base learners have been selected based on the F1-Score, which is the harmonic mean between precision and recall. An alternative measure to use instead of F1-Score could have been ROC_AUC; however, ROC_AUC measure is more reliable on balanced data and the ratio of benign data to data comprising attacks is 3:1 in the test set and validation set.

Dataset and Pre-Processing

For evaluating the algorithms, CICIDS2017 [5] dataset is used. CICIDS2017 is a publicly available benchmark dataset generated by the Canadian Institute for Cybersecurity, it covers five day, consists of 84 features, about 3 million data instances and covers 14 attack types including newer types that emerged in recent years. For generating the dataset [5] a complete network topology was created including Modem, Firewall, Switches, Routers. In addition, nodes in the network comprised various operating systems such as Windows, Ubuntu and Mac, all using commonly available protocols such as HTTP, HTTPS, FTP, SSH and email protocols. Table I summarises the number attacks per type and benign data instances in CICIDS2017.

TABLE I: CICIDS2017 overall traffic type distribution

Traffic Type	**Count**
Benign	2,358,036
DoS Hulk	231,073
Port Scan	158,930
DDoS	41,835
DoS GoldenEye	10,293
FTP Patator	7,938
SSH Patator	5,897
DoS SlowLoris	5,796
DoS SlowHTTPTest	5,499
Botnet	1,966
Web Attack: Brute Force	1,507
Web Attack: XSS	625
Infiltration	36
Web Attack: SQL Injection	21
HeartBleed	11
Total	**2,829,463**

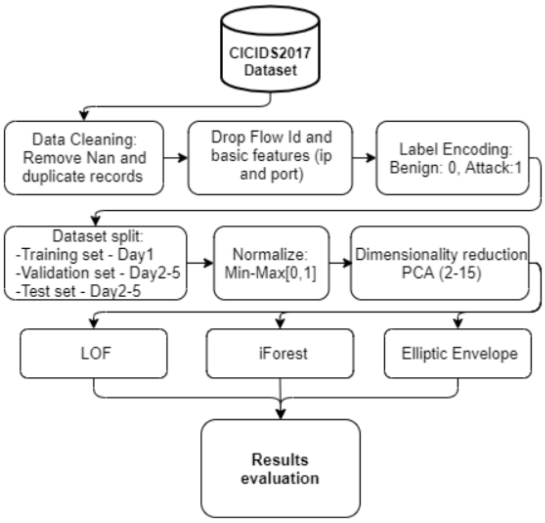
CICIDS2017 Dataset
Data Cleaning: Remove Nan and duplicate records
Drop Flow Id and basic features (ip and port)
Label Encoding: Benign: 0, Attack:1
Dataset split: -Training set - Day1 -Validation set - Day2-5 -Test set - Day2-5
Normalize: Min-Max[0,1]
Dimensionality reduction PCA (2-15)
LOF
iForest
Elliptic Envelope
Results evaluation

Fig. 1. Experimental workflow

All experiments were implemented in Python 3.6 using Google Colaboratory. The dataset was preprocessed before application of the anomaly detection algorithms. The pre-processing workflow is depicted in Figure 1. The first step was data cleaning which comprises removing missing and NaN values, as the used algorithms are designed for numerical data only. Moreover, duplicated records were removed to maintain data quality and avoid biased results. This step is followed by dropping out some features that could affect the model's performance; for instance, ID features were removed as they do not have discriminatory value with respect to attacks. Next features containing IP addresses were also removed as attackers often spoof their email addresses to avoid IP filtering systems [8]. Finally, features representing port information were removed as they cause models to overfit towards socket information [21]. Next, the categorical text in the Label feature was converted to numeric form. Hence, the label for all attack types were converted to 1 and for benign instances to 0. This is because anomaly detection methods are essentially binary classification methods since they distinguish normal data (i.e. benign) versus anomalies (i.e. attacks). After pre-processing, the dataset was split into training, validation, and test sets. Assuming that there is no prior knowledge about the network attacks, the models will be trained only on benign flow. Thus, data from the first day was used to train the model which comprises 529,445 normal data instances (about 19% of the entire dataset). The remaining four-day dataset, which contains 2,298,225 data instances of both attacks and benign flow, were split for validation and testing (50% each). All data instances were normalised between 0-1 using min-max scaling to reduce inductive bias while keeping the shape of the original data distribution. For the experiments, Principal Component Analysis (PCA) is used. PCA has been applied in the Intrusion Detection area since it only requires a few parameters of the principal components to be managed for future detections and most importantly, the statistics can be estimated in a short amount of time during the detection stage, which enables real-time usage of PCA [22], [23].

Evaluation of Anomaly Detection Algorithms as Base Learners for UNAD

The anomaly detection models were learned from the training data (comprising only benign network flow), and the validation data (including all types of attacks) was used to find the best combinations of hyperparameter to maximise F1-Score. For hyperparameter tuning, various Principal Components (PCs) were considered (2-15 PCs) to reduce the data's dimensionality.

Local Outlier Factor (LOF)

LOF detects local outliers by comparing the local density of an object to its adjacent neighbours. LOF considers an object as an outlier if the average of the local reachability density of that object is lower than the local reachability density of its adjacent neighbours [2]. LOF's main advantage is detecting local and neighbouring outliers to data instances in very large datasets with heterogeneous densities [24], [25]. Therefore, for massive network traffic, LOF is expected to play a significant role in detecting attacks. Accordingly, LOF is evaluated here as a potential part of the proposed ensemble-based UNAD. The LOF module from scikit-learn [26] was used. The hyperparameters are contamination and n_neighbours. Contamination is the proportion of the outliers expected in the dataset ranging from 0 to 0.5. We assumed no knowledge about the proportion of outliers constituting non-attacks in the training data. The hyperparameters were tuned using various combinations of values for the contamination value. It was tuned from 0.01 to 0.5 in steps of 0.01. The value of n_neighbours was selected within a range of 5 to 50 in steps of 5. Once the hyperparameters were optimised and the best combination was determined, they were applied to the test set for every number of PCs ranging between 2-15.

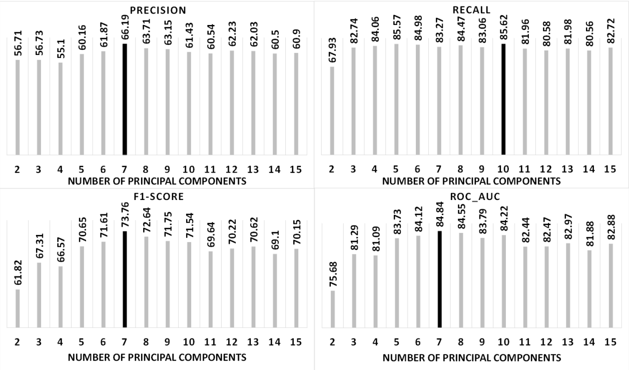

Fig. 2. Summary of Experimental Results expressed in percentages for the LOF based workflow

Fig. 2 shows the best results for each number of PCs used. For each number of PCs, always the best setting of contamination and the number of nearest neighbours is displayed. The figure shows that the highest precision and F1-Score was achieved for 7 PCs (with contamination parameter 0.07 and 30 neighbours), while the highest recall was observed for 10 PCs (with contamination parameter 0.08 and 35 neighbours). Based on the F1-Score, the optimal number of PCs for LOF is 7.

Isolation Forest

iForest consists of a random trees forest that keeps portioning all instances until they are fully separated. Moreover, iForest assumes that anomalies are expected to be split in early partitioning; therefore, instances with shorter path lengths are very likely to be anomalies [27]. iForest provides low linear time-complexity with a low memory requirement, making it ideal for detecting network attacks in a fast and timely manner. Furthermore, iForest can deal with high dimensional data with unrelated attributes [27]. Hence, making it perfect to be integrated in the proposed UNAD ensemble. iForest implementation from scikit-learn [26] was used. The hyperparameters here are contamination factor, n_estimators (number of trees) and max_samples. The contamination parameter is the same as for LOF. We assumed no knowledge about the proportion of outliers constituting non-attacks in the training data. The hyperparameters were optimised using various combinations of values for the contamination value. It was tuned from 0.01 to 0.5 in steps of 0.01. The number of n_estimators was selected from 50 to 450 in steps of 50.

Regarding the max_samples parameter, which selects the portion of the training data for each base estimator, a proportion settings of 25%, 50%, 75% and 100% were used in addition to the default setting of 256 samples. Concerning other parameters, max_features parameter which controls the number of features to be extracted from the dataset to train each estimator [26], it was set to its default values (1.0) to draw all features to train the estimators, and the random_state parameter was set to a fixed number (42) for results reproducibility. Once the hyperparameters were optimised and the best combination was determined, they were applied to the test set for every number of PCs ranging between 2-15.

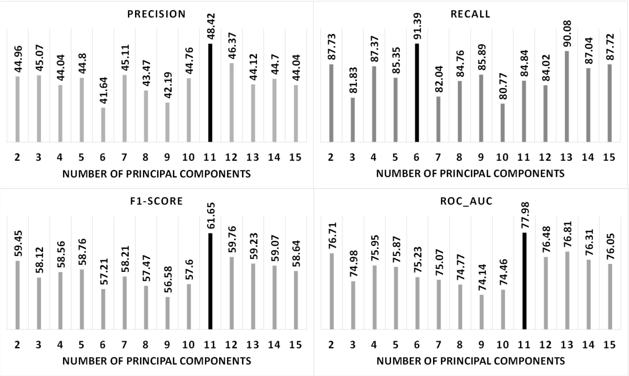

Fig. 3. Summary of Experimental Results expressed in percentages for the iForest based workflow

Fig. 3 shows the best results of iForest for each number of PCs used. For each number of PCs, always the best stetting of contamination and the number of nearest neighbours are displayed. The figure shows that the highest precision and F1-Score was obtained using 11 PCs (with contamination parameter of 0.24, 400 estimators and 25% max_samples), while the highest recall was observed using 6 PCs (with contamination parameter of 0.43, 200 estimators and default setting (256) max_samples). Based on the F1-Score the optimal number of PCs for iForest was at 11.

Elliptic Envelope

Elliptic Envelope detects outliers on multivariate Gaussian distributed datasets. Elliptic Envelope creates and fits an ellipse around the centre of a group of data instances using the Minimum Covariance Determinant. Hence, any data instance that is outside the ellipse is considered an outlier [4]. As the method was developed for Gaussian distributed datasets, it may not perform well on data streams, because the distribution a data stream can change over time due to concept drift. However, since the method has a low computational complexity, and is readily available in scikit-learn [26] it has been evaluated as a potential base learner for UNAD. The Elliptic Envelope contamination hyperparameter is the same as for LOF and iForest. Again, we assumed no knowledge about the proportion of outliers constituting non-attacks in the training data. The contamination parameter value was set ranging from 0.01 to 0.5 in steps of 0.01. Once the contamination parameter was optimised and its best value determined, it was applied to the test set for every number of PCs ranging between 2-15.

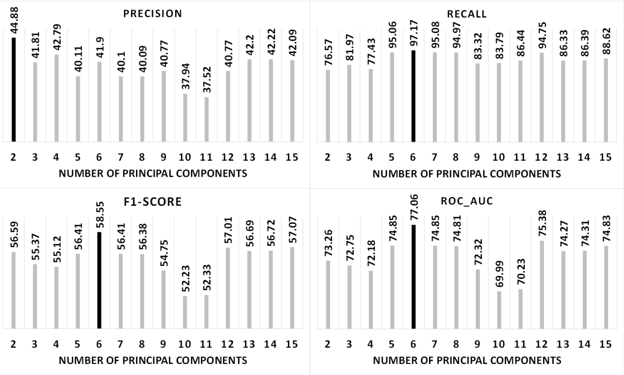

Fig. 4. Summary of Experimental Results expressed in percentages for the Elliptic Envelope based workflow

Fig. 4 depicts the Elliptic Envelope results for each number of PCs used. The figure shows that the highest recall and F1-Score was seen for 6 PCs (with contamination parameter of 0.44), while the highest precision was observed also for 6 PCs (with contamination parameter of 0.29). Based on the F1-Score and assuming equal importance of precision and recall, Elliptic Envelope's best setting was thus at 6 PCs, with contamination parameter of 0.44.

Conjectures and Selection of UNAD Base Learner Types

Although the anomaly detector candidates have been optimised with F1-Score as a target, ROC_AUC was included in the evaluation metrics as well since it is frequently used in anomaly detection literature. Interestingly in all cases using ROC_AUC rather than F1-Score would have lead the same outcome.

With respect to base anomaly detector selection for UNAD, LOF and iForest have been chosen. The reason for choosing LOF is that it achieves a relatively good F1-Score at 7 PCs on its own with 74% and a relatively high recall with 83% for 7 PCS. The precision of LOF is moderate with 66% at 7 PCs. The metrics for iForest are similar, but a bit more extreme. F1-Score is moderate at 61% for 11 PCs. the recall is high at 85% using 11 PCs, yet precision is relatively low with 48%, meaning that about half the anomaly detections are false alarms. Elliptic Envelope achieves the lowest F1-Score of all anomaly detectors with 59% at 6 PCs and even lower precision than iForest at 6 PCs with 42%. It has the highest recall though, with 97%. Since Elliptic Envelope achieves a very low precision, the technique is likely to be counterproductive in the UNAD ensemble and hence is excluded.

UNSUPERVISED ENSEMBLE LEARNER ARCHITECTURE FOR UNKNOWN ATTACK DETECTION

Based on the preliminary results discussed in the previous section, the UNAD ensemble was developed using of iForest and LOF as base learners, excluding Elliptic Envelop. The UNAD ensemble is depicted in Figure 5.

The dataset is cleaned the same way as described in the previous section, normalised and then two versions of the dataset are produced each projected on the best number of PCs for LOF (7 PCs) and iForest (11 PCs) respectively as determined in the experiments outlined in the previous section. Diversity among each type of base learner is created through bagging. For each base learner, bagging is applied on the 529,445 benign data instances of day one. In order to improve the stability and predictive performance of a composite learner [28], bagging was first introduced by Breiman [29]. It involves random sampling of the data instances with replacement. Each data instance is randomly selected whether to be in the sample or not. The size of the sample is equal to that of the original number of data instances. This suggests that some training instances may appear more than once in the same sample set, and some may not be included at all.

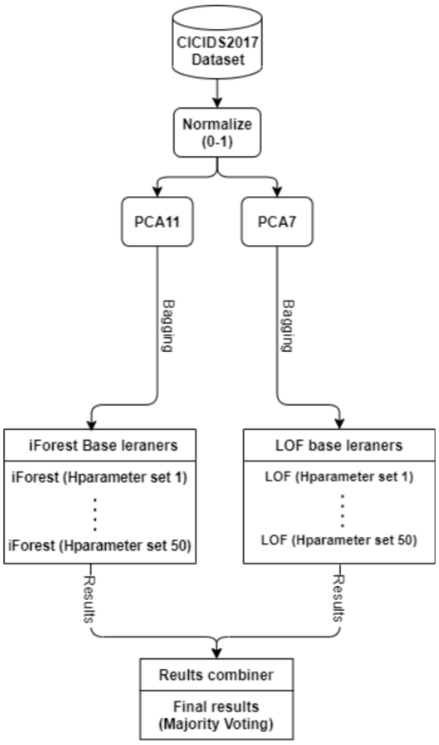

Fig. 5. Proposed UNAD workflow

UNAD induces 50 LOF and 50 iForest base learners, all of which are likely to be different since each has been induced on a separate bootstrap sample. UNAD further diversifies the base learners by randomly choosing a set of parameter values for the number of Nearest Neighbours and a contamination factor. Here parameter values from the top 3 best performing instances of LOF for 7 PCs were considered. These are:

- **Nearest Neighbours**: 25, 30 and 40
- **Contamination**: 0.06, 0.07 and 0.08

Concerning the iForest base learners, similar to the LOF base learners, UNAD chooses randomly the best parameter values from the iForest preliminary experiments with 11 PCs. These are in particular:

- **Number of estimators**: 150, 350 and 400.

The contamination parameter and the max samples parameters for iForest were identical for all top three instances in the preliminary experiments with 11 PCs. Hence, UNAD uses contamination 0.24 and 25% max samples for all iForest instances.

To detect anomalies, UNAD uses a majority voting scheme of all 100 base anomaly detector instances. There is an equal vote per base learner instance and per classification (benign or attack). Please note that due to equal voting and even number of base learners tie breaks are possible. The same number of base learners for LOF and iForest has been chosen to avoid bias towards one type of base learner, hence there is an even number of base learners. Currently, tie breaks are analysed by a human analyst, since they represent an uncertainty of the system. In the *ONGOING and FUTURE WORK* Section we consider reducing the number of tie breaks through a weighted voting mechanism.

EXPERIMENTAL EVALUATION

The UNAD learner was applied on the CICIDS2017 dataset as a case study. The data is pre-processed as already described in Section *UNAD BASE ANOMALY DETECTION METHOD SELECTION*. The dataset attack types and benign distribution is highlighted in Table I. Data from day one of the network flow was used as training data. These 529,445 instances comprised only benign cases. The test set comprising 1,149,112 instances was used to evaluate UNAD. The test set comprised instances with all attack types and also benign data instances.

TABLE II: LOF, iForest and UNAD results comparison in %

Method Measure(%)	**LOF**	**iForest**	**UNAD**
Accuracy	86	74	87
Precision	66	48	71
Recall	83	85	80
F1-Score	74	61	75
ROC_AUC	85	78	85

Table II depicts the EXPERIMENTAL results of UNAD compared with the standalone LOF and iForest results. Although the UNAD's recall was slightly lower than in standalone LOF and iForest algorithms, the precision was considerably improved and also there is some improvement of the F1-Score. The ROC_AUC results are the same compared with the best standalone competitor, and accuracy has slightly improved. However, accuracy and ROC_AUC are not considered a good evaluation metric since the data is imbalanced 3:1 in favour of benign data. Thus, F1-Score is considered a suitable metric, since it describes how well attacks have been detected in terms of true positive detections and portion of overall attacks being detected. It can be seen that F1-Score has improved due to a considerable improvement of precision.

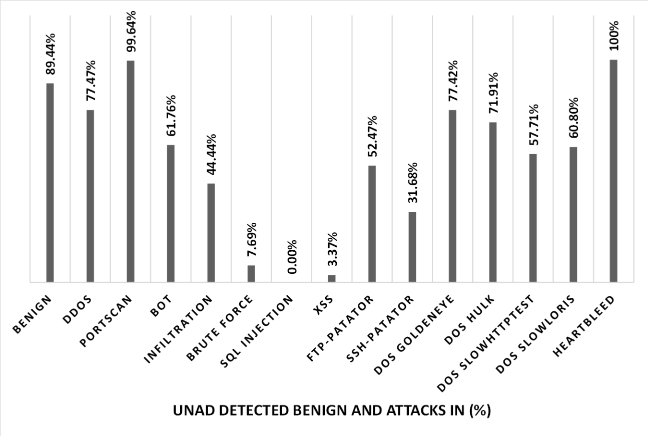

Fig. 6. Summary of UNAD detected benign and attacks

Figure 6 illustrates the percentage of identified benign cases and detected attacks using the UNAD system. UNAD was able to detect all the heartbleed attacks and almost all the portscan attacks (99.64%). UNAD was also able to identify 89.44% of the benign flow. DDoS and DoS detection rate were between 77.47% and 71.91% expect for DoS Slowloris and DoS Slowhttptest, which were 60.8% and 57.71% respectively. Attacks under the Web Attack category were the least well detected attacks with 7.69% for Brute Force, 3.37% for XSS and none of the SQL Injection attacks were detected. One can see that although there is overall a high recall / F1-Score the proportion of identified attacks varies by a large amount from attack type to attack type, yet all attacks, except Injections, are represented in the positive attack detections. However, considering the precondition that UNAD has never seen any of the attack types in the test set, it performs relatively well finding most attacks with a high precision and also almost all attack types are represented. In the *ONGOING WORK* section an approach to improve upon the recall of some attack types is briefly discussed.

TABLE III: Traffic type instances abstained from detection

Type	Count	Percentage (%)
BENIGN	151663	17.4
DDoS	3272	5.1
FTP-Patator	1546	38.9
DoS Slowhttptest	1160	42.1
DoS slowloris	868	29.9
DoS Hulk	824	0.7
DoS GoldenEye	789	15.3
Web Attack: Brute Force	386	51.2
SSH-Patator	671	22.8
PortScan	252	0.3
Web Attack: XSS	143	43.9
Bot	12	1.2
Infiltration	8	44.4
Web Attack: Sql Injection	3	30
Total	**164597**	**14.3**

Since UNAD abstains from detection when uncertain, there are also 161,597 test instances for which UNAD flagged that it was uncertain. How these traffic instances are composed is depicted in Table III. In the next section an approach to mitigate abstaining is briefly discussed.

ONGOING WORK

It was observed in the experimental evaluation that although UNAD did perform with a high F1-Score and precision on detected attacks, two limitations surfaced: (1) a low proportion of some attack types were detected and (2) abstaining from classifying some test instances.

With respect to limitation (1), a supervised adaptive component alongside UNAD is currently being developed that augments UNAD by training on new attack types previously detected by UNAD. Thus, once a new threat has been identified UNAD actively tries to further improve the detection of this particular type of attack. With respect to (2), a weighted majority voting is being considered since it is expected to improve UNAD's performance in general and lower the risk of tie breaks. For this a detection score is currently being developed (calculated on the *out of bag* sample from the bagging procedure). This detection score will be used to weight votes of UNAD base learners and thus reduce the possibility of tie breaks and further improve F1-Score. Tie-breaks resulting in abstaining from detection attempts may still occur; however, a lower number of abstained instances is expected to be more feasible to be examined manually by human analysts.

In addition, ongoing work also includes investigating why SQL injection are not well detected and which type of attacks are causing the ensemble's lower recall.

CONCLUSIONS

The paper discussed the need for unsupervised machine learning techniques to detect network attacks, because new types of network attacks constantly emerge. However, if an attack-type is and previously unknown, a supervised model is generally not capable of detecting such an attack sufficiently. Hence, this paper explores experimentally various anomaly detection methods for their detection capabilities of recent unknown network attacks. Based on this experimental evaluation the authors proposed a heterogeneous ensemble-based Unknown Network Attack Detection (UNAD) system which is composed of some of the evaluated anomaly detection methods, in order to improve precision and recall of unknown attack detection compared with standalone anomaly detection methods. UNAD is evaluated on the CICIDS2017 dataset, which does not pose any privacy issues and comprises recent attack types. The ensemble achieved a high precision and F1-Score and generally outperformed its standalone base anomaly detectors. Ongoing and future work comprise an improved voting strategy for base learners to further improve UNAD's performance and reduce tie breaks. Also an augmentation of UNAD is considered which provides a simultaneously running adaptive parallel supervised learner, which trained / adapted if a new, previously unknown attack, has been identified by UNAD.

References

[1] W. Chen, F. Kong, F. Mei, G. Yuan, and B. Li, "A novel unsupervised anomaly detection approach for intrusion detection system," in *ieee 3rd international conference on big data security on cloud (bigdatasecurity), ieee international conference on high performance and smart computing (hpsc), and ieee international conference on intelligent data and security (ids)*, pp. 69–73, 2017.

[2] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying density-based local outliers," in *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, SIGMOD '00, (New York, NY, USA), p. 93–104, ACM, 2000.

[3] F. T. Liu, K. M. Ting, and Z. Zhou, "Isolation forest," in *2008 Eighth IEEE International Conference on Data Mining*, pp. 413–422, 2008.

[4] P. J. Rousseeuw and K. V. Driessen, "A fast algorithm for the minimum covariance determinant estimator," *Technometrics*, vol. 41, p. 212, Aug 1999. 3.

[5] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization.," in *4th International Conference on Information Systems Security and Privacy (ICISSP)*, pp. 108–116, 2018.

[6] O. Sagi and L. Rokach, "Ensemble learning: A survey," *WIREs Data Mining and Knowledge Discovery*, vol. 8, no. 4, p. e1249, 2018.

[7] H. Zhang, S. Dai, Y. Li, and W. Zhang, "Real-time distributed-random-forest-based network intrusion detection system using apache spark," in *2018 IEEE 37th International Performance Computing and Communications Conference (IPCCC)*, pp. 1–7, 2018.

[8] C. B. Freas, R. W. Harrison, and Y. Long, "High performance attack estimation in large-scale network flows," in *2018 IEEE International Conference on Big Data (Big Data)*, pp. 5014–5020, 2018.

[9] Y. Zhou, G. Cheng, S. Jiang, and M. Dai, "Building an efficient intrusion detection system based on feature selection and ensemble classifier," *Computer Networks*, vol. 174, p. 107247, 2020.

[10] M. Zhang, B. Xu, and J. Gong, "An anomaly detection model based on one-class svm to detect network intrusions," in *2015 11th International Conference on Mobile Ad-hoc and Sensor Networks (MSN)*, pp. 102–107, 2015.

[11] I. Razzak, K. Zafar, M. Imran, and G. Xu, "Randomized nonlinear one-class support vector machines with bounded loss function to detect of outliers for large scale iot data," *Future Generation Computer Systems*, vol. 112, pp. 715 – 723, 2020.

[12] L. Sun, S. Versteeg, S. Boztas, and A. Rao, "Detecting anomalous user behavior using an extended isolation forest algorithm: An enterprise case study," 2016.

[13] D. Haidar and M. M. Gaber, "Adaptive one-class ensemble-based anomaly detection: an application to insider threats," in *2018 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–9, IEEE, 2018.

[14] A. Lazarevic, L. Ertoz, V. Kumar, A. Ozgur, and J. Srivastava, "A comparative study of anomaly detection schemes in network intrusion detection," in *Proceedings of the 2003 SIAM international conference on data mining*, pp. 25–36, SIAM, 2003.

[15] M. Ashrafuzzaman, S. Das, A. A. Jillepalli, Y. Chakhchoukh, and F. T. Sheldon, "Elliptic envelope based detection of stealthy false data injection attacks in smart grid control systems," in *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1131–1137, 2020.

[16] KDD99, "The UCI KDD Archive." http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html. Accessed: 18.06.2020.

[17] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the kdd cup 99 data set," in *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, pp. 1–6, 2009.

[18] J. Song, H. Takakura, Y. Okabe, M. Eto, D. Inoue, and K. Nakao, "Statistical analysis of honeypot data and building of kyoto 2006+ dataset for nids evaluation," in *Proceedings of the First Workshop on Building Analysis Datasets and Gathering Experience Returns for Security*, BADGERS '11, (New York, NY, USA), p. 29–36, ACM, 2011.

[19] N. Moustafa and J. Slay, "Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set)," in *2015 Military Communications and Information Systems Conference (MilCIS)*, pp. 1–6, 2015.

[20] B. Schölkopf, R. Williamson, A. Smola, J. Shawe-Taylor, and J. Platt, "Support vector method for novelty detection," in *Proceedings of the 12th International Conference on Neural Information Processing Systems*, NIPS'99, (Cambridge, MA, USA), p. 582–588, MIT Press, 1999.

[21] O. Faker and E. Dogdu, "Intrusion detection using big data and deep learning techniques," in *Proceedings of the 2019 ACM Southeast Conference*, pp. 86–93, 2019.

[22] K. S. M. Shyu, S. Chen and L. Chang, "A novel anomaly detection scheme based on principal component classifier," in *Proceedings of the IEEE Foundations and New Directions of Data Mining Workshop, in conjunction with the Third IEEE International Conference on Data Mining (ICDM03)*, p. 172–179, 2003.

[23] S. Almotairi, A. Clark, G. Mohay, and J. Zimmermann, "A technique for detecting new attacks in low-interaction honeypot traffic," in *2009 Fourth International Conference on Internet Monitoring and Protection*, pp. 7–13, 2009.

[24] D. Pokrajac, A. Lazarevic, and L. J. Latecki, "Incremental local outlier detection for data streams," in *2007 IEEE Symposium on Computational Intelligence and Data Mining*, pp. 504–515, 2007.

[25] M. Salehi, C. Leckie, J. C. Bezdek, T. Vaithianathan, and X. Zhang, "Fast memory efficient local outlier detection in data streams," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 12, pp. 3246–3260, 2016.

[26] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[27] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation-based anomaly detection," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 6, no. 1, pp. 1–39, 2012.

[28] L. Rokach, "Ensemble-based classifiers," *Artificial Intelligence Review*, vol. 33, no. 1, pp. 1–39, 2010.

[29] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, p. 5–32, Oct. 2001.

SAIF ALZUBI is a PhD student in Computer Science at the University of Reading, UK. His main research interests are in Machine Learning and Data Mining. Mr Alzubi worked as an Information System Developer at the e-learning centre, and as a senior database programmer at the IT centre, University of Bahrain, Bahrain. He obtained his MSc degree in Computer Science in 2018 from Coventry University, UK and his BSc degree in Computer Science in 2009 from University of Jordan, Jordan.

FREDERIC STAHL is Senior Researcher at the German Research Center for Artificial Intelligence (DFKI). He has been working in the field of Data Mining for more than 15 years. His particular research interests are in (i) developing scalable algorithms for building adaptive models for real-time streaming data and (ii) developing scalable parallel Data Mining algorithms and workflows for Big Data applications. In previous appointments Frederic worked as Associate Professor at the University of Reading, UK, as Lecturer at Bournemouth University, UK and as Senior Research Associate at the University of Portsmouth, UK. He obtained his PhD in 2010 from the University of Portsmouth, UK and has published over 65 articles in peer-reviewed conferences and journals.

MOHAMED MEDHAT GABER Mohamed Gaber is a Professor in Data Analytics at Birmingham City University, and currently seconded as the Dean of the Faculty of Computer Science and Engineering at Galala University. He has published over 200 papers, co-authored 3 monograph-style books, and edited/co-edited 7 books on Artificial Intelligence. His work has attracted nearly six thousand citations, with an h-index of 40. According to the latest study conducted by Stanford University and Elsevier, and released in 2020, Mohamed is among the top 2% of the most cited scientists worldwide. Mohamed's research interests span many areas of Artificial Intelligence including, but not limited to: (1) ensemble learning, (2) learning from data streams, (3) medical image analysis, (4) natural language processing, (5) time series classification, and (6) deep learning.