Computing and Informatics, Vol. 41, 2022, 12–33, doi: 10.31577/cai_2022_1_12

USING MACHINE LEARNING FOR INTRUSION DETECTION SYSTEMS

Quang-Vinh Dang

Industrial University of Ho Chi Minh City
Vietnam
e-mail: dangquangvinh@iuh.edu.vn

Abstract. Given the importance of the computer systems in our daily life today, it is decisive to be able to protect the computer systems against attacks. Intrusion Detection Systems (IDSs) are the crucial component of modern cybersecurity systems. IDSs are built-in in the devices of the major providers such as Cisco and Juniper. Since the early days of the Internet up to now, the IDSs rely heavily on signature-based detection methods. However, in recent years, researchers utilize the power of machine learning techniques and achieve very good performance in classifying network attacks. In this paper, we analyze the machine learning techniques that have been proposed in recent years. We propose some new techniques to improve the performance of the existing methods. The experimental results using real-world datasets show that our suggestions can boost the predictive accuracy of the models.

Keywords: Intrusion detection system, machine learning, computer security, cyber security

1 INTRODUCTION

It is hard to deny the importance of the computer systems in our modern life. As the computers play more and more a crucial role for human being, the attackers discovered more effective attacking methods to the systems. Furthermore, when the Internet of Things (IoT) becomes a reality, it is believed that every device can be attacked [1].

One of the most famous cyber-attacks is probably the Distributed Denial-of-Service (DDoS) [2]. The first DDoS attack we know today occurred in 1996 to Panix [3], one of the eldest ISPs in the world, using the SYN Flood attack [4].

Overtime, the number of DDoS attacks has increased dramatically [5]. In October 2016, a major DDoS attack has been launched to Domain Name System (DNS) that leads to a consequence that many websites such as Twitter, Netflix and Spotify have been shut down [6].

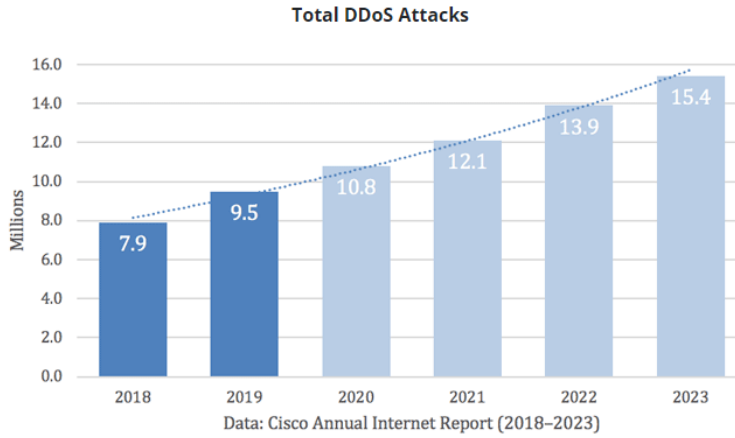

Figure 1. Number of DDoS attacks all over the world

Cybersecurity is defined as "prevention of damage to, protection of, and restoration of computers, electronic communications systems, electronic communications services, wire communication, and electronic communication, including information contained therein, to ensure its availability, integrity, authentication, confidentiality, and nonrepudiation." [7].

In order to deal with the cyber-attacks, the intrusion detection systems (IDSs) [8] are one of the most important components of the defense systems. We rely on the definition of intrusion and IDS from the work of [9]. We recall the computer system security policies that are Confidentiality, Integrity and Availability (CIA). An intrusion is an attack that tries to violate the CIA principles. An IDS is the system that is to detect any intrusion from outside of the system.

In years, the IDS are mostly based on signature matching techniques [9, 10, 11]. The IDSs based on signature-detection will look for predefined signature (or patterns) of the incoming network flow to stop the suspected ones. In reality, the techniques have achieved good performance in detecting known threats [12]. However, the systems that rely on signature matching cannot deal with novel attacks or zero-day attacks [13].

In recent years, many research studies utilized the rapid development of machine learning techniques to enhance the quality of IDSs. The core idea is to build a machine learning model that automatically learns the patterns rather than defining them manually, then let the model classify the incoming flows. The experimen-

tal results in several real-world datasets claimed the advantage of machine learning based methods.

The rest of the paper is organized as follows. In Section 2 we review the up to date methods, particularly machine learning techniques, that have been used as the core of an IDS. We review the popular IDS dataset to train and evaluate models in Section 3. We discuss our method to improve in Section 4. We conclude our paper and draw some potential future research directions in Section 5.

2 RELATED WORKS

Since the early days of the computer systems up to date, many IDSs rely on signature matching to function. The method is known as knowledge based detection or misuse detection [9]. A signature is defined as a pattern or string or any other specific characters that is known to be as an attack or a threat [14]. An IDS based on the signature matching will try to match a network flow to a known attack to detect them. As of this writing, two major network solution providers Cisco and Juniper both implemented IDSs on their devices and they all rely on signature matching, hence the signature database must be updated periodically. The most important advantage of the approach is its speed that allows the devices to perform as in normal conditions without affecting the entire network. On the other hand, in order to use the approach the network devices cannot work autonomously, and it cannot deal with novel and zero-attack [15, 16].

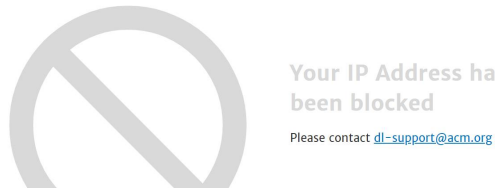

Figure 2. A signature-based prevention method. In this case, the rule is: if the source IP is in a black list, block it.

In the last decade, many research studies focus on using machine learning to replace the signature matching [17, 18]. In short, the aim of machine learning is to equip a computer the ability to learn [19]. Hence, the cybersecurity researchers do not need to define explicitly the patterns anymore, and the signature database can be updated automatically.

In this study, we classify the machine learning techniques using in IDSs in four categories: supervised learning, unsupervised learning, deep learning and reinforcement learning. There is no clear distinction between them as a method might use a combination of different methods, or an algorithm might belong to more than one category.

The first category we consider is the supervised learning approach. In general, the researchers have a set of network traffic that are labelled already, i.e. the researchers know what flow is *benign* and what flow is *malicious*. The researchers will generate the features to describe these flows, either manually or automatically [18]. Then the researchers build a machine learning model which is usually a classifier to learn the characteristics of different flow types, then use the model to classify the future flows. As of this writing, supervised learning is probably the most popular approach in literature.

As discussed above, a model that is employed inside an IDS must satisfy both requirements: high accuracy in classification and little running time. Due to these requirements and the limitation of the computational power of the network devices where IDSs are installed, one of the most popular algorithms is the decision tree algorithm [20, 10, 21, 22], random forest [23] and SVM [24, 25].

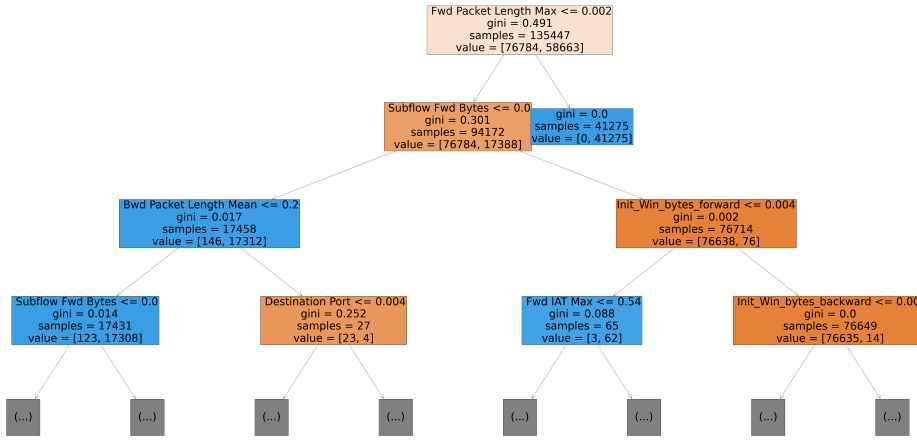

Figure 3. A decision tree to classify the DDoS attack trained on the CICIDS2017 dataset. Source: [15]

In Figure 3, we visualize a single decision tree build on top of the CICIDS 2017 dataset [26]. Even the decision tree is simple and usually considered as a weak learner [27], in many of cases its performance is good enough. More importantly, it is easy to explain the prediction of a decision tree [28]. The requirement of the explainability is addressed at least since the year of 2000 in a technical report by [29] but the requirement has been ignored for a long time, mostly due to the fact that the popular algorithms during this period of time are self-explainable. However, in recent years there is more and more demand to request the machine learning models be able to explain their outcome [15]. In fact, the decision tree is considered as one of the most easy to understand algorithms for human. Several researchers [30] are working on converting any learning algorithms to the decision tree to explain.

There are several efforts to perform reverse engineering in a particular machine learning model to convert the model into a decision tree [30]. However, the predictive performance of the decision tree algorithm is usually not comparable to other algorithms. The decision tree is often considered as a *weak learner*. Two main approaches to improve the predictive performance of a decision tree is *boosting* and *bagging*, as visualized in Figure 4.

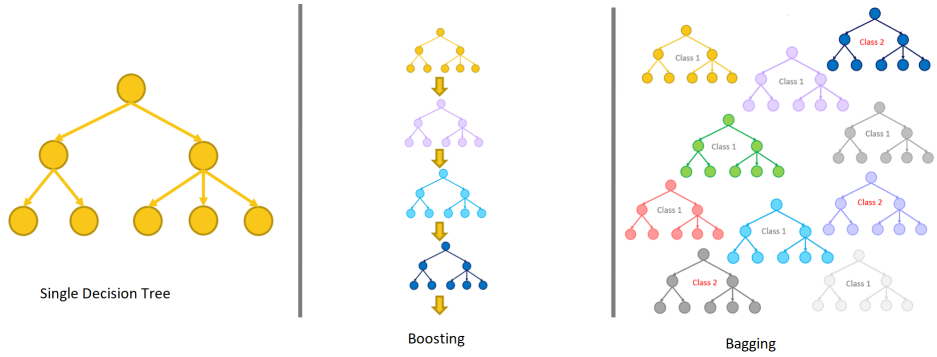

Figure 4. Bagging and boosting mechanisms to improve the predictive performance of decision tree [15]

The idea of the bagging approach is to build multiple trees independently then summarize their outcome. On the other hand, the core idea of the boosting algorithm is let the following tree to fix the error the previous trees made. The most famous instance of the bagging mechanism is the random forest algorithm [31]. There are several implementations of the boosting mechanisms such as xgboost [32], LightGBM [33] and CatBoost [34].

CatBoost has been evaluated in [35] to detect intrusion in the KDD99 dataset, and xgboost is studied comprehensively in [17]. We might conclude that their predictive performances are similar. However, the recent research work [36] showed that CatBoost might not be as good as xgboost in term of the predictive accuracy, however its superiority lies on the computing speed which is much faster than xgboost.

When the computational power of the network devices is improved, more complicated algorithms are considered. For instance, in [17], the *xgboost* algorithm [32] is used. The authors achieved the near-perfect predictive results with the AUC score is almost 1.0. However, the authors claimed that we might not need a heavy model like *xgboost* to do the task: a careful feature engineering process can boost the performance of other weak learning algorithms such as the Naive Bayes classification.

As the predictive performance is reaching perfect in some scenarios, many research studies are now focusing on reduction of the required computational powers for IDSs.

One approach is leveraging the active learning [37]. In the active learning setting, a learning model starts with a small subset of the training dataset then actively selects the new training instance from the pool based on some metrics to add to the training dataset. By doing so, the learning model only need to deal with a small number of training samples that have the highest impact to the performance while ignoring the ineffective samples. Research studies [38, 39, 40, 41] showed empirically that the strategy of using active learning can maintain the performance of the model while reducing the training data size.

Another approach to scale down the power consumption of an IDS is to limit the number of dimension of the dataset using the dimensional reduction techniques such as Principal Component Analysis (PCA) [42]. The core idea of PCA is to project the existing data into a new space with fewer dimensions but it still can explain as much as the variance of the original data. The researchers [43, 44] applied PCA before feeding the new data into a classifier, here it is SVM. However, these approaches have some limitation, as addressed in [15]:

- Training PCA itself takes a lot of time.
- PCA requests the null-handling method is used beforehand.
- New data will need to be fed through the PCA model before the learning model, thus increasing the entire processing time rather than reducing it.

More recently, the authors of [45] proposed to use the Deep Belief Networks [46] for the automatic feature learning, integrating with Particle Swarm Optimization (PSO). The model of the work is displayed in Figure 6.

The second category we study in this paper is the unsupervised learning approach. The unsupervised learning approach is used when the attacks are not known or difficult to gather and define. The most popular unsupervised learning techniques using in the literature is the family of anomaly detection techniques [47]. Several traditional anomaly detection techniques such as LOF, k-nearest neighbors, Mahalanobis distance and unsupervised SVM are evaluated in [48].

The authors of [49] rely on mixture models and probability modelling to detect the anomaly. The work is extended to the Bayesian setting in [50]. In [51], the authors included recent techniques such as Isolation Forest [52] into the comparison. The idea of the Isolation Forest is to classify a single instance in the dataset. More difficult it is to classify a particular instance, more outlier the instance is. The authors of [41] integrated the Isolation Forest into the active learning scheme to select the next training instance.

In [53], the authors review comprehensively different anomaly detection algorithms, including clustering algorithms like K-means, statistic-based methods like Histogram-Based Outlier Score (HBOS) [54], classification methods like One-class SVM or Isolation Forest, Neural Networks, neighbour-based methods like kNN, angle-based methods [55], density-based methods [56] and mixed methods on different datasets, range from NSL-KDD (1999) to CICIDS 2018. The experimental results show that the classification methods (One-class SVM, Isolation Forest)

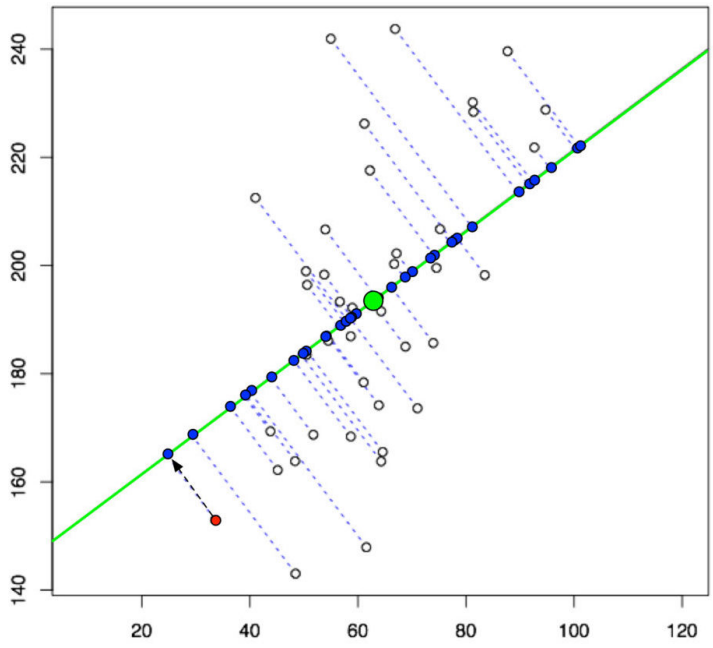

Figure 5. Principal Component Analysis. If we project the original data to the green line, we can reduce the number of dimension to 1 while keeping most of the variance of the data.

achieve the highest predictive score. However, the authors pointed out a very important point: a high variation of performance of all algorithms between datasets. It means we did not yet find a global anomaly algorithm that at least can reliably perform in multiple data.

We refer to two reviews on the anomaly detection techniques in network intrusion detection [57, 58] for a more detailed review of anomaly detection in the network.

We note that a different approach in clustering for anomaly detection which is called fuzzy clustering existed [59]. However, the method has not been fully studied in literature.

The third category we take into consideration is to use deep learning techniques to power the IDSs. For instance, in [60], the authors use the multi-layer feed-forward neural networks as the core model of the classification task. The authors of [61] leveraged natural language processing techniques to analyze the system logs.

The deep learning techniques have been studied comprehensively in the IoT settings [62]. The difference is that the problem can be formulated as the multi-agent setting [34] in the IoT scenario. In [63], the researchers design a simulated test-bed and a deep learning model follows a feature selection using random decision trees and Pearson correlation. Deep learning has been also utilized in other IoT envi-

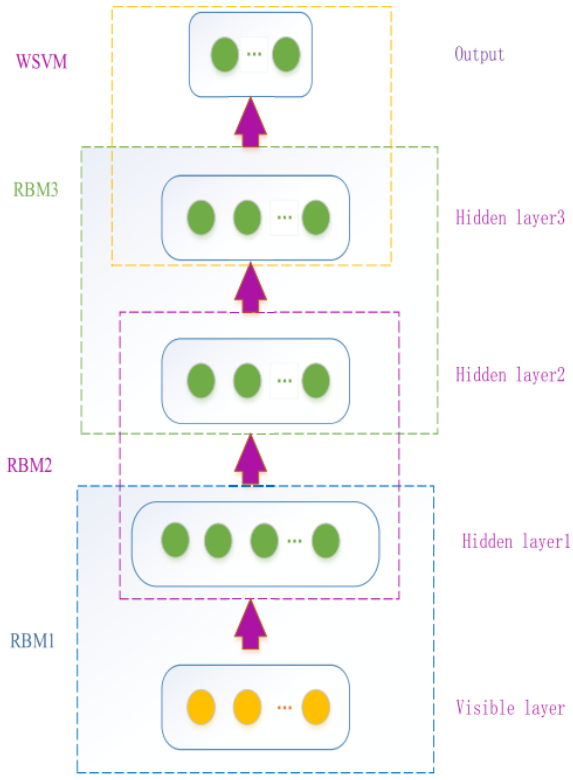

Figure 6. Deep belief networks with weighted SVM [45]

ronments, such as in-vehicle IoT [64] or IoT at home [65]. Most recently, several research studies utilize the graph neural networks to detect the anomaly [66, 67, 62]. For instance, [66] suggested to use an attention-based temporal graph convolutional neural network [68] to detect the anomalous edges. The most prominent usage of deep learning is probably to be used as a feature extractor [18]. One instance is [45] where the researchers used the Deep Belief Networks [46] for the automatic feature learning – followed by the Particle Swarm Optimization (PSO). The authors of [69] used LSTM-Auto Encoder to extract the features automatically.

The fourth category we study is the reinforcement learning for intrusion detection systems. One of the earliest works is [70]. The idea of the reinforcement learning is that the model does not know fully the nature (benign/malicious) of the network flows like in the supervised learning but something about the impact (the reward) that the allowed flows might cause to the system. By interacting with the

system (allow/deny a particular flow), the model can gain enough information to make a better decision [71, 72, 73].

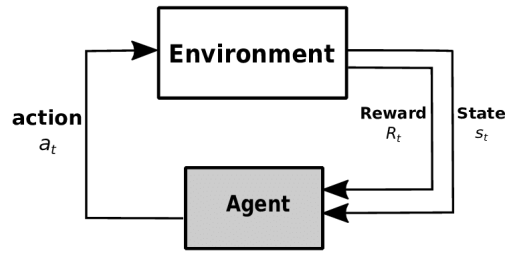

Figure 7. Reinforcement Learning algorithms learns by interacting with the environment, here the external networks. The agent performs something and observes the response (reward), then updates the policy.

The main disadvantage of the reinforcement learning scheme is that the model requires to interact with the system and be allowed to fail enough times. This requirement might not be available in practice. The authors of [74] suggested a novel method to replace the environment by a sampling strategy from a labelled dataset, but it limited the application of using the reinforcement learning. There is still a lot of room for improvement in using reinforcement learning for intrusion detection.

Lately, the research works on offline reinforcement learning [75] shed a new light into the problem of using reinforcement learning for the intrusion detection problem. In offline reinforcement learning, the agent (here the IDS) does not interact with the environment, but sampling the data from an offline dataset. The setting makes the reinforcement learning more practical in real-world settings.

3 DATASETS AND EVALUATION METRICS

3.1 Datasets

In this section we review some most popular datasets to train and evaluate the IDSs. We refer the audience to recent surveys [76] for further details. Most of the widely used IDSs datasets in the literature belong to the NetFlow family [77].

The very first dataset for IDSs is probably the dataset DARPA98 created at MIT Lincoln Lab. The enhanced version of the dataset, known as KDD-99 [78], is one of the most popular dataset used in the literature [58], together with another enhanced version which is NSL-KDD [79]. These datasets play an important role in the development history of intrusion detection research, but even since the beginning the quality of the datasets have been questioned [17]. A common critical point is that the dataset does not reflect the true distribution of attacks. As of this writing,

these datasets are somehow outdated and do not represent an efficient tool for the current attack methods [76].

The DEFCON-8 and DEFCON-10 datasets are releases of 2000 and 2002, respectively [80]. These datasets were created for a competition and not supposed to be realistic. However, the datasets have been used extensively for evaluation [26].

In the year of 2005, the Lawrence Berkeley National Laboratory released the LBNL dataset [81]. The CDX dataset [82] is released to be a replacement of KDD datasets. However, both datasets are considered as not realistic enough to be used in practice [83].

The Canadian Institute for Cybersecurity (CIC) has spent a lot of time and effort in recent years to create a realistic dataset for training and evaluating IDSs. The first outcome was introduced in 2012 [84] as the result of seven days of data collection, but the dataset lacks of HTTPS traffic [76]. The enhanced version of this dataset is CICIDS 2017 which includes a lot of modern protocols [26, 83]. One year later, the CIC released an improved version of the CICIDS 2017 which is known as CSE-CIC-IDS2018 [85]. The CIC also released the dataset DoHBrw-2020 [86] for a specific type of attacks on Domain Name System.

As the CICIDS 2018 is considered as a modern intrusion dataset that is built upon a realistic context, it still has a problem of extremely imbalanced dataset [15]. It means that a number of attack types have very few instances, so we cannot draw any statistically significant conclusions.

Dataset	Testbed Configuration Realistic	Traffic Realistic	Labeled	IoT	Attacks Diverse	Full Packet	New Generated Features
Bot-IoT	T	T	T	T	T	T	T
CAIDA	T	T	F	F	F	F	F
DARPA 98	T	F	T	F	T	T	F
DEFCON 8	F	F	F	F	T	T	F
KDD 99	T	F	T	F	T	T	T
CICIDS 2017	T	T	T	F	T	T	T
CICIDS 2018	T	T	T	F	T	T	T
DoHBrW 2020	T	T	T	F	F	T	T

Table 1. Summary of the intrusion datasets

One major drawback of the above datasets is that they do not explicitly state the *testset*, hence each researchers might (and actually will) use a different configuration of train/test split. Usually the split is random and not reproducible, so it is impossible to accurately compare the performance of different algorithms.

Another drawback is that the datasets are released with predefined features but not the raw data. It limits the potential of researchers to create features different from the existing ones. Furthermore, it is difficult to join the different datasets because they have different feature sets.

3.2 Evaluation Metrics

As the problem of intrusion detection is formalized as a classification problem, standard evaluation metrics are often being used [87]. These metrics include accuracy, true positive rate, false positive rate, F1-score and MCC [88]. These metrics all require the confusion matrix to calculate, meaning that they require an instance in the testset needed to be labelled as benign/malicious directly by the IDS. We recall the formulas of the metrics as follows.

$$accuracy = \frac{\#of\_correct\_prediction}{\#of\_prediction}, \tag{1}$$

$$F1 = 2 * \frac{precision * recall}{precision + recall}, \tag{2}$$

$$MCC = \frac{TP * TN - FP * FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}. \tag{3}$$

On the other hand, the AUC score does not require the confusion matrix but probability of being malicious assigned to each instance by the IDS. The AUC gives more power to the practitioners because they can modify the threshold to satisfy the business requirements. Furthermore, let us consider a case when there are two instances with the probability of being malicious determined by an IDS as 0.499999 and 0.5000001 – in fact the two instances are mostly the same from the point of view of the IDS, but if we use the threshold of 0.5, one will be blocked and the other one can go through to enter the computer system. Unfortunately the AUC is available only for binary classification, even though there are some efforts to extend the metric to the multi-class classification case [89].

4 EXPERIMENTAL RESULTS

In this section we analyze our experiments. We performed multiple classification with comparison, and applying different techniques to enhance the predictive performance of the classification, including data augmentation, regularization, feature selection and active learning [41]. The feature selection is done based on the feature importance assessment of models [16], hence we keep removing the features until the predictive performance of the model drops. We try some data augmentation techniques [90].

We use the CICIDS 2018 dataset for the evaluation. We divided the train – evaluation – test set by the ratio of 60:20:20.

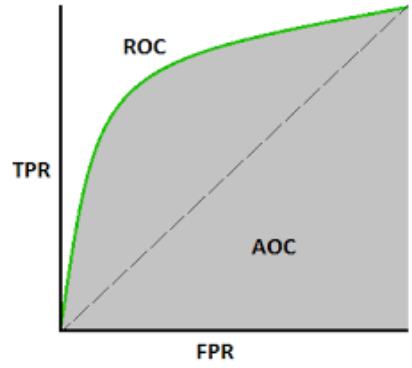

Figure 8. Area Under the Curve. We measure the performance of a classifier by the area under the curve formed by the True Positive Rate (TPR) and False Positive Rate (FPR). Higher value is better.

We present our experimental results in Table 2 and Figure 9. We can see that xgboost and catboost achieved the near-perfect predictive results. Furthermore, the training and inference time of algorithms are presented in Table 3. We notice that, due to the very good performance already of tree ensemble methods, feature engineering does not significantly improve the performance anymore.

We can confidently claim that there is no much room for improvement in traditional benign/malicious classification problem as the classifier is almost perfect. The future research studies should focus on different research problems. We discuss these problems in the next section.

Algorithm	AUC
Naive Bayes	0.5
Logistic Regression	0.55
SVM (linear kernel)	0.62
OCSVM (RBF kernel)	0.57
Random Forest	0.92
xgboost	0.9992
xgboost with Active Learning [41]	0.95
xgboost with Feature Engineering [15]	0.999995
catboost	0.9992

Table 2. AUC of different classifier in binary setting [17]

5 CONCLUSIONS

In this study, we review and present the usage of machine learning models, including supervised learning, unsupervised learning, deep learning and reinforcement

Algorithm	Training Time (seconds)	Predicting Time (seconds)
Naive Bayes	2	0.2
Logistic Regression	8 000+	20
SVM (linear kernel)	10 000+	25
OCSVM (RBF kernel)	4 000+	24
Random Forest	300	2.3
xgboost	3 600+	10
xgboost with Active Learning [41]	3 600+	10
xgboost with Feature Engineering [15]	3 600+	10
catboost	700+	12

Table 3. Training and inference time of different classifier in binary setting [17]

learning for the problem of intrusion detection. The core idea is to detect an intrusion before letting it enter the protected computer system. The IDSs do that by classification and allow only benign network flows go through. We review several popular datasets, started from some classical ones like DARPA 98 up to the recent released datasets. We claim that, by using the state-of-the-art machine learning algorithms running on powerful machines, the classification problems are mostly solved.

We believe that the future research works should address the following problems:

- How to optimize the running cost of the IDSs?
  - It is not an accident that big companies like Cisco still use signature-based methods. The IDSs are usually employed in network devices with limited computational power but real-time processing requirements. A comprehensive algorithm might be good in research but will not be practical in real life.
- How to learn with limited number of training data points?
  - In fact, many research studies have to ignore some kinds of attacks, such as the Heartbleed attack in the CICIDS 2018 dataset [16] because there is not enough instances of these attacks for both training and evaluation. However, letting only one instance of attacks to the internal system might be more than enough to destroy the entire system. We need to find a method to cope with serious attack like this.
- How to let the IDSs to work completely autonomously, including self-evolving without a human intervention?
  - An IDS shall have the ability to know when the database is outdated, or there is some error/noise in the training dataset and retrain itself.
- Distributed IDSs and sharing data to deal with novel and rare attacks.

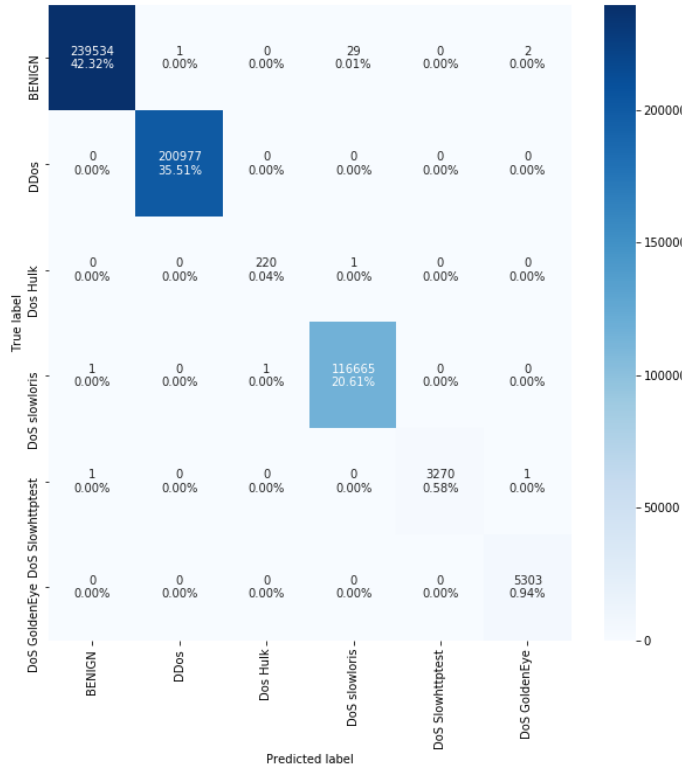

Figure 9. Confusion matrix of the intrusion classifier on all types of attacks [15]

  - When a novel attack occurs in one node, the entire network should be noticed. Furthermore, the training process might be distributed to speed up and utilize the idle nodes.

- Off-policy reinforcement learning for IDSs.

  - A standard reinforcement learning is not applicable in the context of IDSs as we cannot let the attacks happen to learn from the feedback of the environment.

- Robust anomaly detection for IDSs.

  - Current anomaly detectors tend to vary in term of predictive performance in different datasets, hence they cannot deal with distribution shift.

It is no doubt that the problems like IDSs will never be completed, as new attack methods will be introduced over time and new problems will be raised in the future.

REFERENCES

[1] Diogenes, Y.—Ozkaya, E.: Cybersecurity – Attack and Defense Strategies. 2nd Edition. Packt Publishing Ltd., 2020.

[2] Douligeris, C.—Mitrokotsa, A.: DDoS Attacks and Defense Mechanisms: Classification and State-of-the-Art. Computer Networks, Vol. 44, 2004, No. 5, pp. 643–666, doi: 10.1016/j.comnet.2003.10.003.

[3] Panix. 2020, https://www.panix.com/ [accessed 06-Dec-2020].

[4] Bogdanoski, M.—Shuminoski, T.—Risteski, A.: Analysis of the SYN Flood DoS Attack. International Journal of Computer Network and Information Security (IJCNIS), Vol. 5, 2013, No. 8, pp. 1–11, doi: 10.5815/ijcnis.2013.08.01.

[5] Cisco: Cisco Annual Internet Report (2018–2023) White Paper. 2020.

[6] McAfee: Major Websites Twitter, Spotify, Netflix Shut Down by DDoS Attack. 2016, https://www.mcafee.com/blogs/consumer/consumer-threat-notices/dyn-ddos-attack/ [accessed 6-Dec-2020].

[7] Committee on National Security Systems: CNSSI No. 4009: Committee on National Security Systems (CNSS) Glossary. 2015.

[8] Ozkaya, E.: Cybersecurity: The Beginner's Guide. Packt, 2019.

[9] Liao, H. J.—Lin, C. H. R.—Lin, Y. C.—Tung, K. Y.: Intrusion Detection System: A Comprehensive Review. Journal of Network and Computer Applications, Vol. 36, 2013, No. 1, pp. 16–24, doi: 10.1016/j.jnca.2012.09.004.

[10] Kruegel, C.—Toth, T.: Using Decision Trees to Improve Signature-Based Intrusion Detection. In: Vigna, G., Kruegel, C., Jonsson, E. (Eds.): Recent Advances in Intrusion Detection (RAID 2003). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 2820, 2003, pp. 173–191, doi: 10.1007/978-3-540-45248-5_10.

[11] Verwoerd, T.—Hunt, R.: Intrusion Detection Techniques and Approaches. Computer Communications, Vol. 25, 2002, No. 15, pp. 1356–1365, doi: 10.1016/S0140-3664(02)00037-3.

[12] Depren, O.—Topallar, M.—Anarim, E.—Ciliz, M. K.: An Intelligent Intrusion Detection System (IDS) for Anomaly and Misuse Detection in Computer Networks. Expert Systems with Applications, Vol. 29, 2005, No. 4, pp. 713–722, doi: 10.1016/j.eswa.2005.05.002.

[13] Bilge, L.—Dumitraş, T.: Before We Knew It: An Empirical Study of Zero-Day Attacks in the Real World. Proceedings of the 2012 ACM Conference on Computer and Communications Security (CCS '12), 2012, pp. 833–844, doi: 10.1145/2382196.2382284.

[14] Stavroulakis, P.—Stamp, M.: Handbook of Information and Communication Security. Springer Science and Business Media, 2010, doi: 10.1007/978-3-642-04117-4.

[15] Dang, Q. V.: Understanding the Decision of Machine Learning Based Intrusion Detection Systems. In: Dang, T. K., Küng, J., Takizawa, M., Chung, T. M. (Eds.): Future Data and Security Engineering (FDSE 2020). Springer, Cham, Lecture Notes in Computer Science, Vol. 12466, 2020, pp. 379–396, doi: 10.1007/978-3-030-63924-2_22.

[16] Dang, Q. V.: Improving the Performance of the Intrusion Detection Systems by the Machine Learning Explainability. International Journal of Web Information Systems, Vol. 17, 2021, No. 5, pp. 537–555, doi: 10.1108/IJWIS-03-2021-0022.

[17] Dang, Q. V.: Studying Machine Learning Techniques for Intrusion Detection Systems. In: Dang, T., Küng, J., Takizawa, M., Bui, S. (Eds.): Future Data and Security Engineering (FDSE 2019). Springer, Cham, Lecture Notes in Computer Science, Vol. 11814, 2019, pp. 411–426, doi: 10.1007/978-3-030-35653-8_28.

[18] Kumar, P.—Kumar, A. A.—Sahayakingsly, C.—Udayakumar, A.: Analysis of Intrusion Detection in Cyber Attacks Using DEEP Learning Neural Networks. Peer-to-Peer Networking and Applications, Vol. 14, 2021, No. 4, pp. 2565–2584, doi: 10.1007/s12083-020-00999-y.

[19] Halder, S.—Ozdemir, S.: Hands-On Machine Learning for Cybersecurity. Packt, 2018.

[20] Li, X.—Ye, N.: Decision Tree Classifiers for Computer Intrusion Detection. Journal of Parallel and Distributed Computing Practices, Vol. 4, 2001, No. 2, pp. 179–190.

[21] Amor, N. B.—Benferhat, S.—Elouedi, Z.: Naive Bayes vs Decision Trees in Intrusion Detection Systems. Proceedings of the 2004 ACM Symposium on Applied Computing (SAC '04), 2004, pp. 420–424, doi: 10.1145/967900.967989.

[22] Stein, G.—Chen, B.—Wu, A. S.—Hua, K. A.: Decision Tree Classifier for Network Intrusion Detection with GA-Based Feature Selection. Proceedings of the 43rd Annual Southeast Regional Conference – Volume 2 (ACM-SE 43), ACM, 2005, pp. 136–141, doi: 10.1145/1167253.1167288.

[23] Resende, P. A. A.—Drummond, A. C.: A Survey of Random Forest Based Methods for Intrusion Detection Systems. ACM Computing Surveys, Vol. 51, 2019, No. 3, Art. No. 48, doi: 10.1145/3178582.

[24] Reddy, R. R.—Ramadevi, Y.—Sunitha, K. V. N.: Effective Discriminant Function for Intrusion Detection Using SVM. 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI), IEEE, 2016, pp. 1148–1153, doi: 10.1109/ICACCI.2016.7732199.

[25] Bhamare, D.—Salman, T.—Samaka, M.—Erbad, A.—Jain, R.: Feasibility of Supervised Machine Learning for Cloud Security. 2016 International Conference on Information Science and Security (ICISS), 2016, pp. 1–5, doi: 10.1109/ICISSEC.2016.7885853.

[26] Sharafaldin, I.—Lashkari, A. H.—Ghorbani, A. A.: Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP 2018), 2018, pp. 108–116, doi: 10.5220/0006639801080116.

[27] Reis, B.—Maia, E.—Praça, I.: Selection and Performance Analysis of CICIDS2017 Features Importance. In: Benzekri, A., Barbeau, M., Gong, G., Laborde, R., Garcia-Alfaro, J. (Eds.): Foundations and Practice of Security (FPS 2019). Springer, Cham, Lecture Notes in Computer Science, Vol. 12056, 2020, pp. 56–71, doi: 10.1007/978-3-030-45371-8_4.

[28] Molnar, C.: Interpretable Machine Learning. Lulu.com, 2020.

[29] Axelsson, S.: Intrusion Detection Systems: A Survey and Taxonomy. Technical Report. Chalmers University of Technology, Göteborg, Sweden, 2000.

[30] Buhrmester, V.—Münch, D.—Arens, M.: Analysis of Explainers of Black Box Deep Neural Networks for Computer Vision: A Survey. Machine Learning and Knowledge Extraction, Vol. 3, 2021, No. 4, pp. 966–989, doi: 10.3390/make3040048.

[31] Ho, T. K.: Random Decision Forest. Proceedings of the 3rd International Conference on Document Analysis and Recognition, Vol. 1, 1995, pp. 278–282, doi: 10.1109/ICDAR.1995.598994.

[32] Chen, T.—Guestrin, C.: XGBoost: A Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16), 2016, pp. 785–794, doi: 10.1145/2939672.2939785.

[33] Ke, G.—Meng, Q.—Finley, T.—Wang, T.—Chen, W.—Ma, W.—Ye, Q.—Liu, T. Y.: LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In: Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (Eds.): Advances in Neural Information Processing Systems 30 (NIPS 2017). 2017, pp. 3146–3154.

[34] Prokhorenkova, L.—Gusev, G.—Vorobev, A.—Dorogush, A. V.—Gulin, A.: CatBoost: Unbiased Boosting with Categorical Features. In: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (Eds.): Advances in Neural Information Processing Systems 31 (NeurIPS 2018). 2018, pp. 6638–6648.

[35] Jin, D.—Lu, Y.—Qin, J.—Cheng, Z.—Mao, Z.: KC-IDS: Multi-Layer Intrusion Detection System. 2020 International Conference on High Performance Big Data and Intelligent Systems (HPBD & IS), IEEE, 2020, pp. 1–5, doi: 10.1109/HPBDIS49115.2020.9130573.

[36] Dang, Q. V.: Detecting the Attacks to DNS. In: Antipova, T. (Ed.): Comprehensible Science (ICCS 2021). Springer, Cham, Lecture Notes in Networks and Systems, Vol. 315, 2021, pp. 173–179, doi: 10.1007/978-3-030-85799-8_15.

[37] Settles, B.: Active Learning Literature Survey. Technical Report No. 1648, University of Wisconsin-Madison, Department of Computer Sciences, 2009.

[38] Görnitz, N.—Kloft, M.—Rieck, K.—Brefeld, U.: Active Learning for Network Intrusion Detection. Proceedings of the 2nd ACM Workshop on Security and Artificial Intelligence (AISec'09), 2009, pp. 47–54, doi: 10.1145/1654988.1655002.

[39] Yang, K.—Ren, J.—Zhu, Y.—Zhang, W.: Active Learning for Wireless IoT Intrusion Detection. IEEE Wireless Communications, Vol. 25, 2018, No. 6, pp. 19–25, doi: 10.1109/MWC.2017.1800079.

[40] Deka, R. K.—Bhattacharyya, D. K.—Kalita, J. K.: Active Learning to Detect DDoS Attack Using Ranked Features. Computer Communications, Vol. 145, 2019, pp. 203–222, doi: 10.1016/j.comcom.2019.06.010.

[41] Dang, Q. V.: Active Learning for Intrusion Detection Systems. 2020 RIVF International Conference on Computing and Communication Technologies (RIVF), IEEE, 2020, doi: 10.1109/RIVF48685.2020.9140751.

[42] Abdi, H.—Williams, L. J.: Principal Component Analysis. Wiley Interdisciplinary Reviews: Computational Statistics, Vol. 2, 2010, No. 4, pp. 433–459, doi:

10.1002/wics.101.

[43] Kausar, N.—Samir, B. B.—Sulaiman, S. B.—Ahmad, I.—Hussain, M.: An Approach Towards Intrusion Detection Using PCA Feature Subsets and SVM. 2012 International Conference on Computer and Information Science (ICCIS), IEEE, Vol. 2, 2012, pp. 569–574, doi: 10.1109/ICCISci.2012.6297095.

[44] Xu, X.—Wang, X.: An Adaptive Network Intrusion Detection Method Based on PCA and Support Vector Machines. In: Li, X., Wang, S., Dong, Z. Y. (Eds.): Advanced Data Mining and Applications (ADMA 2005). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 3584, 2005, pp. 696–703, doi: 10.1007/11527503_82.

[45] Wu, Y.—Lee, W. W.—Xu, Z.—Ni, M.: Large-Scale and Robust Intrusion Detection Model Combining Improved Deep Belief Network with Feature-Weighted SVM. IEEE Access, Vol. 8, 2020, pp. 98600–98611, doi: 10.1109/ACCESS.2020.2994947.

[46] Hinton, G. E.: Deep Belief Networks. Scholarpedia, Vol. 4, 2009, No. 5, Art. No. 5947, doi: 10.4249/scholarpedia.5947.

[47] Ranga Suri, R. N. N.—Murty, M. N.—Athithan, G.: Outlier Detection: Techniques and Applications. Intelligent Systems Reference Library, Springer, Cham, Vol. 155, 2019, doi: 10.1007/978-3-030-05127-3.

[48] Lazarevic, A.—Ertoz, L.—Kumar, V.—Ozgur, A.—Srivastava, J.: A Comparative Study of Anomaly Detection Schemes in Network Intrusion Detection. Proceedings of the 2003 SIAM International Conference on Data Mining (SDM), SIAM, 2003, pp. 25–36, doi: 10.1137/1.9781611972733.3.

[49] Eskin, E.: Anomaly Detection over Noisy Data Using Learned Probability Distributions. Proceedings of the Seventeenth International Conference on Machine Learning (ICML-2000), Morgan Kaufmann, 2000, pp. 255–262.

[50] Alhakami, W.—ALharbi, A.—Bourouis, S.—Alroobaea, R.—Bouguila, N.: Network Anomaly Intrusion Detection Using a Nonparametric Bayesian Approach and Feature Selection. IEEE Access, Vol. 7, 2019, pp. 52181–52190, doi: 10.1109/ACCESS.2019.2912115.

[51] Dang, Q. V.: Outlier Detection in Network Flow Analysis. 2018, arXiv: 1808.02024.

[52] Liu, F. T.—Ting, K. M.—Zhou, Z. H.: Isolation Forest. 2008 Eighth IEEE International Conference on Data Mining, 2008, pp. 413–422, doi: 10.1109/ICDM.2008.17.

[53] Zoppi, T.—Ceccarelli, A.—Capecchi, T.—Bondavalli, A.: Unsupervised Anomaly Detectors to Detect Intrusions in the Current Threat Landscape. ACM/IMS Transactions on Data Science, Vol. 2, 2021, No. 2, Art. No. 7, doi: 10.1145/3441140.

[54] Goldstein, M.—Dengel, A.: Histogram-Based Outlier Score (HBOS): A Fast Unsupervised Anomaly Detection Algorithm. In: Wölfl, S. (Ed.): Poster and Demo Track of the 35th German Conference on Artificial Intelligence (KI-2012). 2012, pp. 59–63.

[55] Kriegel, H. P.—Schubert, M.—Zimek, A.: Angle-Based Outlier Detection in High-Dimensional Data. Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'08), 2008, pp. 444–452, doi: 10.1145/1401890.1401946.

[56] Vázquez, F. I.—Zseby, T.—Zimek, A.: Outlier Detection Based on Low Density Models. 2018 IEEE International Conference on Data Mining Workshops (ICDMW), IEEE, 2018, pp. 970–979, doi: 10.1109/ICDMW.2018.00140.

[57] Jyothsna, V.—Rama Prasad, V. V.: A Review of Anomaly Based Intrusion Detection Systems. International Journal of Computer Applications, Vol. 28, 2011, No. 7, pp. 26–35, doi: 10.5120/3399-4730.

[58] Ahmed, M.—Mahmood, A. N.—Hu, J.: A Survey of Network Anomaly Detection Techniques. Journal of Network and Computer Applications, Vol. 60, 2016, pp. 19–31, doi: 10.1016/j.jnca.2015.11.016.

[59] Dang, Q. V.: Studying the Fuzzy Clustering Algorithm for Intrusion Detection on the Attacks to the Domain Name System. 2021 Fifth World Conference on Smart Trends in Systems Security and Sustainability (WorldS4), IEEE, 2021, pp. 271–274, doi: 10.1109/WorldS451998.2021.9514038.

[60] Diro, A. A.—Chilamkurti, N.: Distributed Attack Detection Scheme Using Deep Learning Approach for Internet of Things. Future Generation Computer Systems, Vol. 82, 2018, pp. 761–768, doi: 10.1016/j.future.2017.08.043.

[61] Vinayakumar, R.—Alazab, M.—Soman, K. P.—Poornachandran, P.—Al-Nemrat, A.—Venkatraman, S.: Deep Learning Approach for Intelligent Intrusion Detection System. IEEE Access, Vol. 7, 2019, pp. 41525–41550, doi: 10.1109/ACCESS.2019.2895334.

[62] Protogerou, A.—Papadopoulos, S.—Drosou, A.—Tzovaras, D.—Refanidis, I.: A Graph Neural Network Method for Distributed Anomaly Detection in IoT. Evolving Systems, Vol. 12, 2021, No. 1, pp. 19–36, doi: 10.1007/s12530-020-09347-0.

[63] Yavuz, F. Y.—Devrim, Ü.—Gül, E.: Deep Learning for Detection of Routing Attacks in the Internet of Things. International Journal of Computational Intelligence Systems, Vol. 12, 2018, No. 1, pp. 39–58, doi: 10.2991/ijcis.2018.25905181.

[64] Kang, M. J.—Kang, J. W.: Intrusion Detection System Using Deep Neural Network for In-Vehicle Network Security. PloS ONE, Vol. 11, 2016, No. 6, Art. No. e0155781, doi: 10.1371/journal.pone.0155781.

[65] Brun, O.—Yin, Y.—Gelenbe, E.—Kadioglu, Y. M.—Augusto-Gonzalez, J.—Ramos, M.: Deep Learning with Dense Random Neural Networks for Detecting Attacks Against IoT-Connected Home Environments. In: Gelenbe, E. et al. (Eds.): Security in Computer and Information Sciences (Euro-CYBERSEC 2018). Springer, Cham, Communications in Computer and Information Science, Vol. 821, 2018, pp. 79–89, doi: 10.1007/978-3-319-95189-8_8.

[66] Zheng, L.—Li, Z.—Li, J.—Li, Z.—Gao, J.: AddGraph: Anomaly Detection in Dynamic Graph Using Attention-Based Temporal GCN. Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI'19), 2019, pp. 4419–4425, doi: 10.24963/ijcai.2019/614.

[67] Chaudhary, A.—Mittal, H.—Arora, A.: Anomaly Detection Using Graph Neural Networks. 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), IEEE, 2019, pp. 346–350, doi: 10.1109/COMITCon.2019.8862186.

[68] Kipf, T. N.—Welling, M.: Semi-Supervised Classification with Graph Convolutional Networks. 2016, arXiv: 1609.02907.

[69] Van, N. T.—Sach, L. T.—Thinh, T. N.: Temporal Features Learning Using Autoencoder for Anomaly Detection in Network Traffic. In: Huang, Y. P., Wang, W. J., Quoc, H. A., Giang, L. H., Hung, N. L. (Eds.): Computational Intelligence Methods for Green Technology and Sustainable Development (GTSD 2020). Springer, Cham, Advances in Intelligent Systems and Computing, Vol. 1284, 2021, pp. 15–26, doi: 10.1007/978-3-030-62324-1_2.

[70] Cannady, J. D.: Next Generation Intrusion Detection: Autonomous Reinforcement Learning of Network Attacks. Proceedings of the 23rd National Information Systems Security Conference, 2000, pp. 1–12.

[71] Servin, A.—Kudenko, D.: Multi-Agent Reinforcement Learning for Intrusion Detection: A Case Study and Evaluation. In: Bergmann, R., Lindemann, G., Kirn, S., Pěchouček, M. (Eds.): Multiagent System Technologies (MATES 2008). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 5244, 2008, pp. 159–170, doi: 10.1007/978-3-540-87805-6_15.

[72] Dang, Q. V.—Vo, T. H.: Studying the Reinforcement Learning Techniques for the Problem of Intrusion Detection. 2021 4th International Conference on Artificial Intelligence and Big Data (ICAIBD), IEEE, 2021, pp. 87–91, doi: 10.1109/ICAIBD51990.2021.9459006.

[73] Dang, Q. V.—Vo, T. H.: Reinforcement Learning for the Problem of Detecting Intrusion in a Computer System. In: Yang, X. S., Sherratt, S., Dey, N., Joshi, A. (Eds.): Proceedings of Sixth International Congress on Information and Communication Technology. Springer, Singapore, Lecture Notes in Networks and Systems, Vol. 236, 2022, pp. 755–762, doi: 10.1007/978-981-16-2380-6_66.

[74] Lopez-Martin, M.—Carro, B.—Sanchez-Esguevillas, A.: Application of Deep Reinforcement Learning to Intrusion Detection for Supervised Problems. Expert Systems with Applications, Vol. 141, 2020, Art. No. 112963, doi: 10.1016/j.eswa.2019.112963.

[75] Agarwal, R.—Schuurmans, D.—Norouzi, M.: An Optimistic Perspective on Offline Reinforcement Learning. In: Daumé III, H., Singh, A. (Eds.): Proceedings of the 37th International Conference on Machine Learning (ICML). Proceedings of Machine Learning Research (PMLR). Vol. 119, 2020, pp. 104–114.

[76] Dwibedi, S.—Pujari, M.—Sun, W.: A Comparative Study on Contemporary Intrusion Detection Datasets for Machine Learning Research. 2020 IEEE International Conference on Intelligence and Security Informatics (ISI), 2020, doi: 10.1109/ISI49825.2020.9280519.

[77] Sarhan, M.—Layeghy, S.—Moustafa, N.—Portmann, M.: NetFlow Datasets for Machine Learning-Based Network Intrusion Detection Systems. In: Deze, Z., Huang, H., Hou, R., Rho, S., Chilamkurti, N. (Eds.): Big Data Technologies and Applications (BDTA 2020, WiCON 2020). Springer, Cham, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, Vol. 371, 2021, pp. 117–135, doi: 10.1007/978-3-030-72802-1_9.

[78] McHugh, J.: Testing Intrusion Detection Systems: A Critique of the 1998 and 1999 DARPA Intrusion Detection System Evaluations as Performed by Lincoln Labora-

tory. ACM Transactions on Information and System Security, Vol. 3, 2000, No. 4, pp. 262–294, doi: 10.1145/382912.382923.

[79] Abrar, I.—Ayub, Z.—Masoodi, F.—Bamhdi, A. M.: A Machine Learning Approach for Intrusion Detection System on NSL-KDD Dataset. 2020 International Conference on Smart Electronics and Communication (ICOSEC), IEEE, 2020, pp. 919–924, doi: 10.1109/ICOSEC49089.2020.9215232.

[80] Gharib, A.—Sharafaldin, I.—Lashkari, A. H.—Ghorbani, A. A.: An Evaluation Framework for Intrusion Detection Dataset. 2016 International Conference on Information Science and Security (ICISS), IEEE, 2016, pp. 1–6, doi: 10.1109/ICISSEC.2016.7885840.

[81] Nechaev, B.—Allman, M.—Paxson, V.—Gurtov, A.: Lawrence Berkeley National Laboratory (LBNL)/ICSI Enterprise Tracing Project. Berkeley, CA, LBNL/ICSI, 2004.

[82] Sangster, B.—O'Connor, T. J.—Cook, T.—Fanelli, R.—Dean, E.—Adams, W. J.—Morrell, C.—Conti, G.: Toward Instrumenting Network Warfare Competitions to Generate Labeled Datasets. Proceedings of the 2nd Conference on Cyber Security Experimentation and Test (CSET'09), 2009.

[83] Sharafaldin, I.—Lashkari, A. H.—Ghorbani, A. A.: A Detailed Analysis of the CICIDS2017 Data Set. In: Mori, P., Furnell, S., Camp, O. (Eds.): Information Systems Security and Privacy (ICISSP 2018). Springer, Cham, Communications in Computer and Information Science, Vol. 977, 2018, pp. 172–188, doi: 10.1007/978-3-030-25109-3_9.

[84] Shiravi, A.—Shiravi, H.—Tavallaee, M.—Ghorbani, A. A.: Toward Developing a Systematic Approach to Generate Benchmark Datasets for Intrusion Detection. Computers and Security, Vol. 31, 2012, No. 3, pp. 357–374, doi: 10.1016/j.cose.2011.12.012.

[85] The Communications Security Establishment (CSE) and The Canadian Institute for Cybersecurity (CIC): A Realistic Cyber Defense Dataset (CSE-CIC-IDS 2018). 2018, `https://registry.opendata.aws/cse-cic-ids2018/` [accessed 6-Dec-2020].

[86] MontazeriShatoori, M.—Davidson, L.—Kaur, G.—Lashkari, A. H.: Detection of DoH Tunnels Using Time-Series Classification of Encrypted Traffic. 2020 IEEE International Conference on Dependable, Autonomic and Secure Computing, International Conference on Pervasive Intelligence and Computing, International Conference on Cloud and Big Data Computing, International Conference on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech), IEEE, 2020, pp. 63–70, doi: 10.1109/DASC-PICom-CBDCom-CyberSciTech49142.2020.00026.

[87] Japkowicz, N.—Shah, M.: Evaluating Learning Algorithms: A Classification Perspective. Cambridge University Press, 2011, doi: 10.1017/CBO9780511921803.

[88] Chicco, D.—Jurman, G.: The Advantages of the Matthews Correlation Coefficient (MCC) over F1 Score and Accuracy in Binary Classification Evaluation. BMC Genomics, Vol. 21, 2020, No. 1, Art. No. 6, doi: 10.1186/s12864-019-6413-7.

[89] Hand, D. J.—Till, R. J.: A Simple Generalisation of the Area Under the ROC Curve for Multiple Class Classification Problems. Machine Learning, Vol. 45, 2001, No. 2, pp. 171–186, doi: 10.1023/A:1010920819831.

[90] Ashrapov, I.: Tabular GANs for Uneven Distribution. 2020, arXiv: 2010.00638.

Quang-Vinh Dang received his Bachelor degree from the College of Technology, Vietnam National University Hanoi, Vietnam, and his Ph.D. in computer science from Université de Lorraine, Nancy, France. Currently, he is Lecturer at the Industrial University of Ho Chi Minh City, Ho Chi Minh City, Vietnam. His research interests include machine learning applied in security and finance.