Which Algorithm can Detect Unknown Attacks? Comparison of Supervised, Unsupervised and Meta-Learning Algorithms for Intrusion Detection

Tommaso Zoppi*, Andrea Ceccarelli, Tommaso Puccetti, Andrea Bondavalli

Department of Mathematics and Informatics, University of Florence,Viale Morgagni 65, 50142 - Florence - Italy

Abstract

There is an astounding growth in the adoption of machine learners (MLs) to craft intrusion detection systems (IDSs). These IDSs model the behavior of a target system during a training phase, making them able to detect attacks at runtime. Particularly, they can detect known attacks, whose information is available during training, at the cost of a very small number of false alarms, i.e., the detector suspects attacks but no attack is actually threatening the system. However, the attacks experienced at runtime will likely differ from those learned during training and thus will be unknown to the IDS. Consequently, the ability to detect unknown attacks becomes a relevant distinguishing factor for an IDS. This study aims to evaluate and quantify such ability by exercising multiple ML algorithms for IDSs. We apply 47 supervised, unsupervised, deep learning, and meta-learning algorithms in an experimental campaign embracing 11 attack datasets, and with a methodology that simulates the occurrence of unknown attacks. Detecting unknown attacks is not trivial: however, we show how unsupervised meta-learning algorithms have better detection capabilities of unknowns and may even outperform classification performance of other ML algorithms when dealing with unknown attacks.

Keywords

Intrusion Detection, Machine Learning, Unsupervised, Meta-Learning, Zero-Day Attacks

1. Introduction

It is widely acknowledged that modern ICT systems such as industrial control systems [13], medical support systems [2], virtual environments [12], and the Internet of Things [1] can be the target of attackers [3], [17], [18]. There is significant evidence on the risk of cyberattacks, both in terms of the likelihood of being targeted and the cost and impact of a successful attack.

The number of computer security incidents has been steadily growing over the past few years: in 2021, SonicWall [11] reported an average of 28 million cyberattacks detected daily, with 140 000 of them being novel malware samples. Starting from 2020, the European Union Agency for Cybersecurity (ENISA) observed a spike in non-malicious incidents, most likely because the COVID-19 pandemic became a multiplier for human errors and system misconfigurations, and attributed them as the root cause for the majority of security breaches [10].

The consequence of a successful cyberattack (simply termed an *attack* from now on) may range [25] from confidentiality issues to availability reduction or the loss of sensitive data and thus integrity concerns. Importantly, security threats may also have a safety impact; for example, an attack that aims at making the automatic braking system of a vehicle unavailable may also have severe impacts on the health of the driver, the infrastructures and the surrounding environment. Consequently, systems must be conceptualized, designed, and implemented to ensure that appropriate security requirements are met. Among the possible countermeasures, the detection of ongoing attacks is typically included in the mandatory security requirements [17], [18].

Intrusion detection systems (IDSs) are well-known means to promptly detect attacks. Given a target system to protect, an IDS monitors its performance indicators: examples include memory usage [12], throughput of buses [13],

Ital-IA 2023: 3rd National Conference on Artificial Intelligence, organized by CINI, May 29-31, 2023, Pisa, Italy

EMAIL: {tommaso.zoppi, andrea.ceccarelli, tommaso.puccetti, bondavalli}@unifi.it

ORCID: 0000-0001-9820-6047, 0000-0002-2291-2428, 0000-0002-0297-2108, 0000-0001-7366-6530

active sessions [14], and system calls [19]. The set of monitored values (i.e., features) gathered by an IDS at a given instant is called a *data point*: collections of data points are typically collected in the form of tabular datasets. An IDS contains a machine learning (ML) algorithm that performs binary classification [7], [9] (i.e., it is a binary classifier) discerning between data points corresponding to an attack and data points corresponding to the normal behavior of a system. The ML algorithm undergoes a training phase in which it processes a training dataset: it learns a model, which at a later stage will be deployed in the production environment to detect attacks occurring at runtime.

Unfortunately, data points collected in the production environment may differ from the data points in the training dataset. This is a very frequent scenario for two reasons. First, systems are becoming increasingly more complex and dynamic, committing updates and reconfigurations. Consequently, training may quickly become obsolete [23] and reduce the effectiveness of the model learned. Second, during its operational life, a system may be targeted by attacks that were not known at training time, which we call *unknown attacks*. Unknown attacks are significant threats, with the same effect as zero days [3], , i.e., new attacks or variations of existing attacks that are specifically created to exploit new vulnerabilities. It is credible that during its life, a system will be the target of unknown attacks, [10], [11], [17]; therefore, IDSs must be prepared to deal with them to avoid major security issues.

This paper reviews classifiers for intrusion detection, evaluating their capability to detect unknown attacks. We organize classifiers into five categories: unsupervised (UNS), supervised (SUP), deep learning (DEEP), supervised meta-learning (META-SUP), and unsupervised meta-learning (META-UNS). UNS classifiers do not use labels during training, i.e., they are not aware of whether a data point is an attack. In contrast, SUP and DEEP classifiers need labeled data points during training. The SUP and DEEP classifiers are both supervised classifiers, and the latter uses deep neural networks. We apply this distinction because the literature on ML for tabular data shows different classification performance between deep neural networks and other supervised classifiers. Often, DEEP classifiers are considered to perform worse than SUP on tabular data [8], [22]. Last, META-SUP and META-UNS are supervised and unsupervised classifiers, respectively, that employ meta-learning: meta-learning uses knowledge acquired during base-learning episodes, i.e., meta-knowledge, to improve classification capabilities at the meta-level [15]. META-SUP classifiers require labeled data for training, while META-UNS classifiers do not.

We exercise a total of 47 classifiers on 11 public attack datasets, which we manipulate to simulate the occurrence of unknown attacks. Very briefly, the procedure is as follows. Some attacks are removed from the training datasets and used only for testing, which makes them unknown attacks. The whole procedure is repeated for all the attack categories and all the datasets. We analyze and compare the detection performance when the number of unknown attacks increases, and we explain which classifiers are more suited to detect unknown attacks. Our analysis reveals the following:

- Classifiers suffer the introduction of unknown attacks, either because unknowns are undetected (especially when using DEEP and SUP classifiers) or because many false alarms are raised (this mostly occurs with UNS).
- SUP, META-SUP, and, to a lesser extent, DEEP classifiers are effective in detecting known attacks, but their detection performance drops significantly when unknown attacks occur.
- Instead, UNS classifiers have better detection performance of unknown attacks but are clearly outperformed by DEEP, SUP, and META-SUP when dealing with known attacks.

Meta-learning enhances the classification performance of both SUP and UNS classifiers. Most noticeably and contrary to common knowledge, META-UNS classifiers based on bagging [5] and boosting [6] ensembles improve detection performance to a point at which they are slightly worse than SUP, META-SUP, and DEEP classifiers against known attacks but have superior ability to detect unknown attacks.

2. Experimental Plan

This section details the experimental setup to compare the detection performance of supervised and unsupervised classifiers, with and without meta-learning, addressing known and unknown attacks. We designed and performed a quantitative evaluation organized into steps M1 to M5.

M1. We collect 11 public datasets containing data about intrusion detection. These datasets contain features collected by monitoring real or simulated systems during their normal operation and when they are under attack.

M2. We preprocess each dataset to obtain tabular CSV files. Each row of the CSV file represents a data point, and each column represents a feature, except for the last column, which is the binary label (normal/attack) and describes whether a row corresponds to an attack or the normal operation.

M3. Preprocessed datasets are used to generate training variants. Very briefly, given a dataset split into a training set and a test set and one attack category, we remove all the attacks of such categories from the training set; this way, we obtain a *training variant*. We repeat this procedure for all 11 training sets and all attack categories contained in each dataset. In total, we obtain 58 training variants. Classifiers are trained on the 11 training sets and the 58 training variants. The 11 training sets include all the attacks that are in the test sets, meaning that all attacks are known by the classifier. The training variants miss one attack category each, which instead appears in the test sets; this is equivalent to having unknown attacks. An example is shown in Figure 1.

M4. Afterward, we select classifiers from the categories SUP, DEEP, UNS, META-SUP, and META-UNS (see Table I). In total, we select 6 SUP, 4 DEEP, 11 UNS, 4 META-UNS, and 22 META-UNS classifiers. We train each of the 47 classifiers on each of the 11 training sets and each of the 58 training variants.

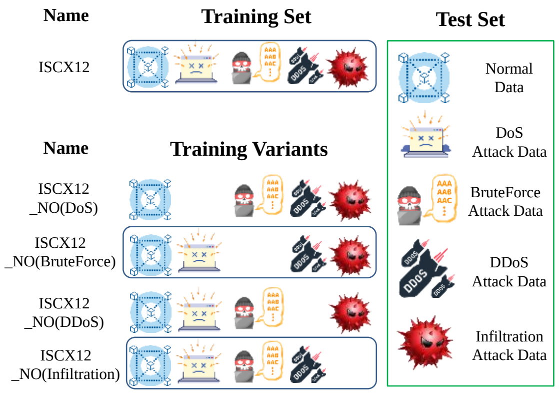


Figure 1. Creation of 4 training variants from the ISCX12 training set. The same approach is applied to the 11 datasets. The normal data (no attacks) in the training set and in the training variants are the same. Each training variant has one attack category less than the training set. The test set is the same for the training set and the training variants.

Table I: Classifiers used in this study.

	Uses Meta-Learning		
	No		*Yes*
Supervised	**DEEP**	**SUP**	**META-SUP**
	AutoGluon, FastAI, Py-Custom, TabNet	kNN, LDA, Naïve Bayes, Logistic Regression, SVM	*Bagging*: Random Forest *Boosting*: ADABoost, Gradient Boosting, XGBoost
Unsupervised	**UNS**		**META-UNS**
	COF, FastABOD, G-Means, K-Means, LOF, ODIN, One-Class SVM, HBOS, LDCOF, SDO, SOM, iForest		*Bagging*: ensembles of each UNS *Boosting*: ensembles of each UNS

M5. Finally, we collect the metric scores of all classifiers, and we create tables and plots to drive discussions and analyses.

Experiments are executed on a Dell Precision 5820 Tower with an Intel I9-9920X, GPU NVIDIA Quadro RTX6000 with 24 GB VRAM, 192 GB RAM, and Ubuntu 18.04, and they required approximately 6 weeks of 24 h execution. We provide GPU support to exercise DEEP classifiers.

The *Scikit-Learn* and *xgboost Python* packages contain all the code needed to exercise SUP and META-SUP classifiers, including mechanisms for grid searches. Instead, we exercised UNS and META-UNS classifiers through RELOAD [29], a Java open-source tool that includes many implementations of unsupervised classifiers and supports the creation of meta-learners. These frameworks allow the easy calculation of Accuracy (ACC), Matthews Correlation Coefficient (MCC), and Recall (REC) metric values. Classifiers were trained using the combination of parameters that resulted in the highest MCC after grid searches, embracing many hyper-parameter combinations. Overall, we trained each of the 47 classifiers on each of the 11 training sets and the 58 training variants. Models obtained at the end of this process are used to evaluate detection performance.

Notably, ACC, MCC, and REC metrics do not quantify detection capabilities with respect to unknown attacks. Therefore, we define two new quantities TU and FU similar to TP and FN but related to the occurrence of unknown attacks. Then, we combine TU and FU into the Recall-Unknown (Rec-Unk) metric as follows:

$$Rec - Unk = \frac{TU}{TU + FU}$$

Rec-Unk shows the fraction of unknown attacks detected by the classifier out of all the unknown attacks. The higher the Rec-Unk is, the better coverage a classifier has in detecting unknown attacks. Computing Rec-Unk required writing a simple Python function that filters out normal data and known attacks from the test set. This way, it becomes easy to compute TU and FU.

3. Discussion

We evaluate the ability to detect unknown attacks of DEEP, META-SUP, and META-UNS. We avoid considering SUP and UNS classifiers because their meta-learning counterparts, META-SUP and META-UNS, respectively, are definitely better. Figure 2 shows box plots for the META-SUP XGBoost, DEEP AutoGluon, and META-UNS FastABOD - Boosting classifiers with varying numbers of unknown attacks. These three classifiers have the best ACC and MCC scores in their groups. The blue box on the left of the figure draws MCC scores when no unknowns occur: XGBoost and AutoGluon scores are higher than those of FastABOD. However, increasing the percentage of unknowns in the test set makes the MCC of supervised and deep classifiers drop by a noticeable amount, whereas the MCC of FastABOD suffers only a minor degradation (see red, green, and purple boxes).

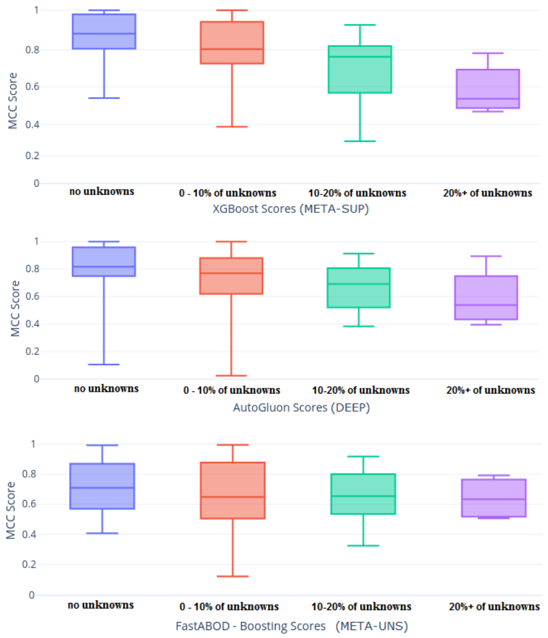


Figure 2: Box-plots showing the MCC scores of META-SUP XGBoost, DEEP AutoGluon and META-UNS FastABOD - Boosting classifiers when i) all attacks are known, ii) at most 10% are unknown attacks, iii) between 10 and 20% are unknown attacks, and iv) more than 20% are unknown attacks.

We elaborate on this with the aid of Figure 3, which we built according to the following procedure. We train all classifiers using a training set, evaluate them on the test set and select the supervised (SUP, META-Sup, or DEEP) and unsupervised (UNSUP or META-UNS) classifiers with the highest MCC. These two classifiers are considered the best supervised and unsupervised approaches for a given training set. We repeat this process by training all classifiers using training variants, evaluating them on the test set, and selecting the supervised and unsupervised classifiers that have the highest MCC. Last, we repeat the procedure but measure Rec-Unk instead of MCC. In total, this produces 69 points in the figure, obtained from the 11 training sets and the 58 training variants.

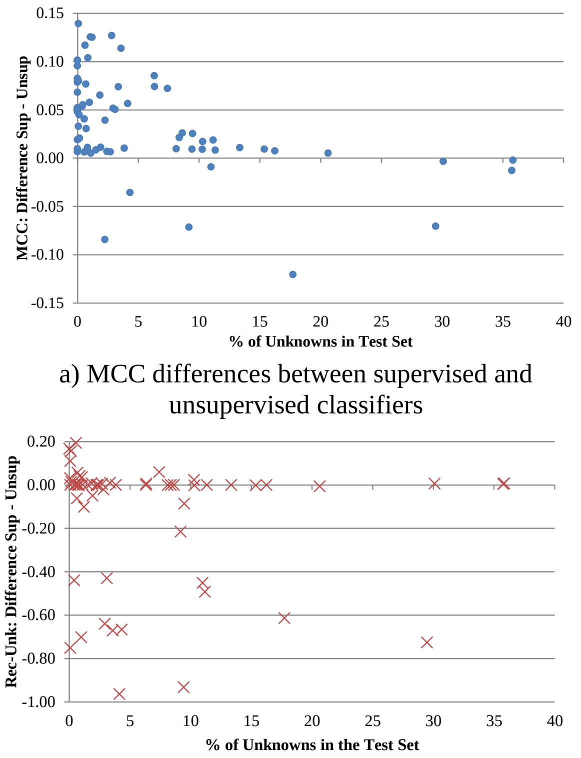
For each training set and training variant, we compute the difference in MCC (Figure 3a) and Rec-Unk (Figure 3b) between the two best classifiers previously selected. These differences are ultimately depicted in a scatterplot against the percentage of unknowns in the test set. As previously discussed, there are no unknowns when training on the training sets: the corresponding results are on the x=0 axes. In the other cases, the ratio of unknowns differs depending on the training variant, from 0.5% to almost 40%. Both scatterplots contain 69 items, i.e., one item for each training (on the 11 datasets and 58 variants). Items above the x-axis point to datasets or training variants where a supervised classifier is better than an unsupervised classifier.

Clearly, different classifiers may be selected when varying the training sets and training variants. Particularly, the META-SUP XGBoost is selected in 36 out of 69 cases, the META-SUP Random Forests in 18 out of 69, and the DEEP AutoGluon in 10 out of 69. For unsupervised classifiers, FastABOD outperforms others in 26 out of 69 cases, SDO in 12 out of 69, HBOS in 7 out of 69, and ODIN in 6 out of 69.

Figure 3a highlights that SUP, DEEP and META-SUP classifiers usually result in higher MCC scores, with fewer misclassifications – both FPs and FNs – than UNS and META-UNS classifiers. This trend becomes progressively less evident as the number of unknowns in the test set increases: on the right of the plot, the difference in MCC scores becomes almost negligible or even negative, meaning that there is a turning point at which unsupervised classifiers become better overall. Figure 3b shows the superior capabilities of UNS and META-UNS in detecting unknown attacks. The difference in Rec-Unk is almost always negative: unsupervised classifiers are better than supervised classifiers in identifying unknown attacks.

4. Conclusions

Supervised classifiers and, to a lesser extent, deep learners are usually accurate in detecting known attacks, but they cannot effectively detect unknown attacks (either brand new attacks, variants of existing exploits, or threats against which the intrusion detector is unprepared). Conversely, unsupervised classifiers are usually less accurate than supervised classifiers, but they do not suffer major degradation in case of unknown attacks. This paper conducted an experimental analysis to compare supervised and unsupervised classifiers, with and without the adoption of meta-learning, to quantitatively analyze their ability in detecting known and unknown attacks. Results showed that unsupervised meta-learners are the best solution to detect unknown attacks, and can detect known attacks similarly to several supervised classifiers. Summarizing, unsupervised meta-learning is a promising approach to implement IDSs: it offers satisfactory detection accuracy in case of both known and unknown attacks.



a) MCC differences between supervised and unsupervised classifiers

b) RecUnk differences between supervised and unsupervised classifiers

Figure 3: Differences in MCC (Figure 3a) and Rec-Unk (Figure 3b) of the best supervised classifier versus the best unsupervised classifier when trained on the 11 training sets and the 58 training variants. Differences are plotted against the % of unknowns in the test set.

5. Acknowledgements

This work presents a summary of the open access paper Zoppi, T., Ceccarelli, A., Puccetti, T., & Bondavalli, A. (2023). Which Algorithm can Detect Unknown Attacks? Comparison of Supervised, Unsupervised and Meta-Learning Algorithms for Intrusion Detection. Computers & Security, 103107 and it has been supported by the Regione Toscana POR FESR 2014-2020 SPaCe and by the NextGenerationEU program, Italian DM737 – CUP B15F21005410003.

6. References

[1] Akyildiz, I. F., & Kak, A. (2019). The Internet of Space Things/CubeSats: A ubiquitous cyber-physical system for the connected world. Computer Networks, 150, 134-149.

[2] Dey, N., Ashour, A. S., Shi, F., Fong, S. J., & Tavares, J. M. R. (2018). Medical cyber-physical systems: A survey. Journal of medical systems, 42(4), 1-13.

[3] A zero-day guide for 2020: Recent attacks and advanced preventive techniques (online), https://blog.malwarebytes.com/exploits-and-vulnerabilities/2020/06/a-zero-day-guide-for-2020/

[4] Goldstein, M., & Uchida, S. (2016). A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. PloS one, 11(4), e0152173.

[5] Breiman, Leo. "Random forests." Machine learning 45.1 (2001): 5-32.

[6] Rätsch, Gunnar, Takashi Onoda, and K-R. Müller. "Soft margins for AdaBoost." Machine learning 42.3 (2001): 287-320.

[7] Zhang, C., Jia, D., Wang, L., Wang, W., Liu, F., & Yang, A. (2022). Comparative Research on Network Intrusion Detection Methods Based on Machine Learning. Computers & Security, 102861.

[8] Shwartz-Ziv, R., & Armon, A. (2022). Tabular data: Deep learning is not all you need. Information Fusion, 81, 84-90.

[9] Erickson, N., Mueller, J., Shirkov, A., Zhang, H., Larroy, P., Li, M., & Smola, A. (2020). Autogluon-tabular: Robust and accurate automl for structured data. arXiv preprint arXiv:2003.06505.

[10] Ardagna, C., Corbiaux, S., Sfakianakis, A., Douliger, C., ENISA Threat Landscape 2021 (online), https://www.enisa.europa.eu/topics/threat-risk-management/threats-and-trends (last accessed: 4th August, 2022)

[11] Connell, B., “2022 SonicWall Threat Report” (online), https://www.sonicwall.com/2022-cyber-threat-report/ (last accessed: 4th August, 2022)

[12] Domenico Cotroneo, Roberto Natella, and Stefano Rosiello. 2017. A fault correlation approach to detect performance anomalies in Virtual Network Function chains. In Software Reliability Engineering (ISSRE), 2017 IEEE 28th Int. Symposium on. IEEE, 90–100.

[13] Cruz, T., Barrigas, J., Proença, J., Graziano, A., Panzieri, S., Lev, L., & Simões, P. (2015, May). Improving network security monitoring for industrial control systems. In 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM) (pp. 878-881) IEEE.

[14] Zoppi, T., Ceccarelli, A., & Bondavalli, A. (2019). MADneSs: A multi-layer anomaly detection framework for complex dynamic systems. IEEE Transactions on Dependable and Secure computing, 18(2), 796-809.

[15] Brazdil P, Giraud-Carrier C, Soares C, Vilalta R (2009) Metalearning: applications to data mining. Springer, Berlin.

[16] He, P., Zhu, J., He, S., Li, J., & Lyu, M. R. (2017). Towards automated log parsing for large-scale log data analysis. IEEE Transactions on Dependable and Secure Computing, 15(6), 931-944.

[17] Chou, D., & Jiang, M. (2021). A survey on data-driven network intrusion detection. ACM Computing Surveys (CSUR), 54(9), 1-36.

[18] Buczak, Anna L., and Erhan Guven. "A survey of data mining and machine learning methods for cyber security intrusion detection." IEEE Communications surveys & tutorials 18.2 (2015): 1153-1176.

[19] Al, S., & Dener, M. (2021). STL-HDL: A new hybrid network intrusion detection system for imbalanced dataset on big data environment. Computers & Security, 110, 102435.

[20] XGboost package (online) https://xgboost.readthedocs.io/en/stable/python/python_intro.html (last accessed: 4th August, 2022)

[21] Additional files for Submission (online ZIP file) https://github.com/tommyippoz/Miscellaneous-Files/blob/master/COSE22_Zoppi_SupportingMaterial.zip (last accessed: 4th August, 2022)

[22] Gorishniy, Y., Rubachev, I., Khrulkov, V., & Babenko, A. (2021). Revisiting deep learning models for tabular data. Advances in Neural Information Processing Systems, 34.

[23] Casas, Pedro, Johan Mazel, and Philippe Owezarski. "Unsupervised network intrusion detection systems: Detecting the unknown without knowledge." Computer Communications 35.7 (2012): 772-783

[24] Catillo, M., Pecchia, A., Rak, M., & Villano, U. (2021). Demystifying the role of public intrusion datasets: a replication study of DoS network traffic data. Computers & Security, 108, 102341.

[25] Avizienis, A., Laprie, J. C., Randell, B., & Landwehr, C. (2004). Basic concepts and taxonomy of dependable and secure computing. IEEE transactions on dependable and secure computing, 1(1), 11-33.

[26] Zhang, Z., Liu, Q., Qiu, S., Zhou, S., & Zhang, C. (2020). Unknown attack detection based on zero-shot learning. IEEE Access, 8, 193981-193991.

[27] Moller, F., Botache, D., Huseljic, D., Heidecker, F., Bieshaar, M., & Sick, B. (2021). Out-of-distribution detection and generation using soft brownian offset sampling and autoencoders. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 46-55).

[28] Ashrapov, I. (2020). Tabular GANs for uneven distribution. arXiv preprint arXiv:2010.00638.

[29] Zoppi, T., Ceccarelli, A., & Bondavalli, A. (2019, October). “Evaluation of Anomaly Detection algorithms made easy with RELOAD” In Proceedings of the 30th Int. Symposium on Software Reliability Engineering (ISSRE), pp 446-455, IEEE.