ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

AI-based algorithm for intrusion detection on a real dataset

Alejandro Esteban Martínez[1],
David Esteban Martínez[1],
Bertha Guijarro-Berdiñas[1],
Amparo Alonso-Betanzos[1], Elena Hernández-Pereira[1], and Julio Hernández-Castro[2]

1- Universidade da Coruña
A Coruña - Spain

2- Universidad Politécnica de Madrid
Madrid - Spain

Abstract.
In the realm of cybersecurity, the detection of network intrusions stands as a paramount challenge, with ever-evolving threats demanding innovative solutions. This study delves into the application of diverse machine learning algorithms on a contemporary dataset (UGR'16) comprising real-world instances of intrusion in software systems. Specifically, several Machine Learning models (Outlier Detectors, Ensemble Methods, Deep Learning, and Conventional Classifiers) were tested and compared with previously reported results using a standard methodology. The obtained results reveal that the Ensemble Methods have been capable of improving the results from prior research. Particularly, the Extreme Gradient Boosting (XGBoost) algorithm offers better results than the original solution with Random Forest, with an AUC of 0.9218 as opposed to 0.8977, and more than four times as fast for the problem to solve.

1 Introduction

In the Information Era, our world has become more interconnected than ever, facilitated by the rapid transmission of information across vast distances in mere seconds, all thanks to the Internet. However, with technological advancements come complex challenges. The accessibility of the Internet remotely undermines traditional physical security measures, rendering them ineffective against cyber threats that exploit vulnerabilities in our layered technology. This includes legacy components, resistant to replacement, and susceptible to hacking.

One viable solution is implementing an Intrusion Detection System (IDS) capable of monitoring network traffic for suspicious activities which, upon detecting a potential attack, alerts the relevant parties, typically a system administrator.

This work explores cutting-edge technologies used or potentially employable to enhance IDS detection capabilities. Specifically, it develops an Artificial Intelligence (AI) approach for our system, trained using the UGR'16 dataset `https://nesg.ugr.es/nesg-ugr16/`. This dataset, curated in 2016 by researchers at Universidad de Granada (Spain), stands as one of the most comprehensive and contemporary network flow datasets publicly accessible. Our AI system follows a standardized methodology recently proposed [1] for evaluating

ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

Machine Learning (ML) models for Network IDSs, aiming to surpass existing results on the UGR'16 dataset [2]. The primary contributions of this work encompass: (1) Analysis of IDS development techniques using ML, (2) Proposition of an AI-based IDS trained with the UGR'16 dataset with state-of-the-art results, and (3) Comparison with existing state-of-the-art models.

2 State of the Art in Network Intrusion Detection Systems

Multiple past works have approached Network IDS using Machine Learning [3, 4] due to the need to cope with the development of new attacks. Still, these works hinge on the quality of the training data, which has continuously adapted to address the challenges posed by real-world network traffic and attacks [5, 6].

2.1 The UGR'16 Dataset

The UGR'16 dataset, introduced in 2017 by the University of Granada (Spain), caters specifically to IDS utilizing cyclostationary analysis. It represent significant progress in addressing the limitations of previous datasets, providing more realistic and diverse environments for evaluating Intrusion Detection System (IDS) methods. It leverages real-world network traffic data captured from a Tier 3 internet service provider. It is divided into a calibration set, which assists in building and calibrating normality models, and a test set which contains a mix of real and simulated attacks. The characteristics of the dataset and subsets can be seen in Table 1 and Table 2. Each element of the dataset can be classified as attack or normal data, and contains the following information: timestamps, flow duration, source and destination IP addresses, source and destination ports, protocol, flags, forwarding status, type of service, packets exchanged within the flow, and their corresponding byte counts.

Dataset	**Size**	**Time**
UGR'16	16.9 billion	4 months
Calibration	13 billion	100 days
Test	3.9 billion	1 month

Table 1: Characteristics of the different subsets that compose the UGR'16 dataset

2.2 Free Framework for Machine Learning (FF4ML)

AI systems and ML are widely adopted in IDSs today. However, as highlighted by other authors such as [7] or [8], there is a pressing need for a standardized methodology to compare different ML approaches over datasets, which is where Free Framework for Machine Learning Methodology (FF4ML) [1], presented in [7] comes. This methodology encompasses four distinct steps designed to extract crucial information from the UGR'16 dataset, transform it for consistent ML model training, and present the results in a format suitable for comparative

ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

Class	Count	Percentage from Total (%)
Background	811375	92.12%
Spam	27957	3.17%
Dos	4959	0.56%
Scan44	2417	0.27%
Blacklist	2374	0.27%
UDPScan	1005	0.11%
Botnet	961	0.11%
Scan11	566	0.06%

Table 2: Class Distribution from subsampled Test file

analysis with other models and previous state-of-the-art approaches. These steps are:(1) transforming raw data into model-friendly format using Feature as a Counter (FaaC), (2) selecting important features eliminating less important or redundant ones, (3) normalizing the numerical features for consistent model input, (4) performing hyper-parameter selection with Bayesian Search.
Using this methodology and three models, Multinomial Logistic Regression, Support Vector Machine (SVM) with Linear and Radial Basis Function kernels, and Random Forest, the study [7] evaluates these models performance, trained and tested with the UGR'16 Dataset.

Their evaluation exposes significant discrepancies in model accuracy, particularly across attack classes and performance metrics. While Random Forest and Support Vector Machine with Radial Basis Function kernel models exhibit strong performance, especially for synthetically generated attacks (DoS, Botnet, Scan), real-world attacks (Spam) and real network traffic (Background) present challenges due to their intricate patterns. Random Forest demonstrates robustness for unbalanced classifications, outperforming other models in terms of weighted averages of performance metrics.

3 Experimentation

In this work we propose experimentation with different models in hopes of improving the original results from [7]. Our used models are: (1) Conventional Classifiers: K-Nearest Neighbours (KNN) and Extreme Gradient Boosting (XGBoost), (2) Deep Learning models: Multilayer Perceptron (MLP), (3) Outlier Detectors: Isolation Forest (IF) and One-Class Support Vector Machine (OC-SVM), and (4) Ensemble Models: Voting and Stacking.

After a process of hyper-parameter tuning, using Bayesian optimization, we arrived at the following best hyper-parameters for each model, seen in Table 3.

Finally, Table 4 shows the results obtained for the models detailed above.

As it can be seen, the best performing models are by far XGBoost and MLP, but only XGBoost improves on the scores obtained with the original Random Forest model. However, both XGBoost and MLP are significantly faster than Random Forest, with training plus testing time of 173.18 and 108.87 seconds

ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

Model	Configuration
KNN	metric='euclidean', n_neighbors=3, weights='distance'
XGBoost	max_depth=6, n_estimators=1000
IF	bootstrap=True, contamination=0.0289, max_features=30, max_samples=272, n_estimators=316
OC-SVM	coef0=5.2869, degree=2, gamma=0.0023, kernel='poly', max_iter=24806, nu=0.3487, tol=6.5637e-05
MLP	n_layers= 2, layer_sizes=[49, 59],activation='relu', solver= 'sgd', alpha= 0.0001, max_iter=229

Table 3: Best configurations for all models

Model	**Precision**	**Recall**	**F1 Score**	**AUC**
KNN	0.8419	0.8417	0.8416	0.8260
XGBoost	**0.9291**	**0.9290**	**0.9289**	**0.9219**
IF	0.6001	0.6165	0.5692	0.5731
OC-SVM	0.7760	0.7323	0.7373	0.7556
MLP	0.8893	0.8933	0.8913	0.8962
Random Forest	0.9147	0.9095	0.9110	0.8977

Table 4: Weighted Performance for all models with best hyper-parameter configuration

as opposed to the almost 700 seconds it takes for Random Forest for the same problem. KNN performs well and fast, with the fastest time at 72 seconds, but has enough of a downgrade in performance to be rejected; also, the results make clear that using anomaly detection models for this type of problem is not the right approach, both IF and OC-SVM being the worst performing models for the problem. Based on these results, we utilized the three best-performing models to create the ensemble models, XGBoost, MLP with 2 hidden layers and the replicated Random Forest from the original study. The Voting Ensemble model combines these models with an uneven number of estimators to avoid draws in the process of voting, and soft voting, which uses the prediction probabilities, as is recommended for well-calibrated estimators. We used custom weights as they proved to be better than uniform, searching for weights using Bayesian Search, which unexpectedly assigned lower weight of 6 to XGBoost's decisions, even with it being the best-performing model, which could be explained due to the difference in predictions between the MLP and RF being beneficial to the final result, which where both assigned a weight of 7.

For our stacking ensemble, no hype-parameter optimization was needed, and we employed the default Logistic Regression model as our final estimator to give the predictions, as it is usually recommended not to change it, we also computed with and without passthrough, a process that adds the original input features to the set of predictions used for making the final decision. The results of the Ensemble models can be seen in Table 5, and in Figure 1 we can see a graphical

ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

comparison of the results from the best models in each group considered.

Model	P	R	F1	AUC
Voting Classifier with uniform weights	0.9298	0.9264	0.9277	0.9188
Voting Classifier with custom weights	0.9300	**0.9309**	0.9301	0.9209
Stacking without passthrough	0.9307	0.9284	0.9294	0.9224
Stacking with passthrough	**0.9301**	0.9308	**0.9303**	**0.9237**

Table 5: Weighted Performance for Voting Models

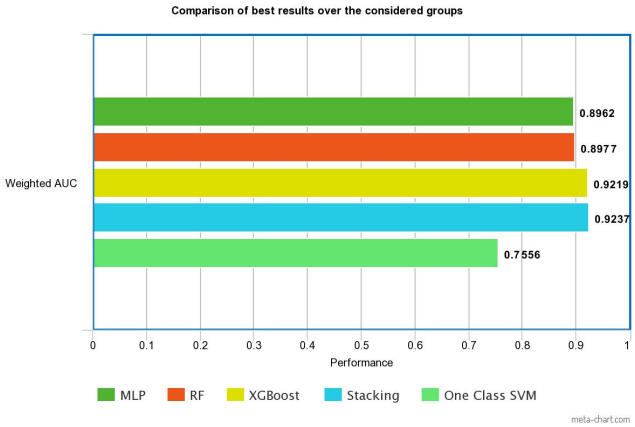

Fig. 1: Comparison of best results over the considered groups

These results are moderately better than with just XGBoost in the case of the Stacking model, and slightly worse with the Voting classifier. Since the improvements are not significant, following Occam's Razor principle, XGBoost is selected as better, due to being simpler and faster than the Ensemble models.

4 Conclusions

In this work, we have applied ML algorithms to improve the reported results of a Cyclo-Stationary Network Intrusion Detection System. For this we have used the UGR'16 Dataset [2] as a train and test set for our Machine Learning models and followed the methodology proposed in [7] to be able to compare our results and their previous results.

The original work's results were surpassed with the XGBoost model, with an AUC of 0.9219, and the Ensemble Methods, which used a combination of Random Forest, XGBoost and Multilayer Perceptron (as these 3 methods had the best results) for Voting and Stacking Classifiers, with AUCs of 0.9209 and 0.9237, against the AUC of 0.8977 of the original work with Random Forest. The best results are given by the Stacking Classifier,but followed closely by the XGBoost alone, and considering the time of training and following Occam's Razor

ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

principle, the XGBoost model is considered the best to classify network data and detect attacks as it has lower complexity than the Stacking Classifier.

Aknowledgements

CITIC, as a center accredited for excellence within the Galician University System and a member of the CIGUS Network, receives subsidies from the Department of Education, Science, Universities, and Vocational Training of the Xunta de Galicia. Additionally, it is co-financed by the EU through the FEDER Galicia 2021-27 operational program (Ref. ED431G 2023/01)

References

[1] UCADatalab. ff4ml: Free framework for machine learning. `https://github.com/ucadatalab/ff4ml`, 2022.

[2] Gabriel Maciá-Fernández, José Camacho, Roberto Magán-Carrión, Pedro García-Teodoro, and Roberto Therón. Ugr'16: A new dataset for the evaluation of cyclostationarity-based network idss. *Computers & Security*, 73:411–424, 2017.

[3] Shadi S M Aljawarneh, Monther Aldwairi, and Muneer Bani Yassein. Machine learning in intrusion detection systems: A survey. *International Journal of Advanced Computer Science and Applications*, 10(1):557–567, 2019.

[4] Muhammad Ahsan Ahmad, Fehaid Alqahtani, Awais Aziz Shah, Fadia Ali Khan, Mujeeb Ur Rehman Khan, Jan Sher Khan, Kathryn-Ann Tait, and Jawad Ahmad. Network intrusion detection system: A systematic study of machine learning and deep learning approaches. *IEEE Access*, 8:132894–132917, 2020.

[5] Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali A Ghorbani. Kdd cup 1999 data. In *Proceedings of the Second International Workshop on Security and Artificial Intelligence*, page 1–6, 2009.

[6] Romain Fontugne, Pierre Borgnat, Patrice Abry, and Kensuke Fukuda. Mawilab: Combining diverse anomaly detectors for automated anomaly labeling and performance benchmarking. *Proceedings of the ACM CoNEXT Conference (CoNEXT)*, 2010.

[7] Roberto Magán-Carrión, Daniel Urda, Ignacio Díaz-Cano, and Bernabé Dorronsoro. Towards a reliable comparison and evaluation of network intrusion detection systems based on machine learning approaches. *IEEE Access*, 8:112227–112247, 2020.

[8] Giovanni Apruzzese, Luca Pajola, and Mauro Conti. The cross-evaluation of machine learning-based network intrusion detection systems. *IEEE Transactions on Network and Service Management*, 19(4):5152–5169, dec 2022.