Journal of Scientific and Technological Research, Vol.5 (1), 2023
https://doi.org/10.59738/jstr.v5i1.23(1-8).nuyz9622

Print ISSN: 2309-7167
e- ISSN: 2959-7943

বাংলাদেশ উন্মুক্ত বিশ্ববিদ্যালয়
BANGLADESH OPEN UNIVERSITY

Journal of Scientific and Technological Research

Feature Selection based Performance Analysis of Machine Learning Algorithms in Network Intrusion Detection

Samrat Kumar Dey*
School of Science and Technology, Bangladesh Open University, Gazipur-1705, Bangladesh.

Abstract

Information and data security is one of the most challenging tasks for the massive-scale digital revolution all over the world. The very first step to secure our data is to identify invasive conduct and intrusive behavior. However, due to the high scalability of most modern systems and the complex nature of the attacks, the traditional detection system is less reliable. To overcome this challenge, it is necessary to build intelligent and adaptive intrusion detection technologies. That is why this research developed an intrusion detection system using commonly used ML algorithms and analyzed performance from different perspectives. In our pipeline, this exploration applied both supervised and unsupervised learning algorithms. The training and test data were split in multiple ways to evaluate the performance of the models. From the experimental results, it was found that the Light Gradient Boosting Machine (LightGBM) performs better in our context in terms of both precision and recall.

Keywords: Intrusion Detection, LightGBM, Machine learning, Algorithms, KDDCup-99, Evaluation.

1. Introduction

Networked computing is the backbone of the modern digital world. Any vulnerability to networked devices and computing platforms can invade the entire network with various attacks and bring catastrophic consequences. Recently, cybercrimes have been on the rise, such as data theft, phishing, carding, viruses, financial fraud, intrusions, and many more. The most obvious one is the network attacks; being able to seize attacks promptly is highly critical and, at the same time, very challenging. There are several traditional security measures to protect computers and networks from attacks and intrusions. However, we need a system that is intelligent, adaptive, and highly scalable. An intrusion detection system (IDS) is a system that is used to monitor the computer network for malicious activities to protect the system from being compromised (Nanda et al., 2016). Most of the detection systems are rule-based and work on observing abnormal activities in a network. However, with the evolution of technologies, the patterns of attacks are so complex that the traditional security mechanism can easily be deceived. Several approaches have been suggested in the literature to improve the rate of detection of an attack and to facilitate the handling of IDS (Vigna & Kemmerer, 1998). Already, many researchers have made some good contributions to analyzing network intrusion detection systems. This section summarizes the relevant prior works related to IDS and Machine Learning (ML). This study briefly explains some ML algorithms used in our pipeline, such as neural networks, Bayesian networks, evolutionary algorithms, and support vector machine detection. Ashraf *et al*. presented machine learning methods that might be

*Corresponding author: Samrat Kumar Dey (samrat.sst@bou.ac.bd)
Journal homepage: https://jstr.bousst.edu.bd

utilized to manage network intrusions (Ashraf & Latif, 2014). Ali et al., 2015 presented the survey results, which included a list of many distinct problems and solutions that have been suggested in the literature to account for network risks, as well as a discussion of the network's future challenges. On the NSL-KDD dataset, (Patgiri et al., 2018) built an Intrusion Detection System that utilizes Random Forest and SVM methods. Dey & Rahman, 2018 provided a thorough examination of the security implications of SDN. The article covers a variety of methods and approaches for dealing with security issues, including machine learning and deep learning. Alsughayyir et al., 2019 proposed a Deep Learning (DL) method for developing a more effective and efficient Intrusion Detection System (IDS). The creation of IDS is focused on distinguishing between regular and anomalous network activity. The suggested approach outperforms all traditional methods in terms of real-time and practical applications, with a training accuracy of 99 percent and a test phase accuracy of 91.18 percent. The framework is an NSL- KDD dataset-based auto-encoder. According to (Vigna & Kemmerer, 1998), NetSTAT can figure out which network events should be watched and where they should be monitored. Omrani et al., 2017 investigated the effect of machine learning techniques (ANN, SVM) in distinguishing between normal and suspicious TCP connection traffic. However, employing ANN with SVM is a costly strategy on its own. They suggested combining two classifiers, ANN and SVM. They discovered a promising approach by using this combination method with various network connections from the NSL-KDD DARPA dataset (Xu et al., 2021). The CNN and RNN in LuNet learn to input traffic data in sync with ever-increasing granularity as both spatial and temporal characteristics of the data can be retrieved successfully. In comparison to advanced network intrusion detection methods, their experimental findings on two network traffic datasets demonstrate that they not only offer a high degree of detection capacity but also have a significantly reduced false alarm rate. Dey & Rahman, 2020 used particular machine learning techniques to establish the effects of different techniques in SDN. On the KDD99 dataset, (Dey et al., 2018) used neural networks and machine learning methods to perform classification network assaults, which resulted in a greater detection rate and reduced false alarm rate in a shorter period.

Ahmad et al., 2021 used C4.5, KNN, MLP, Linear Programming, and Support Vector Machine to create an experimental framework for comparative analysis on the KDD CUP99 dataset. Patgiri *et al.*, 2018 constructed an IDS that employs two algorithms: Random Forest and Support Vector Machine. Nanda et al., 2016 presented an IDS that uses four well-known machine-learning techniques to identify possible malicious connections and attack destinations, according to Bhati et al., 2020, who utilized the NSL-KDD dataset to evaluate the performance of SVM methods. Most IDS development and analysis are based on the NSL-KDD dataset, and most studies employ two to six machine learning methods, according to our findings. However, the data-driven approaches are more effective in the context of the new significant data era. Some machine learning (ML) techniques can derive features even without human intervention. The model can tune itself based on new forms of attacks, and that is why ML-based detection systems are more appropriate for a modern cyber defense infrastructure. In this paper, we apply ten different ML algorithms; among them, seven are supervised, and three are unsupervised. We employed the algorithms mentioned above to predict the potentially vulnerable hosts and detect malicious activities in the network effectively. The reason behind using these ten different machine learning models to develop intrusion detection systems and make a comparative analysis among them is that this analysis helps us understand which model is better than others for our specific dataset and train test dataset ratio. Most of the previous similar types of research work use one or two algorithms to make a comparison; one or two models are not enough for practical comparative analysis. So, our study provides additional information from several angles. We tested our pipeline on the datasets collected from Kaggle. The main contributions of this paper are enumerated as follows:

1. Feasibility study of ten different ML algorithms for network intrusion detection.
2. Develop an Intrusion Detection System using ten different machine learning algorithms.
3. Compare performance evaluations on a wide variety of train and test data ratios, focusing on precision and recall.

2. Materials and Methods

An intrusion detection system (IDS) is a software program that scans networks for harmful or unauthorized access. Machine learning algorithms are used to train the model for real-time surveillance of these malicious actions or assaults, such that a data packet is dropped when it is detected as malicious data or attacks. For training the model, the sequential steps are shown in Fig. 1, which includes data collection, data preprocessing, feature selection, the model for training, and performance measures. Based on previous network attack data, we applied supervised and unsupervised machine learning (ML) techniques to anticipate potential assaults. We analyze the performance of ten different machine learning algorithms for predicting the host that would be attacked: SVM, Logistic Regression, Naive-Bayes, LightGBM, XGBoost, KNN, ANN, K-means clustering, Isolation Forest, and DBSCAN methods.

2.1 DATA COLLECTION

For the implementation, we used three different datasets from Kaggle: intrusion detection, KDD Cup-99, and intrusions simulated in a military network environment. Three datasets are combined here, having 645185 data with 42 features. We have categorized all kinds of attacks as anomalies. Thus, we have created two classes: Normal and Anomaly data. Class distribution of Anomaly: 467116 (72.4%) and Normal: 178069 (38.12%) data.

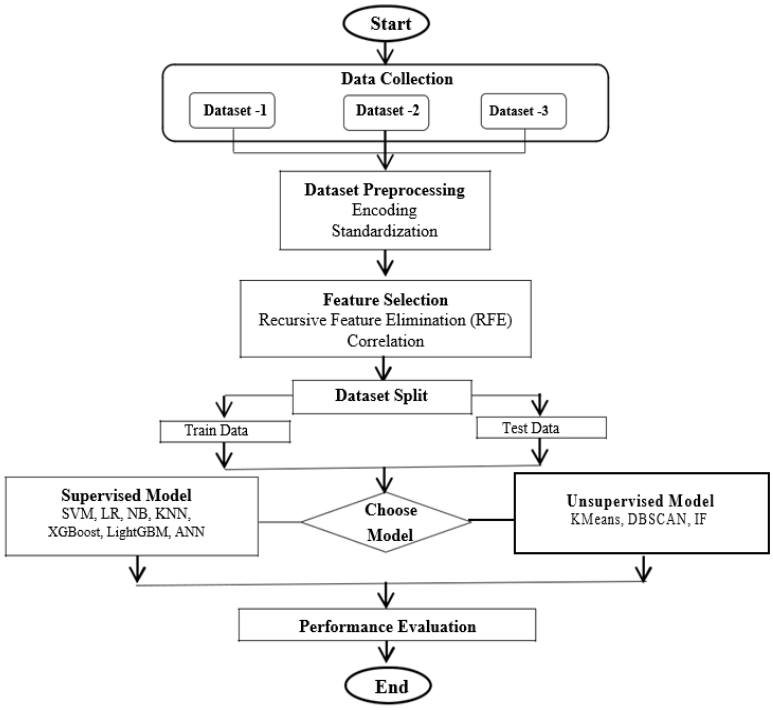

Figure 1: Flow Diagram of the Proposed Method

2.2 DATA PREPROCESSING

Data preprocessing means preparing raw data to make it appropriate for the construction and training of Machine Learning models. Our data is preprocessed in three ways.

(a) Cleaning: Our datasets contain a large amount of data. Large datasets are often noisy, redundant, and contain a variety of data kinds, making data modeling difficult. The cleaning

process discovers incomplete, erroneous, inaccurate, or irrelevant data in datasets and detects and corrects or removes corrupt or inaccurate data.

(b) Encoding: Convert categorical values to numeric values using Label Encoding. Before we can fit and evaluate a model, categorical data must be encoded into integers. It is an essential preprocessing step for the structured dataset in supervised learning.

(c) Standardization: Data normalization is the process of rescaling one or more attributes to have a mean value of 0 and a standard deviation of 1.

For supervised learning algorithms, we have normalized the dataset using the Standard Scaling method. For unsupervised learning algorithms, we have normalized the dataset using the Min-Max Scaling method.

2.3 SELECTION OF FEATURE

To select the feature, we use recursive feature elimination and random forest method for the supervised algorithm and correlation feature selection method for the unsupervised machine learning algorithm.

(a) Recursive Feature Elimination (RFE): Recursive feature removal is a wrapper method for removing predictors in order to obtain the best combination for model performance. Random forest is an intrinsic feature selection method. This built-in model includes predictors that help maximize accuracy. It prioritizes the most significant properties, discards the ones that aren't, and refits the model. Fig. 2 shows the graph representation of feature importance using the recursive feature elimination method of our experiment.

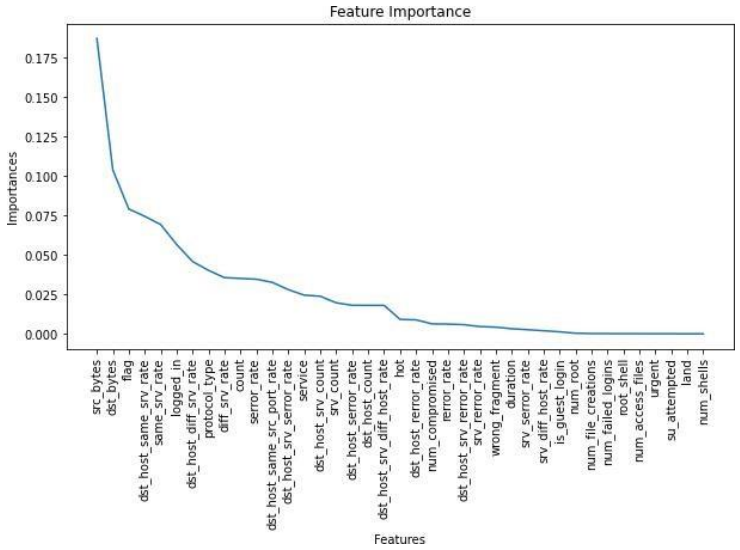

Figure 2: Feature importance for recursive feature elimination method

(b) Correlation: Correlation is a statistical term that describes how close two variables are to having a linear connection with each other. Fig. 3 shows the graph representation of our experiment's Correlation method.

2.4 MACHINE LEARNING MODEL SELECTION

We have applied ten machine learning techniques, seven of which are supervised algorithms and others unsupervised algorithms. Then, we used three-fold cross-validation and computed the individual output for these six supervised algorithms. For ANN, we used the ReLu function for the input layer and hidden layer and the sigmoid function for the output layer. We computed the accuracy using a classification report.

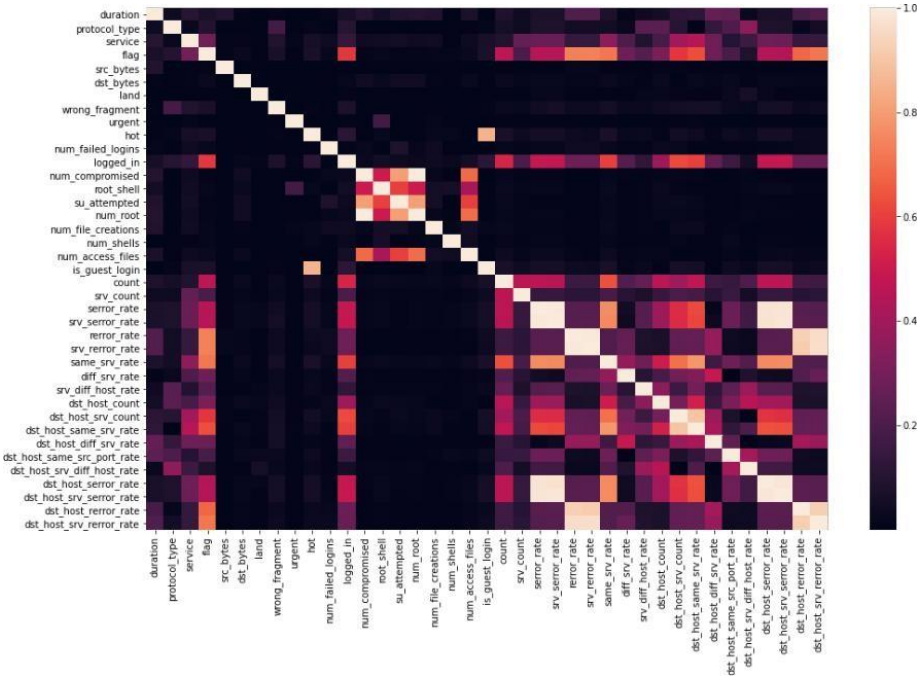

Figure 3: Correlation feature selection

From our literature review section, we find out that many intrusion detection systems were developed based on the SVM algorithm *(*Omrani et al., 2017*)*. Compared to other latest algorithms like neural networks, with a thousand samples, SVM has two key advantages: it is faster and performs better.

3. Results and Discussion

Table 1 shows precision and recall for different train test data ratios for different machine learning models. Detection is performed after training the model using SVM, Logistic Regression, Naive Bayes, KNN, XGBoost, Light- GBM, ANN, K-Means clustering, and DBSCAN on the given dataset for various attacks that are represented by confusion matrices. The information provided in a confusion matrix is used to calculate precision and recall. We did a large-scale experiment. We have used three different train test data ratios for all the algorithms we chose except K-means clustering and DBSCAN. The train test spilled is a technique for evaluating the performance of a machine learning algorithm. The objective behind dividing the dataset into the train test dataset is to estimate the performance of the machine learning model on new data, meaning data that are not used to train the model. By varying the size of train test data, we tried to figure out which model gives a more stable accuracy result for each case and how much variation of result happens in different models for different train test data ratios. This train test split also helps to understand the computational efficiency of the machine learning model. There is no optimal split percentage. We chose three different training tests spilled:

(a) Train: 0.6, Test: 0.4
(b) Train: 0.7, Test: 0.3
(c) Train: 0.8, Test: 0.2

From Fig. 4, we can say that LightGBM gives better performance, and IF gives the worst performance among those seven different algorithms for all different train and test data ratios. Table 2 shows precision and recall for different unsupervised machine learning models. Fig. 5 shows precision and recall K-means clustering and DBSCAN. From Table 2 and Fig. 5, we can say that K-

means gives better performance than DBSCAN. From these graphical and tabular representation results of precision and recall, we can say that LightGBM gives the best performance for all the different train test data ratios. On the other hand, IF gives the worst performance for all the different train test data ratios. Another important observation is that the precision and recall results vary more for IF for the different train test data than other machine learning models.

Table 1: Detection accuracy of different machine learning algorithms for the datasets in different training/testing datasets spilled scenarios

Algorithms	**Train 60%, Test 40%**		**Train 70%, Test 30%**		**Train 80%, Test 20%**	
	Precision	**Recall**	**Precision**	**Recall**	**Precision**	**Recall**
SVM	99.80	99.79	99.82	99.81	99.81	99.80
Logistic Regression	98.94	97.75	99.80	97.55	99.03	97.73
Naïve Bayes	96.91	96.89	96.95	96.81	96.95	96.93
KNN	99.86	99.84	99.85	99.86	99.86	99.85
XGBoost	99.91	99.80	99.90	99.81	99.89	99.79
LightGBM	99.98	99.96	99.98	99.96	99.98	99.96
ANN	99.00	99.00	99.00	99.00	99.42	99.36
Isolation Forest (IF)	86.02	93.59	87.44	93.03	90.30	92.75

Table 2: Detection accuracy of two different unsupervised machine learning algorithms

Algorithms	**Precision**	**Recall**
K-means	93.0	91.0
DBSCAN	73.5	49.5

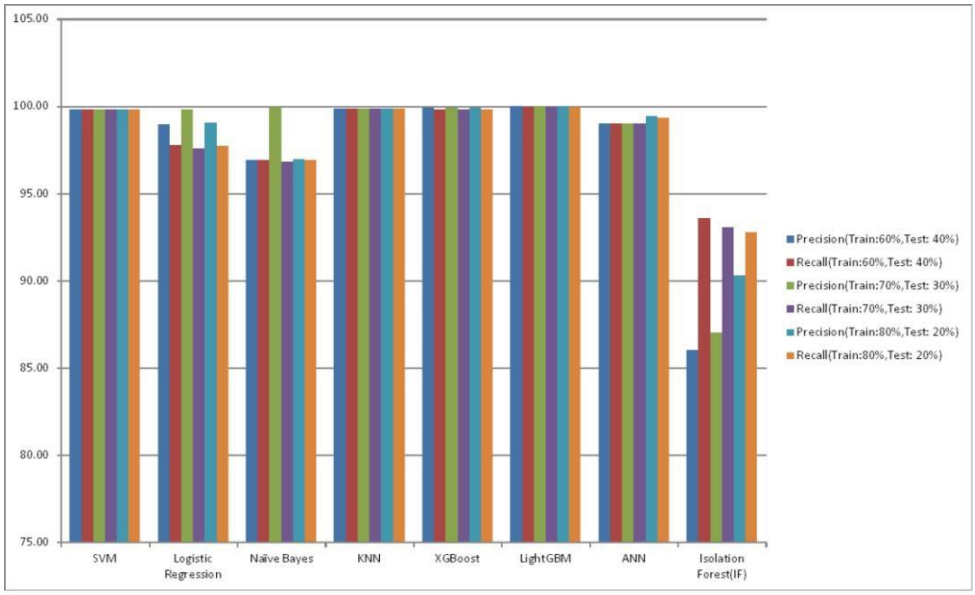

Figure 4: Prediction accuracy of different ml algorithms for the datasets in three different training/testing split ratio

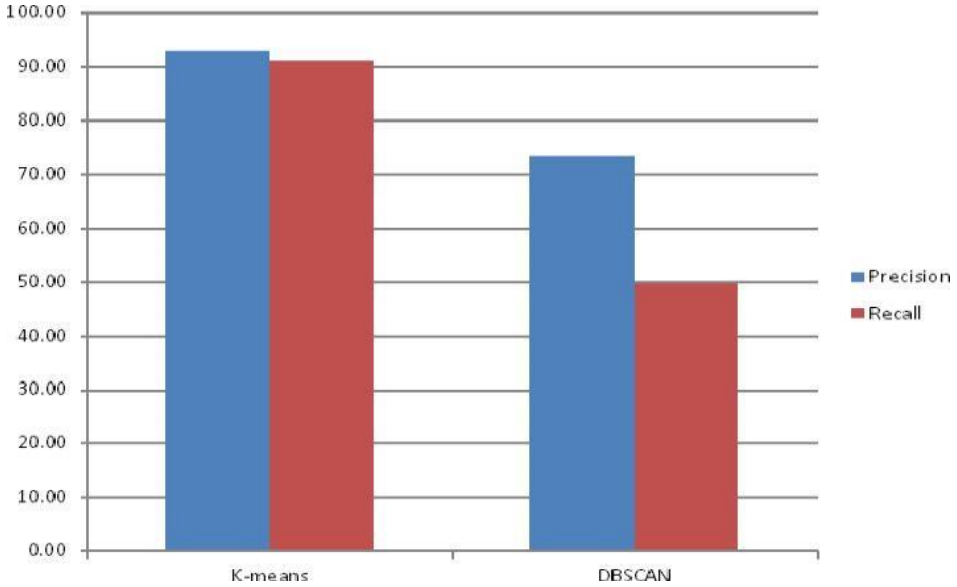

Figure 5: Prediction accuracy using different unsupervised machine learning model

4. Conclusion

In this paper, SVM, Logistic Regression, Naive Bayes, KNN, XGBoost, LightGBM, ANN, IF, K-means, and DBSCAN algorithms have all been used to do a complete examination of intrusion detection. The different feature selection and train test data ratios have been applied to the KDD-99 dataset, intrusions simulated in a military network environment, and another intrusion detection dataset. Python and Google Colab were utilized for simulations. The results were examined with the help of a classification report. Model-building time and accuracy for every technique have been observed. Individual attack class detection accuracy has been identified and analyzed. From all the perspectives, we found that LightGBM shows better performance. Our experiment, evaluation, and observation will help the researchers who will work with the Intrusion Detection System (IDS) based on machine learning in the future to understand better how and why the accuracy of the machine learning model varies.

Source of Funding
The study has not received any funds from any organization.

Competing Interests Statement
The authors have declared no competing interests.

Consent for Publication
The authors declare that they consented to the publication of this study.

References

Ahmad, Z., Shahid Khan, A., Wai Shiang, C., Abdullah, J., & Ahmad, F. (2021). Network intrusion detection system: A systematic study of machine learning and deep learning approaches. *Transactions on Emerging Telecommunications Technologies*, *32*(1), e4150. https://doi.org/10.1002/ett.4150

Ali, S. T., Sivaraman, V., Radford, A., & Jha, S. (2015). A Survey of Securing Networks Using Software Defined Networking. *IEEE Transactions on Reliability*, *64*(3), 1086–1097. https://doi.org/10.1109/TR.2015.2421391

Alsughayyir, B., Qamar, A. M., & Khan, R. (2019). Developing a Network Attack Detection System Using Deep Learning. *2019 International Conference on Computer and Information Sciences (ICCIS)*, 1–5. https://doi.org/10.1109/ICCISci.2019.8716389

Ashraf, J., & Latif, S. (2014). Handling intrusion and DDoS attacks in Software Defined Networks using machine learning techniques. *2014 National Software Engineering Conference*, 55–60. https://doi.org/10.1109/NSEC.2014.6998241

Bhati, B. S., Rai, C. S., Balamurugan, B., & Al-Turjman, F. (2020). An intrusion detection scheme based on the ensemble of discriminant classifiers. *Computers & Electrical Engineering*, *86*, 106742. https://doi.org/10.1016/j.compeleceng.2020.106742

Dey, S. K., & Rahman, M. M. (2020). Effects of Machine Learning Approach in Flow-Based Anomaly Detection on Software-Defined Networking. *Symmetry*, *12*(1), Article 1. https://doi.org/10.3390/sym12010007

Dey, S. K., & Rahman, Md. M. (2018). Flow Based Anomaly Detection in Software Defined Networking: A Deep Learning Approach With Feature Selection Method. *2018 4th International Conference on Electrical Engineering and Information & Communication Technology (iCEEiCT)*, 630–635. https://doi.org/10.1109/CEEICT.2018.8628069

Dey, S. K., Rahman, Md. M., & Uddin, Md. R. (2018). Detection of Flow Based Anomaly in OpenFlow Controller: Machine Learning Approach in Software Defined Networking. *2018 4th International Conference on Electrical Engineering and Information & Communication Technology (iCEEiCT)*, 416–421. https://doi.org/10.1109/CEEICT.2018.8628105

Nanda, S., Zafari, F., DeCusatis, C., Wedaa, E., & Yang, B. (2016). Predicting network attack patterns in SDN using machine learning approach. *2016 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, 167–172. https://doi.org/10.1109/NFV-SDN.2016.7919493

Omrani, T., Dallali, A., Rhaimi, B. C., & Fattahi, J. (2017). Fusion of ANN and SVM classifiers for network attack detection. *2017 18th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA)*, 374–377. https://doi.org/10.1109/STA.2017.8314974

Patgiri, R., Varshney, U., Akutota, T., & Kunde, R. (2018). An Investigation on Intrusion Detection System Using Machine Learning. *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, 1684–1691. https://doi.org/10.1109/SSCI.2018.8628676

Vigna, G., & Kemmerer, R. A. (1998). NetSTAT: A network-based intrusion detection approach. *Proceedings 14th Annual Computer Security Applications Conference (Cat. No.98EX217)*, 25–34. https://doi.org/10.1109/CSAC.1998.738566

Xu, X., Hu, H., Liu, Y., Zhang, H., & Chang, D. (2021). An Adaptive IP Hopping Approach for Moving Target Defense Using a Light-Weight CNN Detector. *Security and Communication Networks*, *2021*, e8848473. https://doi.org/10.1155/2021/8848473