ISSN 2669-2023 (print) / ISSN 2538-8959 (on-line) / https://doi.org/10.47459/cndcgs.2022.7

Network Attack Detection Using Machine Learning Methods

Nataliya ZAGORODNA[1], Mariia Stadnyk[1], Borys LYPA[1], Mykola GAVRYLOV[1], Ruslan KOZAK[1]

[1]Department of Cybersecurity, Faculty of Computer Information Systems and Software Engineering, Department of Cybersecurity, Ternopil Ivan Puluj National Technical University, Ruska 56, 46001, Ternopil, Ukraine

E-mails: [2]zagorodna.n@gmail.com; [2] maria.stadnyk@gmail.com; [3]borislipa699@gmail.com; [4]gavrilovnikolay1999@gmail.com, [5]ruslan.o.kozak@gmail.com

Abstract

This paper presents the result of the study of network intrusion detection using machine learning algorithms. The creation and training of such algorithms is seriously limited by the small number of actual datasets available for public access. The CSE-CIC-IDS2018 data set, used in research, includes 7 subsets of different attack scenarios. Each subset is labeled using a few subtypes of a given attack or normal behavior. That is why the problem of network attack detection has been considered a multiclassification problem. Some of the most popular classifiers will be tested on the chosen data set. Classification algorithms are developed using a standard Python programming environment and the specialized machine learning library Scikit-learn. In the paper, a comparative analysis of the results was performed based on the the application of Random Forest, XGBoost, LR, and MLP classifiers.

KEY WORDS: *network attack, DOS, DDOS, botnet, cybersecurity, machine learning, classification, Random Forest, XGBoost, MLP classifier, LR*

1. Introduction

Information technologies (more generally, ICT, information and communication technologies) have become an integral component of human life now and will definitely occupy a significant place in the future society. ICT has contributed a lot in changing our daily life, like cities are becoming smart, cars are becoming self-driving, call centers can be serviced by robots, etc. Many gadgets are used anywhere and anytime to help people. It is now difficult to find a field in which information technologies have not yet been used. According to [1], we can sum up that the impact of information technologies on our lives and modern society is rather high and continues to increase.

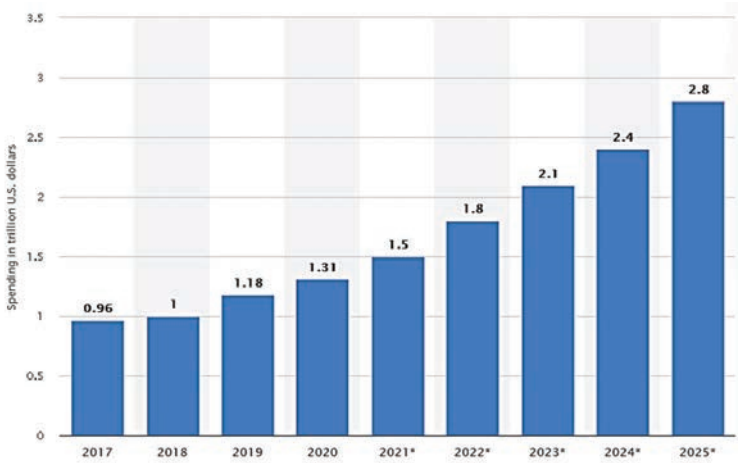

Fig.1. Worldwide spending on digital transformation technologies and services from 2017 to 2025 [1].

The digitization of many aspects of human life and social services became one of the priority directions of the development of Ukraine. In Ukraine, the concept of the development of the digital economy and society of Ukraine

[1] Corresponding author. Tel.: 38096-387-06-78.
E-mail address: maria.stadnyk@gmail.com

has been approved and the necessary legislation base for digitalization has already been adopted. First, the COVID-19 pandemic and then the war in Ukraine caused the rapid transition of various spheres of life to the digital world. People start to communicate, study, buy, work, and get different services online. But, at the same time the number of cyber threats is also growing. People are used somehow to defend themselves in the real world, but they are much more vulnerable in the virtual world.

According to [2] the COVID-19 pandemic has led to an increase in cybercrime by 600%. At the same time, worldwide cyber-crimes are estimated to be around $10.5 trillion annually by 2025. Network attacks such as DOS, DDOS, botnet etc. take a significant place among the attacks. For example, according to [3] the popularity of DDOS attacks is breaking records. In November 2021 the largest DDOS was recorded. Web-application attack can be very critical for business and can cause from leakage of sensitive information to reputation losses. So, a very important scientific problem to be solved is the possibility of identifying network cyber-attacks in time and taking appropriate countermeasures.

2. The Theoretical Background

The purpose of creating a network attack detection system is to collect information about traffic and its specific characteristics, on the basis of which it is possible to detect an attack. Accordingly, this is a classification problem, which involves learning on prelabeled data sets, in the terminology of machine learning, supervised learning.

To implement the optimal classification algorithm, it is necessary to perform the corresponding steps in sequence, which are presented in Fig. 2.

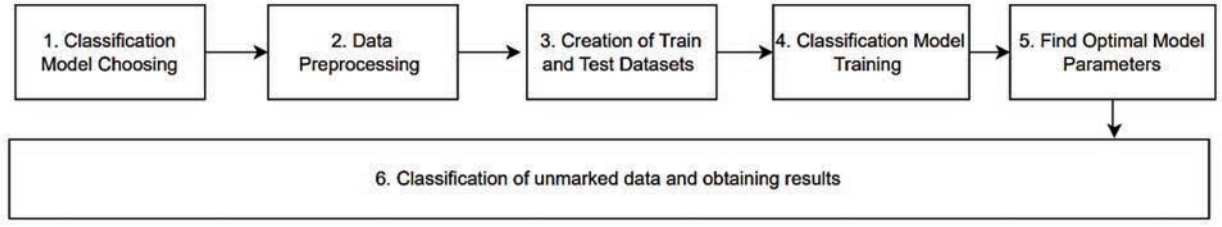

Fig.2. Schematic representation of the stages of the classification algorithm

For selecting the classification model, it is necessary to consider the parameters in the dataset and understand the nature of the occurrence of the phenomenon or process and characteristics that can be diagnostic. Classification models used to detect network anomalies include support vector machines (SVM) [4], neural networks [5], and decision trees and their modifications [6]. The authors in [7] make a detailed review of existing machine learning methods (supervised, unsupervised learning, and with reinforcement) to detect various types of network attacks based on artificially created data sets, such as CSE-CIC-IDS2018 and NSL-KDD. Comparative tables with the methods and their advantages are provided [7]. For example, based on the NSL-KDD dataset, the K-Means algorithm was used to detect a DDoS attack in order to classify unlabeled data using prelabeled data. This algorithm shows high performance in detection with a low percentage of labeled data, but one of the main disadvantages is the test time being too long.

Based on previous research work, it was chosen to use four classification models: Random Forest (RF), XGBoost, Linear Regression (LR), Multilayer Perceptron (MLP). One of the newest algorithms in the decision tree family is XGBoost. It was decided to use it in the work for comparison with RF as the widely used model for the network attacks classification. XGBoost is one of the representatives from the decision tree family of machine learning algorithms. On the basis of the application of gradient boosting and parallelization during the construction of the tree, the optimal speed of the algorithm was achieved compared to the existing algorithms. This modification increases not only the speed of algorithm execution, but also the accuracy of the classification model.

3. Dataset investigation

In the investigation, the CSE-CIC-IDS2018 dataset obtained by the Canadian Cyber Security Institute (CIC) using Amazon Web Services (AWS) [8] to detect network attacks was used. This data set contains seven attack scenarios: Heartbleed, Brute-Force, Botnet, DoS and DDoS, attacks on web services, and insider attack emulation. Fifty working machines were used to perform the attacker's actions, and five departments were used to record the attack, consisting of 420 machines and 30 servers, which reflect the real victim of the attack. The data set obtained includes 80 registered characteristics obtained during traffic congestion analysis using CICFlowMeter-V3 [9,10].

A brief description of the attack given in the CSE-CIC-IDS2018 data set [8]:

- *Brute force* is a method of solving a mathematical problem, which means a search for frequently used passwords or symbols to obtain the correct password. The dataset uses the Python tool Patator for the brute-force implementation. The attacker using Kali Linux tries to find the password for the FTP and SSH modules running on the victim machine, Ubuntu 14.0. The list, which contains 90 million words, is used to sift through possible passwords.
- *A botnet* is a certain number of devices connected to the Internet that run one or more botnet programs. Two botnet networks are used to implement the Botnet scenario: Zeus and Ares. The data set contains data about computers that send screenshots every 400 seconds.
- *Denial of Service (DoS)* uses specialized tools (for example: Slowloris Perl) to attack a web server from a single machine with minimal bandwidth. Heartbleed is an attack related to a flaw in the OpenSSL cryptographic library that allows unauthorized reading of server or client memory, including the desire to obtain the server's private key. An attacker could use this vulnerability to steal encrypted information with SSL/TLS protocols. This, in turn, causes vulnerabilities in e-mail, private virtual networks, privacy of web applications, and SAAS [11]. This data set contains usage data for Heartleech, one of the most well-known tools to exploit the Heartbleed vulnerability.
- *Distributed Denial of Service (DDoS)* is an attack that is carried out from a large number of hosts simultaneously. The attack was implemented using the High Orbit Ion Cannon (HOIC) utility, which executes the attack using 4 different computers.
- To emulate *attacks on web applications* we have used the tool DVWA, Damn Vulnerable Web App. It is the victim's own web application and was created to test the skills of security professionals. The first step of the attack is to scan the web application through a web vulnerability scanner; the next steps are various web attacks, including SQL injection, XSS scripting, and unauthorized file uploads.
- Vulnerable software was used to carry out *network penetration*. The attack scenario includes two steps: the victim receives a malicious document as an email attachment; the attack is carried out using the Metasploit Framework – a utility that allows you to develop and apply malicious code on the victim's machine, after the attack is completed, the attacker will be able to control the victim's computer.

The list of implanted attacks with detailed information such as duration, software, and devices used is presented in Table 1.

Table 1.

List of realized attacks and their duration

Attack	Software	Duration	Attackers	Victim
Brute force method	FTP – Patator, SSH – Patator	1 day	Kali Linux	Ubuntu 16.4 (Web Server)
Botnet attack	Ares: remote console access, file up-load. Screenshots and key captures	1 day	Kali Linux	Windows Vista, 7, 8.1, 10 (32-bit) and 10 (64-bit).
DoS	Hulk, GoldenEye, SlowLoris, and Slowhttptest	1 day	Kali Linux	Ubuntu 16.4 (Apache)
DoS	Heartleech	1 day	Kali Linux	Ubuntu 12.04 (Open SSL)
DDoS	Low orbit ion Cannon (LOIC) for UDP, TCP, and HTTP requests	2 days	Kali Linux	Windows Vista, 7, 8.1, 10 (32-bit) and 10 (64-bit).
Web attack	Damn Vulnerable Web App (DVWA) and in-house selenium framework (XSS and Brute-force)	2 days	Kali Linux	Ubuntu 16.4 (Web Server)
Network penetration	First level: Dropbox download on a Windows machine. Second level: Nmap and portscan	2 days	Kali Linux	Windows Vista or Macintosh

Each type of attack and corresponding network characteristics are represented in a form of separate data set. Each data set is saved as a CSV file containing more than 80 characteristics. Some of them are presented and described in the detailed primary source [8]. For a machine learning algorithm, this is a significant amount, so the feature space was reduced in the paper to increase the classification accuracy.

4. Pre-processing

The first step in any machine learning technique is data preprocessing to avoid not supported data and further classification mistakes. During this step, the following actions were performed:

- characteristics that are not used by the machine learning model, for example, the 'TimeStamp' column, have been removed;
- missing values ("Infinity" or "NaN") were replaced by the average value for each column;
- The values of each column, except for categorical columns (such as 'Destination Port' and "Protocol") were replaced by the difference between the column value and the average value of this column;
- missing values were replaced by '-1' for categorical columns;
- Data were converted to standard data formats.

It is difficult to obtain the data set for research on the basis of real events, because not all networks have the possibility to log a huge number of traffic parameters. Usually, the detection of a network attack takes place after the fact of its occurrence. The dataset studied is artificially created, with the aim of detecting the dependencies of various traffic parameters and the corresponding attack on the network. It should be noted that the data in the dataset is also unbalanced, which poses a certain issue within the training of the machine learning model. Figs. 3 and 4 show the distribution by classes of DoS and brute force attacks, which indicate data imbalance. Each network attack data set is characterized by a certain level of imbalance.

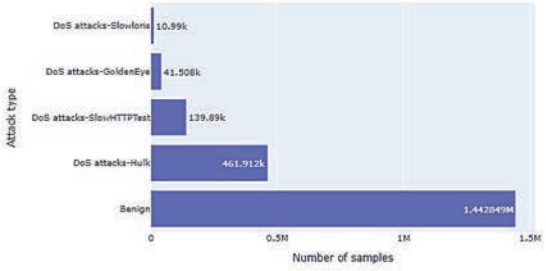

Fig.3. DoS attacks data class distribution.

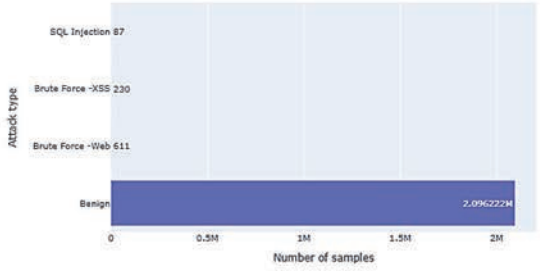

Fig. 4. Brute Force data class distribution

Under- and over-sampling techniques can be used to balance data. Each of them artificially increases or decreases the amount of sample in the data set. One of the under sampling methods is the extraction of similar values that were previously detected using the KNN algorithm. Classic oversampling means increasing the data set using randomly generated data, but the importance of the characteristics is lost, and, accordingly, the classification accuracy decreases. To avoid this, the SMOTE oversampling method [13, 14] was used in the work. Using this clever algorithm, artificial data is generated that reflects original characteristics in the minority class (for example, SQL injection in the data set of the brute force attack). As mentioned above, each data set contains 80 characteristics of network traffic. This number is extremely significant for a machine learning algorithm. Also, some of the network traffic features are not significant and do not reflect the relationship of this characteristic with vulnerability. Therefore, in the work, the space of diagnostic parameters was reduced (up to 5 characteristics) according to each type of attack. The results are presented in Figs. 5-10.

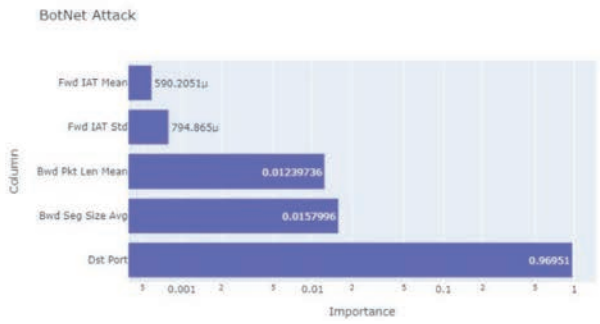

Fig.5. Feature Importance in the Botnet Attack Data Set

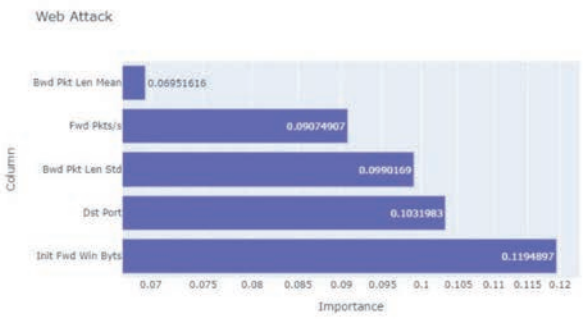

Fig. 6. Importance of features in the Web attack data set

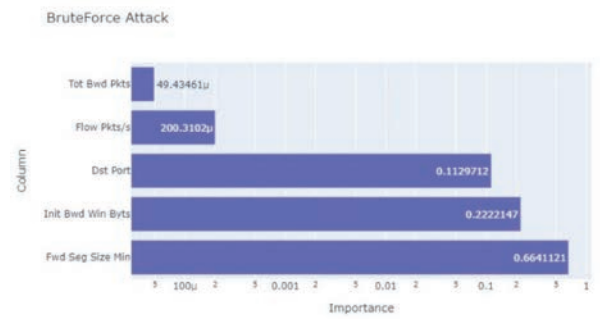

Fig. 7. Importance of features in the brute-force dataset

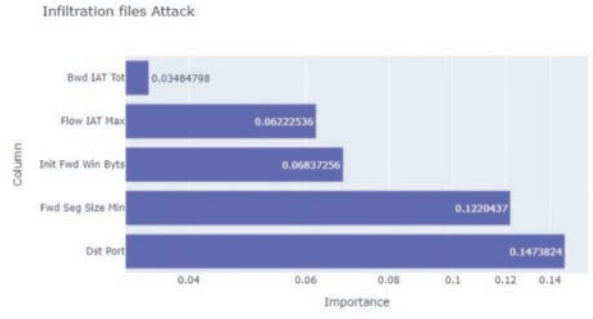

Fig. 8. Importance of characteristics in the infiltration attack data set

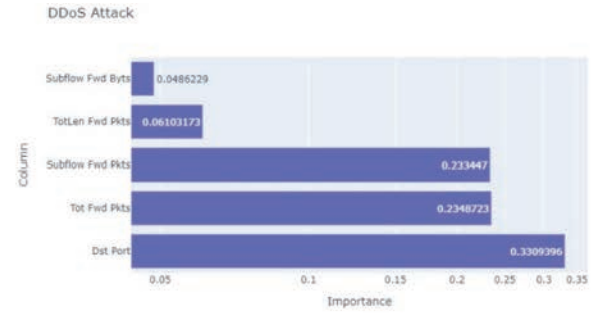

Fig.9. Feature importance in the DDoS attack dataset

Fig. 10. Importance of features in the DoS attack dataset

The results show that the most important features are: Dst Port (Recipient port number), because an incorrectly specified port and its traversal value are signs of an attack, Flow IAT (deviation in time between two streams) and its characteristics, the total size of packets in both directions, average number of bits/packets per sub stream in both directions.

5. The Classification Results

The entire CSE-CIC-IDS2018 dataset is split into seven separate datasets with respect to each type of attack scenario. Each raw (instance) of the data set is labelled with either some attack type (see Fig. 3,4) or “benign” (normal behavior). As each dataset comprises more than two labels, network attack detection based on an artificially constructed dataset is considered as a multiclassification problem. The most popular classification models such as RF, XGBoost, MLP, and LR were chosen to solve this problem.

To estimate the performance of each classifier, every dataset has been split into two parts named training and test sets by 80/20 schema. The optimal values of the parameters of each model have been found using the GridSearchCV function GridSearchCV (Table 2).

Table 2.

Optimal parameters of the machine learning models

Model	Parameters
RF	*'max_depth': 50, 'min_samples_leaf': 5, 'min_samples_split': 5, 'n_estimators': 100*
XGBoost	*'learning_rate': 0.01, 'max_depth': 10, 'n_estimators': 100*
LR	*'C ': 100, 'penalty ': 'l2','solver': 'newton cg'*
MLP	*'Activation': relu,'solver': 'adam','learning_rate': 'adaptive','max_iter': '500'*

For quality estimation of the multiclassification model, we used K-fold CV (cross-validation). According to the method, the data set is divided into k equal parts. The estimation procedure runs k times with each of the the the k subsets subsets subsets obtained assigned to the test set. We used k=5, so each time test set corresponded to 20% of all data. This verification technique generalizes the efficiency results based on statistical analysis. The accuracy of the model in the test set is compared with the accuracy of the model in the training set. If the results are approximately the same, then the model can be considered validated [4]. Accuracy (A), precision (P), and recall (R) were chosen as classification quality assessment metrics.

Generalized results are presented in Table 4 in correspondence to each type of attack. The calculated values are actually the averages of each metric found based on the results of the k-fold cross-validation method.

Table 4.

Estimation of the quality of classification models

Classification model	Brute Force Dataset			Botnet Dataset		
	A, %	P, %	R, %	A, %	P, %	R, %
RF	0.982	0.994	0.992	0.981	0.985	0.987
XGBoost	0.998	0.984	0.982	0.965	0.974	0.921
LR	0.879	0.865	0.780	0.899	0.836	0.862
MLP	0.965	0.987	0.978	0.973	0.924	0.964
	DoS dataset			DDoS data set		
RF	0.965	0.987	0.965	0.968	0.987	0.982
XGBoost	0.987	0.989	0.985	0.988	0.989	0.982
LR	0.825	0.854	0.852	0.834	0.847	0.856
MLP	0.879	0.884	0.881	0.881	0.885	0.890
	Web attack data set			Network Penetration Data Set		
RF	0.983	0.971	0.977	0.971	0.968	0.956
XGBoost	0.962	0.954	0.987	0.964	0.945	0.944
LR	0.879	0.820	0.861	0.856	0.842	0.845
MLP	0.921	0.911	0.903	0.891	0.824	0.846

As most of models based on decision trees have serious drawbacks connected with model overfitting, some measures have been taken to prevent overtraining of models.

Conclusions

The XGBoost classification model is a modified decision tree model similar to RF. So, the classification quality indicators using these models are practically the same, which is not surprising. However, the execution time of XGBoost is longer, which is an issue when applying this model in real-time modes.

The quality of the classification decreases while using the same metrics for the Web attack dataset and Network penetration dataset, because it would be better to use the methods of detecting anomalies in the traffic to detect this type of attack. That can help to perform prediction or registration of the current attack in real time.

In further research, it would be a great idea to apply a combination of under- and over-sampling techniques and evaluate the classification accuracy on balanced data.

Acknowledgements

The CSE-CIC-IDS2018 dataset was used for the study, which is in the public domain and can be redistributed or published in any form, provided the original source is acknowledged.

References

1. Statista Data: https://www.statista.com/statistics/870924/worldwide-digital-transformation-market-size/\
2. Purplesec Data: https://purplesec.us/resources/cyber-security-statistics/
3. 2022 A10 Networks DDoS Threat Report: https://www.a10networks.com/resources/reports/2022-ddos-threat-report/
4. **Xiaoqing G., Hebin G., Luyi C.** "Network intrusion detection method based on Agent and SVM." *2010 2nd IEEE international conference on information management and engineering*. IEEE, 2010. p. 399-402.
5. **Lippmann, Richard P., Robert K. Cunningham**. "Improving intrusion detection performance using keyword selection and neural networks." *Computer networks* 34.4 (2000): 597-603.
6. **Moustafa, Nour, and Jill S.** "The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set." Information Security Journal: A Global Perspective 25.1-3 (2016): 18-31.
7. **Wang S**. et al. "Machine learning in network anomaly detection: A survey." IEEE Access 9 (2021): 152379-152396.
8. Canadian Institute for Cybersecurity Data: https://www.unb.ca/cic/datasets/ids-2018.html
9. **Lashkari, Arash H.**, et al. "Characterization of tor traffic using time based features." ICISSp. 2017.
10. **Draper-Gil, Gerard,** et al. "Characterization of encrypted and vpn traffic using time-related." Proceedings of the *2nd international conference on information systems security and privacy* (ICISSP). 2016.
11. The Heartbleed Bug Data: http://heartbleed.com/
12. Stratosphere Lab Data: https://www.stratosphereips.org/datasets-overview
13. **Chio C.** Machine Learning and Security / C. Chio, D. Freeman., 2018, p.125-180.
14. **Chawla, Nitesh V.**, et al. "SMOTE: synthetic minority over-sampling technique." *Journal of artificial intelligence research* 16 (2002): 321-357.
15. **Scarfone, Karen, Peter M**. "Guide to intrusion detection and prevention systems (idps)." NIST special publication 800.2007 (2007): 94.