2019 International Russian Automation Conference (RusAutoCon)

Attack Detection in Enterprise Networks by Machine Learning Methods

Nadezhda Bakhareva
Department of Informatics and Computer Engineering
Povolzhskiy State University of Telecommunications and Informatics
Samara, Russia
bahareva-nf@psuti.ru

Alexander Shukhman
Department of Geometry & CS
Orenburg State University
Orenburg, Russia
shukhman@gmail.com

Artem Matveev
Department of Computer Safety
Orenburg State University
Orenburg, Russia
artemi645@gmail.com

Petr Polezhaev
Department of Computer Safety
Orenburg State University
Orenburg, Russia
newblackpit@mail.ru

Yury Ushakov
Department of Geometry & CS
Orenburg State University
Orenburg, Russia
ushakov@unpk.osu.ru

Leonid Legashev
Science Library
Orenburg State University
Orenburg, Russia
silentgir@gmail.com

***Abstract*—This paper proposes algorithms for detecting attacks in enterprise networks based on the analysis of network traffic. The CICIDS2017 dataset was used to compare machine learning methods for binary classifying (attack or regular traffic), as well as for multiclass classification to identify the classes of typical attacks such as DoS/DDoS, PortScan, BruteForce, WebAttack, Bot and Infiltration. The balanced accuracy score is used as the main metric for assessing the accuracy of classification. The main advantage of this metric in adequately estimating the accuracy of classification algorithms, considering the strong imbalance in the number of labeled records for each class of dataset. As a result of the experiment, it was found that the CatBoost and LightGBM algorithms work well for both binary classification and multiclass classification of malicious traffic into several attack groups.**

Keywords—corporate network security, traffic analysis, classification, machine learning, attack detection

I. Introduction

Detection of network attacks is one of the urgent problems in the field of information security. Its solution is crucial for enterprise networks.

Active attack prevention tools like antiviruses, firewalls, and intrusion detection systems are usually used to detect DDoS (Distributed Denial of Service) attacks, network worms, port scanning and other attacks. Unfortunately, only active attack prevention tools are not enough, so passive tools (for example, Snort) are used to complement them.

Typical intrusion detection systems use signature rules to check if the network packets correspond to anomalous patterns from the database. However, such systems cannot detect new intrusion methods. In addition, the use of signature bases of large volume negatively affects the performance of the intrusion detection system.

The data for the intrusion detection system are network traffic represented as a sequence of network packets. The raw data are usually preprocessed for the further analysis including aggregation over a certain time interval and normalization to identify a specific feature set.

Thus, the urgent problem is the fast classification of traffic for the detection of network attacks. This problem can be formalized either as a binary classification (normal or abnormal traffic) or as a more complex multi-class classification problem, when abnormal traffic is in turn classified into pre-selected groups of attacks.

Recently, the machine learning methods are actively used to solve problems of network traffic classification.

A significant number of publications are devoted to the opportunities of using deep learning methods for detection and classification of network attacks. In paper [1] two attack classification problems are considered: binary classification and classification into 4 classes of attacks. The authors use recurrent neural networks to classify a large amount of data. As a result, for binary classification an accuracy of less than 0.1% of errors was archived. For classification by type of attack it was 0.5%.

In paper [2] to detect DDoS attacks recurrent neural networks, including LSTM (Long Short-Term Memory) networks, were compared with ordinary random forests method. LSTM networks showed the highest accuracy – 98.4% of correctly detected attacks. In paper [3] authors consider a neural network with autoencoder and stochastic algorithm for determining activation threshold. This method in-creased the attack detection accuracy up to 88.65% on the NSL-KDD dataset. The paper [4] proposes the system for detecting and classifying both known and un-known anomalies into 4 classes. The optimal architecture of neural network was determined experimentally. The paper [5] considers the opportunity of automatic packet clustering for detecting anomalies in

The research work was funded by RFBR, according to the research projects No. 16-29-09639, 18-07-01446, 18-37-00460.

978-1-7281-0265-8/19/$31.00 ©2019 IEEE

corporate networks. Large clusters with high density, as well as small or sparse clusters are considered abnormal. Further binary classification algorithms are trained on this data. It is possible to get accuracy 88% on the NSL-KDD dataset.

In paper [6] the classification of attacks in IEEE 802.11 wireless networks is considered. Attacks are classified into 3 classes by multilayer autoencoder. In [7] convolutional neural networks are used to classify malicious traffic. The idea is in transformation of raw traffic data into images, which are recognized by convolutional networks. In this case, the accuracy of attack detection reaches 99.41%. In [8] deep belief networks are used to detect hardly identifying types of attacks – port scanning and vulnerability search. They combine both approaches of supervised and unsupervised learning. The paper [9] proposes the method of attack binary classification based on fuzzy C-means clustering approach. Partial manual labeling for a small part of the training data is used to improve the accuracy of the algorithm.

A detailed comparison of various machine learning algorithms used in computer security systems is given in paper [10]. The authors consider three problems: intrusion detection, malware analysis and spam detection. Conclusions – for each problem it is better to apply its own methods, which require continuous training and careful adjustment of parameters. In paper [11] clustering methods for intrusion detection based on C-means are considered.

In modern time hybrid approaches become very popular. In [12] authors consider three classifiers: decision tree, support vector machine (SVM), and a combination of these. The hybrid classifier includes two stages. First, the test data are passed to decision trees, which generated a node information as numbers of leaves. Then the test data with node information are processed by the SVM, which output is a classification result. If these three classifiers give different results, the ultimate result is based on weight voting.

In [13] the set of three neural networks and SMV is considered. The output value is a weighted sum of output values of these classifiers. The weights are calculated on the basis of the mean square error (MSE) value. In [14] the authors propose to use the output values of neural networks as the input values for the procedure of weight voting and majority voting. In [15], an individual neural network was constructed for detecting each of the three types of DDoS attacks, which were conducted with the use of protocols TCP, UDP, and ICMP.

In [16] to detect DoS attacks it was proposed to use an approach, which combines the method of normalized entropy for calculating the feature vectors and SVM for their analysis. In [17] authors suggest the approach based on combination of neural, immune and neuro-fuzzy classifiers.

However, most studies use obsolete datasets, such as DARPA 97 and NSL-KDD. In order to achieve high classification accuracy, performance is often neglected. Methods that allow classification on more than 4 classes of attacks are also not investigated.

In our study, several methods of machine learning were compared for both binary and multi-class classification of network traffic by 6 types of attacks based on the latest data set CICIDS2017.

II. Dataset CICIDS2017

As part of this study, the network traffic classifiers were trained on the labeled Intrusion Detection Evaluation Dataset (CICIDS2017) [18]. This dataset differs from previously created by community datasets, because it contains a number of modern attacks that have appeared recently.

All attacks of this datasets were preliminary combined into six groups:

– DoS/DDoS – the denial of service to a server or network resource. This group includes the following attacks: DoS GoldenEye, DoS Hulk, DoS Slowhttptest, DoS slowloris, Heartbleed.

– PortScan – the attacks based on port scanning.

– BruteForce – the attacks based on exhaustive search of passwords, pages or paths on the Internet resource. It also includes FTP Patator and SSH Patator attacks.

– WebAttack – the attacks on Web-servers (SQL-injections, Cross-Site-Scripting (XSS), password brute force over HTTP).

– Bot – the botnet attacks to steal data, send spam, or an attacker can gain access to hacked devices and their connections.

– Infiltration attacks – installing backdoors through vulnerable software such as Acrobat Reader.

In addition, the data were preprocessed using Pandas library:

– All csv files of the dataset for different periods (days of the week) were concatenated into single DataFrame table.

– The starting and the ending spaces in the column names were removed. The labels of “Benign” traffic samples were renamed to “Normal” (ordinary traffic without attacks).

– The columns with the same values for all samples and the column “Destination port”, the rows with missing values in the «Flow Bytes/s» or «Flow Packets/s» columns were removed.

– The remaining missing values were filled with zeros.

– All columns, except for the target “Label”, were standardized (for each column the mean value was subtracted, and the resulting value was divided by the standard deviation).

III. Binary classification of traffic

The following classifiers were trained on the preprocessed CICIDS2017 dataset:

– CatBoost – the algorithm of gradient boosting on decision trees from the Yandex library with the same name;

– LogisticRegression – the logistic regression algorithm (scikit-learn library);

– LinearSVC – the linear SVM (Support Vector Machine) classifier (scikit-learn library);

– LightGBM – the gradient boosting algorithm on decision trees from Microsoft library with the same name.

Two classes were selected for binary classification of traffic samples:

– Attack – the presence of an attack (any of the six attack groups mentioned in previous section), the number of samples is 556556.

– Normal – no attack (2271320 samples).

Obviously, in this case, the classes are imbalanced. Therefore, it is preferable to use model quality metrics that show efficiency as opposed to accuracy (the proportion of correctly predicted class labels): balanced accuracy [19] or ROC AUC (Receiver Operating Characteristic Area Under Curve) [20].

50% of the randomly selected samples of the dataset (preprocessed for binary classification) were used for training. The remaining 50% were used as the final test data.

The hyperparameter tuning was carried out for each classification algorithm (using GridSearchCV with the 5-fold cross-validation) on the training data. The considered hyperparameters and their values are shown in Table 1.

In Table 2, for algorithms with the best hyperparameter combinations the values of following metrics are presented:

– balanced accuracy, calculated for cross-validation (mean value for five folds on training data) and for test data;
– F1-score, precision and recall [21] for test data;
– ROC AUC for test data.

All metric values were rounded to five digits after the decimal point.

The training time and the prediction time for each algorithm are presented in the last two columns of the table. The prediction time was measured on the test data (706969 samples). CatBoost and LightGBM were executed on the GPU, the rest – on the CPU. All experiments were carried out using Google Colaboratory.

TABLE I. THE HYPERPARAMETERS FOR CLASSIFICATION ALGORITHMS AND THEIR VALUES

Algorithm	**Hyperparameter**	**Values**
CatBoost	iterations – the number of iterations (the number of decision trees)	8, 16, 32, 64, 128, 256, 512
	learning_rate – the learning rate	0.01, 0.1, 0.5, 1
	depth – the depth of decision trees	4, 8, 14, 16
LightGBM	learning_rate – the learning rate	0.01, 0.05, 0.1, 0.5, 1
	n_estimators – the number of decision trees	8, 16, 32, 64, 128
	num_leaves – the number of leaves in decision trees	8, 16, 32, 64, 128
LinearSVC	loss – the loss function	'hinge', 'squared_hinge'
	C – the regularization parameter	0.01, 0.1, 1, 10, 100
LogisticRegression	solver – the method for solving the optimization problem	'sag', 'newton-cg', 'lbfgs'
	C – the regularization parameter	0.01, 0.1, 1, 10, 100

TABLE II. THE BEST HYPERPARAMETERS FOR CLASSIFICATION ALGORITHMS AND THEIR QUALITY (BINARY CLASSIFICATION)

Algorithm with the best hyperparameters	CV balanced accuracy	Balanced accuracy	F1-score	Precision	Recall	ROC AUC	Training time, mm:ss	Prediction time, s
CatBoost **depth: 16, iterations: 512, learning_rate: 0.1, loss_function: "Logloss"**	**0.99924**	**0.999000**	**0.999 04**	**0.999 04**	**0.999 04**	**0.999 97**	**10:14**	**7.25**
LightGBM learning_rate: 0.1, n_estimators: 64, num_leaves: 128	0.99 898	0.998 81	0.998 99	0.998 99	0.998 99	0.999 94	00:28	2.46
LinearSVC C: 10, loss: "hinge"	0.92 738	0.874 97	0.929 82	0.929 44	0.930 98	0.977 25	52:47	0.40
LogisticRegression C: 1, solver: "newton-cg"	0.91 776	0.885 49	0.931 69	0.931 35	0.932 19	0.979 44	25:07	**0.10**

The analysis of Table 2 shows that the Cat-Boost and LightGBM algorithms provide the maximal balanced accuracy (calculated on training and test data). However, the CatBoost algorithm is slightly better than the LightGBM algorithm for all quality metrics. The rest of the algorithms show the slightly worse balanced accuracy, greater than 0.87.

It should be noted that the training time for the CatBoost is about 10 minutes, but the training time for the LightGBM is about 28 seconds. The prediction time for the test set is approximately 7.25 seconds for the CatBoost vs 2.46 seconds for the LightGBM with the close accuracy of the algorithms.

It can be seen by precision and recall metrics that both algorithms have a small enough proportion of false positive and false negative errors.

Figure 1 shows the ROC curves for binary classification algorithms. The best algorithms have the largest area under the curves. In this case, they are the CatBoost and LightGBM.

They have almost identical curves that are not distinguished on the graph.

The confusion matrix for the CatBoost algorithm is shown on Figure 2. It is calculated on the test data.

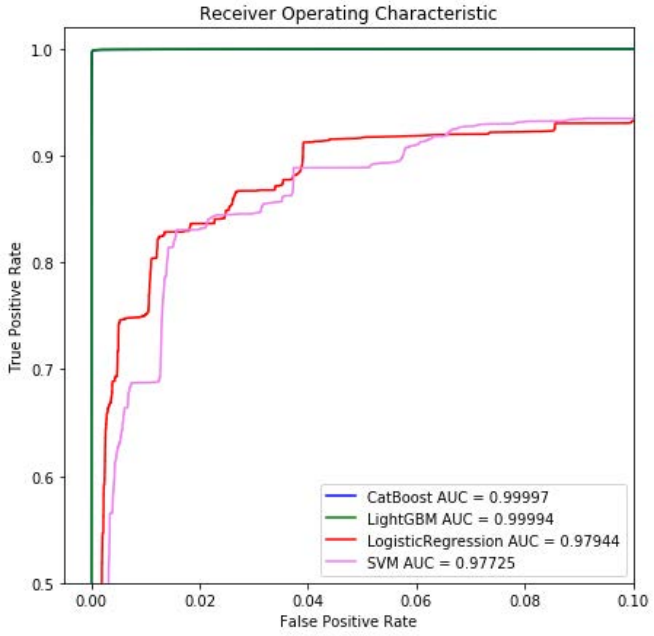

Fig. 1. The ROC curves for binary classification algorithms

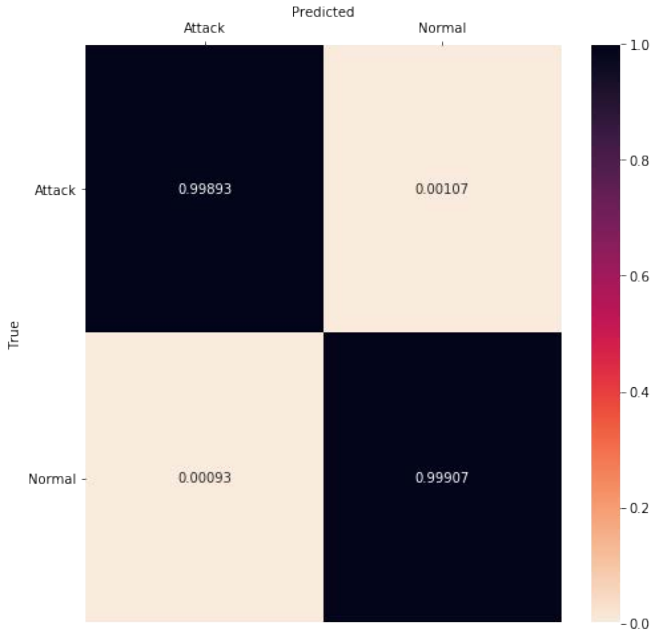

Fig. 2. The confusion matrix for the CatBoost algorithm

It can be seen that the CatBoost algorithm provides 0.093% of false positive and 0.107% of false negative errors, which is quite acceptable. It is necessary to conduct an additional analysis using further classification of traffic into the main groups of attacks. So, it can exclude a part of false positive errors.

In addition, extra analysis tools can be used (for example, signature correlation rules) to exclude the part of false positive and false negative errors.

Next, we consider the classification of attacks into six groups.

IV. Multi-class Classification of Attacks

Table 3 shows the number of labeled samples of each group of attacks taken from the original dataset. The samples with the "Normal" label were excluded.

For multi-class classification, the same algorithms are used as for binary classification. Similarly to binary classification, the dataset was divided into training and test parts in a percentage ratio of 50% - 50%. Also, the same hyperparameters of algorithms were tuned (see Table 1) using cross-validation. The balanced accuracy was the main metric. The F1-score, precision and recall were used as additional metrics.

TABLE III. The Number of Labeled Samples of Each Group of Attacks

Group of attacks	Number of samples
DoS/DDoS	379748
PortScan	158804
BruteForce	13832
WebAttack	2180
Bot	1956
Infiltration	36

Table 4 presents the results of the algorithms experimental study. Analysis of this table shows that the best algorithm is the CatBoost, which provides balanced accuracy on test data, equal to 0.93558. Also, sufficiently good accuracy is shown by the LightGBM algorithm (0.93175). The remaining two algorithms show the worst results.

In addition, it should be noted that the CatBoost algorithm leads in other metrics: the F1-score, precision and recall.

The Analysis of values of the balanced accuracy metric, computed using cross-validation on training data and on test data, shows that for multi-class classification we have the greater overfitting effect than for the binary classification.

The training time of the CatBoost algorithm is about 4 minutes, but the LightGBM algorithm needs 1.5 minutes for training. At the same time, the prediction time for the test data for the CatBoost is 1.52 seconds against 16.4 seconds for the LightGBM. Perhaps it is due to the internal specifics of these algorithms.

TABLE IV. THE BEST HYPERPARAMETERS FOR CLASSIFICATION ALGORITHMS AND THEIR QUALITY (MULTI-CLASS CLASSIFICATION)

Algorithm with the best hyperparameters	CV Balanced Accuracy	Balanced Accuracy	F1-score	Precision	Recall	Training time, mm:ss	Prediction time, s
***CatBoost* depth: 14, iterations: 512, learning_rate: 1**	**0.99975**	**0.93558**	**0.99985**	**0.99985**	**0.99985**	**04:07**	**1.52**
LightGBM learning_rate: 0.01, n_estimators: 128, num_leaves: 128	0.99958	0.93175	0.99964	0.99964	0.99964	01:24	16.40
LinearSVC C: 1, loss: "squared-hinge"	0.99890	0.91610	0.99899	0.99899	0.99900	10:06	0.22
LogisticRegression C: 1, solver: "newton-cg"	0.99855	0.91829	0.99900	0.99900	0.99901	14:23	0.06

Figure 3 shows the confusion matrix for the CatBoost algorithm.

Analysis of this figure leads us to the following conclusions:

a) the CatBoost algorithm accurately detects Bot (99.426%), BruteForce (100%), DoS/DDoS (99.993%), PortScan (99.980%), WebAttack (99.076%) attacks;

b) the CatBoost quite often misclassifies Infiltration attacks into DoS/DDoS (12.5%) or WebAttack (25%) classes.

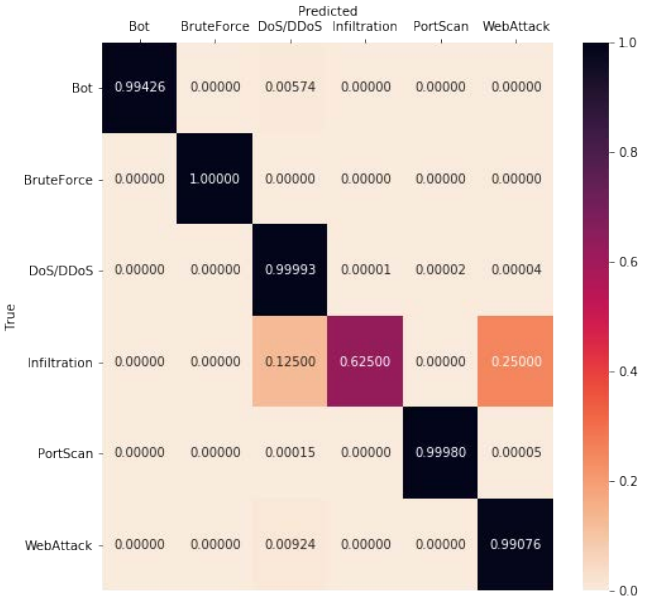

Fig. 3. The confusion matrix for the CatBoost algorithm

Figure 4 shows the confusion matrix for the LightGBM algorithm.

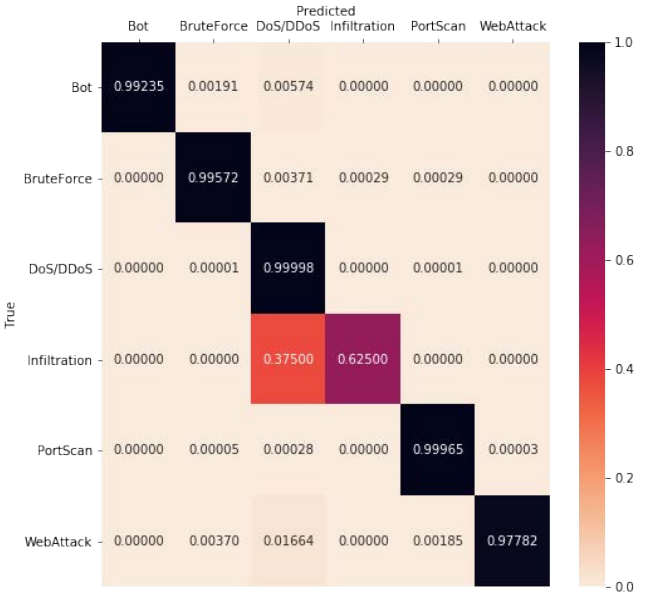

Fig. 4. The confusion matrix for the LightGBM algorithm

Analysis of Figure 4 allows the following conclusions:

a) the LightGBM algorithm accurately detects Bot (99.235%), BruteForce (99.572%), DoS/DDoS (99.998%), PortScan (99.965%), WebAttack (97.782) attacks;

b) the LightGBM algorithm quite often mis-classifies Infiltration attacks into DoS/DDoS attacks (37.5%).

Perhaps the CatBoost and LightGBM algorithms are wrong in classifying Infiltration attacks due to their similarity to other types of attacks.

V. Conclusion

The CatBoost and LightGBM algorithms work well for the binary classification of the network traffic to determine the presence of attacks. They also perform well in the multi-class classification of the malicious traffic into several groups of attacks. The CatBoost algorithm is slightly ahead of the LightGBM algorithm in accuracy and other metrics, but it requires much more time for training (up to 20 times in the case of binary classification and approximately 3 times in multi-class classification).

These algorithms can be used to identify new types of attacks, which are modifications of the existing ones. Especially it is important when the traditional signature methods cannot cope with such modified attacks. On the other hand, signature methods are necessary for accurate identification of known attacks. Therefore, it is recommended to use the combinations of signature-based analysis methods with machine learning algorithms for gradient boosting of decision trees to detect intrusions into corporate networks.

References

[1] C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," IEEE Access, vol. 5, pp. 21954–21961, 2017.

[2] X. Yuan, C. Li, and X. Li, "DeepDefense: identifying DDoS attack via deep learning," IEEE Int. Conf. on Smart Computing, pp. 1–8, 2017.

[3] R. C. Aygun and A. G. Yavuz, "Network anomaly detection with stochastically improved autoencoder based models," 4th Int. Conf. on Cyber Security and Cloud Computing, pp. 193–198, 2017.

[4] N. Van, T. Thinh, and L. Sach, "An anomaly-based network intrusion detection system using deep learning," Int. Conf. on System Science and Engineering, pp. 210–214, 2017.

[5] S. Baek, D. Kwon, J. Kim, and S. C. Suh, "Unsupervised labeling for supervised anomaly detection in enterprise and cloud networks," 4th Int. Conf. on Cyber Security and Cloud Computing, pp. 205–210, 2017.

[6] V. L. L. Thing, "IEEE 802.11 network anomaly detection and attack classification: a deep learning approach," Wireless Communications and Networking Conf., 2017.

[7] W. Wang, M. Zhu, X. Zeng, and X. Ye, "Malware traffic classification using convolutional neural network for representation learning," Int. Conf. on Information Networking, pp. 712–717, 2017.

[8] H. N. Viet, Q. N. Van, L. L. T. Trang, and S. Nathan, "Using Deep Learning Model for Network Scanning Detection," Proc. of the 4th Int. Conf. on Frontiers of Educational Technologies, pp. 117–121, 2018.

[9] T. T. Teoh, Y. Y. Nguwi, Y. Elovici, and W. L. Ng, "Analyst intuition inspired neural network based cyber security anomaly detection," Int. journal of innovative computing information and control, vol. 14(1), pp. 379–386, 2018.

[10] G. Apruzzese, "On the effectiveness of machine and deep learning for cyber security," 10th Int. Conf. on Cyber Conflict, pp. 371–390, 2018.

[11] G. Makkar, M. Jayaraman, and S. Sharma, "Network intrusion detection in an enterprise: unsupervised analytical methodology," Data Management, Analytics and Innovation, pp. 451–463, 2019.

[12] S. Peddabachigari, A. Abraham, C. Grosan, and J. Thomas, "Modeling intrusion detection system using hybrid intelligent systems," Journal of Network Computional Application, vol 30(1), pp. 114–132, 2007.

[13] S. Mukkamala, A. H. Sung, and A. Abraham, "Intrusion detection using ensemble of soft computing paradigms," Intelligent systems design and applications, pp. 239–248, 2003.

[14] S. Mukkamala, A. H. Sung, and A. Abraham, "Intrusion detection using an ensemble of intelligent paradigms," J. Netw. Comput. Appl., vol. 28(2), pp. 167–182, 2005

[15] A. Saied, R. E. Overill, and T. Radzik, "Detection of known and unknown DDoS attacks using artificial neural networks," Neurocomputing, vol. 172, pp. 385–393, 2016.

[16] B. Agarwal and N. Mittal, "Hybrid approach for detection of anomaly network traffic using data mining techniques," Proc. Tech., vol. 6, pp. 996–1003, 2012.

[17] A. Branitskiy and I. Kotenko, "Network attack detection based on combination of neural, immune and neuro-fuzzy classifiers," 18th IEEE Int. Conf. on Computational Science and Engineering, pp. 152–159, 2015.

[18] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," 4th Int. Conf. on Information Systems Security and Privacy, 2018.

[19] Balanced accuracy score. Scikit-learn, 2018. [Online]. Available: https://scikit-learn.org/stable/modules/model_evaluation.html#balanced-accuracy-score

[20] S. Narkhede, "Understanding AUC - ROC Curve," Towards Data Science, 2018. [Online]. Available: https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5

[21] K. P. Shung, "Accuracy, Precision, Recall or F1?" Towards Data Science, 2018. [Online]. Available: https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9