JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

MACHINE LEARNING SAFEGUARDS : NETWORK ATTACK DETECTION

[1]Kokkala Rachana, [2] Katroth Balakrishna Maruthiram

[1]Assistant Professor in CSE, [2]MCA Student

Department of Information Technology, Jawaharlal Nehru Technological University, India

Abstract: Network security relies on a robust Intrusion Detection System (IDS) to swiftly identify and thwart passive and aggressive attacks. Leveraging machine learning, particularly on the NSL-KDD datasets, enhances the efficiency of anomaly detection. By employing a hybrid feature extraction method, combining various mathematical techniques, our model optimally processes network traffic data. This involves selecting pertinent features for training binary classification models, enabling accurate prediction of normal or attack types. The evaluation of our model's performance includes assessing overall accuracy and error rates, reinforcing the role of network traffic analysis in exposing invasions and preventing attacks. The project preprocesses the NSL-KDD dataset to extract relevant features and trains ML models for binary classification of normal behavior and attack patterns. Various ML algorithms are evaluated to determine the most effective intrusion detection approach. The ensemble methods which are Voting Classifier(RF + AdaBoost) and Stacking Classifier(LGBM + MLP + RF + XGB) are implemented, resulting in a noteworthy 99% accuracy in prediction.

1. INTRODUCTION

The internet's widespread use has opened new opportunities for businesses but also introduced significant security risks, especially from unauthorized access. Intrusion Detection Systems (IDS) are vital in network security, detecting and preventing potential attacks. IDS can use signature detection to identify known threats or anomaly detection to recognize unusual activities that might signal new attacks. Machine learning (ML) has become essential in enhancing IDS effectiveness. Various ML models offer different methods to improve intrusion detection. Supervised learning classifies threats using labeled datasets, unsupervised learning identifies patterns through clustering, and reinforcement learning optimizes detection using trial and error. Our study focuses on applying a hybrid feature selection method to improve IDS performance, utilizing the NSL-KDD dataset for training and testing. Effective data preprocessing is crucial to maximizing the accuracy and efficiency of the ML models employed in this detection process..

2. LITERATURE SURVEY

i)On efficiency enhancement of the correlation-based feature selection for intrusion detection systems:

The dramatic increase in the network traffic data has become a major concern in security systems. Intrusion detection systems (TDSs), as common widely used security systems for communication networks, are not an exception. An IDS monitors the network traffic to detect attacks through classifying the network traffic data into normal and abnormal classes. Due to the high dimensionality of the network traffic data, it is not always feasible for an IDS to detect intrusions quickly and accurately. Feature selection emerges as a necessary step in designing an IDS to overcome its shortcoming and enhance its performance through the reduction of its complexity and acceleration of the detection process. To this end, in this paper, we address the problem of dimensionality reduction by proposing an efficient feature selection algorithm that considers the correlation between a subset of features and the behavior class label.

ii)Machine Learning and Deep Learning Methods for Cybersecurity:

With the development of the Internet, cyber-attacks are changing rapidly and the cyber security situation is not optimistic. This survey report describes key literature surveys on machine learning (ML) and deep learning (DL) methods for network analysis of intrusion detection and provides a brief tutorial description of each ML/DL method.Papers representing each method were indexed, read, and summarized based on their temporal or thermal correlations. Because data are so important in ML/DL methods, we describe some of the

commonly used network datasets used in ML/DL, discuss the challenges of using ML/DL for cybersecurity and provide suggestions for research directions.

iii)Improving Classification Accuracy Using Combined Filter+Wrapper Feature Selection Technique:

Large datasets such as medical datasets contain a lot of redundant or irrelevant attributes which will degrade the performance of the classifier. Feature selection methods help to improve the classification accuracy by providing relevant feature subset that best describes the dataset. The generated feature subset will be tendentious to the feature selection approach and the classification accuracy will rely on the relevancy of the features selected. Hence the process of deciding appropriate feature selection algorithm has a crucial role in providing relevant features. Traditional techniques for feature selection are filter methods and wrapper methods.

iv)Review on Network Intrusion Detection Techniques using Machine Learning:

The security given to a network from unapproved access and dangers is broadly called as network security.

It is the obligation of network managers to embrace preventive measures to shield their networks from potential security dangers. Computer networks that are associated with consistent data transactions inside the administration or business require security. The exponential development in the information that streams inside network, the quantity of individuals active on network, makes it essential to have a productive system that disallows outsiders to attack and access secret information. Consistently developing digital attacks should be checked to defend classified information.

3. METHODOLOGY

i) Proposed Work:

Hybrid feature extraction method is proposed to identify the most pertinent features from the pre-processed NSL-KDD dataset [3, 5]. This dataset, curated to retain crucial information for machine learning techniques in network traffic analysis and anomaly detection, serves as the foundation of our approach. Utilizing the selected features, Support Vector Machine (SVM), and Naive Bayesian [11] models are trained. These models undergo rigorous testing to assess their ability to predict network attacks. Evaluation metrics, including precision and error rates, guide the selection of the most effective model for further analysis and deployment in fortifying network security. And also, ensemble methods including Voting Classifier (combining Random Forest and AdaBoost) and Stacking Classifier (integrating Random Forest, Multi-layer Perceptron with LightGBM and XGBoost) are employed to enhance prediction accuracy. This ensemble approach achieved an impressive 99% accuracy, demonstrating its effectiveness in producing robust predictions. Additionally, we developed a user-friendly front end using the Flask framework to facilitate user testing, complete with user authentication features to ensure secure access to the system. This is not only improves the predictive capabilities of our system but also enhances user experience and security during testing and deployment.

ii) System Architecture:

As given in figure 1, the network data is collected to be supplied as input to the model. This data is collected from the existing network dataset which is NSL-KDD[5] dataset. The features in the dataset are subjected to pre- processing as proposed in [6] .We employ hybrid feature extraction methods to choose the most pertinent feature subset from the dataset after pre-processing it. The SVM and Naive Bayesian [11] machine learning models are trained using the retrieved features to learn the parameters. Once the training phase is over, the models are subjected to test the dataset. The test phase is where it predicts the presence of attack in the network. We have used a binary confusion matrix on both models to display the result. It outputs the number of records in the test dataset classified correctly and incorrectly as normal and attack sets. Then, we test each model's accuracy and error rate and find which suits best for binary classification.

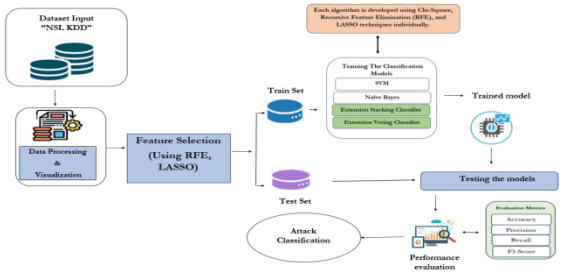

Fig 1 Proposed architecture

iii) Dataset collection: NSL-KDD [3, 5] is a public dataset, which has been developed from the earlier KDD cup99 dataset (Tavallaee et al., 2009). A statistical analysis performed on the cup99 dataset raised important issues that heavily influence the intrusion detection accuracy and results in a misleading evaluation of AIDS (Tavallaee et al., 2009). The main problem in the KDD data set is the huge amount of duplicate packets. Tavallaee et al. analyzed KDD training and test sets and revealed that approximately 78% and 75% of the network packets are duplicated in both the training and testing dataset (Tavallaee et al., 2009). This huge quantity of duplicate instances in the training set would influence machine-learning methods to be biased towards

normal instances and thus prevent them from learning irregular instances that are typically more damaging to the computer system. Tavallaee et al. built the NSL-KDD dataset in 2009 from the KDD Cup'99 dataset to resolve the matters stated above by eliminating duplicated records (Tavallaee et al., 2009). The NSL-KDD train dataset consists of 125,973 records and the test dataset contains 22,544 records. The size of the NSL-KDD dataset is sufficient to make it practical to use the whole NSL-KDD dataset without the necessity to sample randomly. This has produced consistent and comparable results from various research works. The NSL_KDD dataset comprises 22 training intrusion attacks and 41 attributes (i.e., features). In this dataset, 21 attributes refer to the connection itself and 19 attributes describe the nature of connections within the same host (Tavallaee et al., 2009).

	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot	...	dst_host_srv_count	dst_host_same_srv_rate	dst_host_d
0	0	tcp	ftp_data	SF	491	0	0	0	0	0	...	25	0.17	
1	0	udp	other	SF	146	0	0	0	0	0	...	1	0.00	
2	0	tcp	private	S0	0	0	0	0	0	0	...	26	0.10	
3	0	tcp	http	SF	232	8153	0	0	0	0	...	255	1.00	
4	0	tcp	http	SF	199	420	0	0	0	0	...	255	1.00	

Fig 2 NSL KDD dataset

iv) Data Processing:

Data processing transforms raw data into valuable information through collection, organization, cleaning, and analysis, converting it into readable formats like graphs or documents. It can be done manually, mechanically, or electronically, with the goal of enhancing information value and aiding decision-making. Automated data processing, especially with software, is crucial for turning large datasets, including big data, into actionable insights for better management and strategic decisions.

v) Feature selection:

Feature selection involves identifying the most relevant, non-redundant features for building a predictive model. As datasets grow in size and complexity, reducing them is crucial to improving model performance and lowering computational costs. By eliminating irrelevant or redundant features, feature selection streamlines the input for machine learning algorithms, enhancing efficiency and accuracy.

Chi2: Chi-squared (Chi2) is a statistical test used to assess the association between categorical variables or to evaluate the goodness of fit. In feature selection, the Chi2 test measures the independence between each feature and the target variable, helping to identify the most relevant features by ranking them based on their Chi-squared values. Higher scores indicate stronger relevance, making Chi2 a useful tool for selecting features in machine learning models.

LASSO: Lasso (Least Absolute Shrinkage and Selection Operator) is a regression analysis technique that performs both feature selection and regularization. By adding a penalty equal to the absolute value of the coefficients, Lasso encourages the shrinkage of less important feature coefficients to zero, effectively selecting a smaller set of relevant features. This helps improve model interpretability and prevents overfitting, especially in cases with high-dimensional data.

RFE: Recursive Feature Elimination (RFE) is a feature selection technique that iteratively removes the least important features from a model. It works by fitting a model, ranking features based on their importance, and then eliminating the least significant ones. This process is repeated until the desired number of features is reached, helping to improve model performance by selecting the most relevant features.

vi) Algorithms:

Support Vector Machine (SVM): Support Vector Machine (SVM) is a supervised machine learning algorithm used for classification and regression tasks. It works by finding the optimal hyperplane that best separates data into different classes, maximizing the margin between the closest data points (support vectors) of each class. SVM is particularly effective in high-dimensional spaces and for cases where the classes are not linearly separable.

Naive Bayes : Naive Bayes is a probabilistic machine learning algorithm based on Bayes' theorem. It assumes that features are independent given the class label, which simplifies the computation of probabilities. Naive Bayes is commonly used for classification tasks, particularly in text classification and spam filtering [11].

Voting Classifier: A Voting Classifier is an ensemble machine learning technique that combines predictions from multiple models to improve accuracy and robustness. It's primarily used for classification tasks. The main idea is to leverage the strengths of different models to make a final prediction that's more reliable than any individual model.

There are two types of Voting Classifiers:

1. **Hard Voting:** In hard voting, each model (or "voter") makes a class prediction, and the class with the most votes is chosen as the final output.

2. **Soft Voting:** In soft voting, each model predicts the probability of each class, and the final prediction is based on the average (or weighted average) of these probabilities. The class with the highest average probability is selected.

Stacking Classifier: A Stacking Classifier is an ensemble learning method that combines predictions from multiple models to improve overall accuracy. It involves two main steps:

1.Base Models: Train several different machine learning models (base learners) on the same dataset. These models can include a variety of algorithms, such as decision trees, logistic regression, and support vector machines.

2.Meta-Model: Use the predictions from the base models as inputs to a new model, known as the meta-model. This meta-model learns how to best combine the base models' predictions to make the final classification decision.

4. EXPERIMENTAL RESULTS

Precision: Precision evaluates the fraction of correctly classified instances or samples among the ones classified as positives. Thus, the formula to calculate the precision is given by:

Precision = True positives/ (True positives + False positives) = TP/(TP + FP)

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

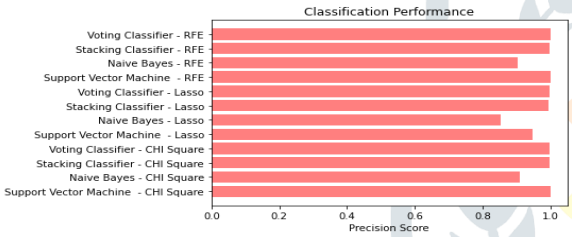

Fig 3 Precision comparison graph

Recall: Recall is a metric in machine learning that measures the ability of a model to identify all relevant instances of a particular class. It is the ratio of correctly predicted positive observations to the total actual positives, providing insights into a model's completeness in capturing instances of a given class.

$$Recall = \frac{TP}{TP + FN}$$

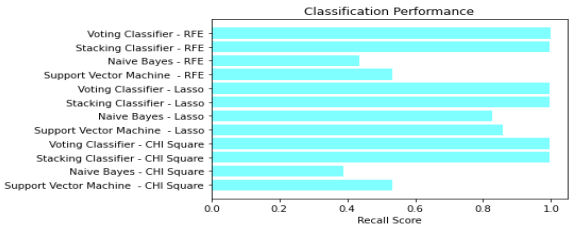

Fig 4 Recall comparison graph

Accuracy: Accuracy is the proportion of correct predictions in a classification task, measuring the overall correctness of a model's predictions.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

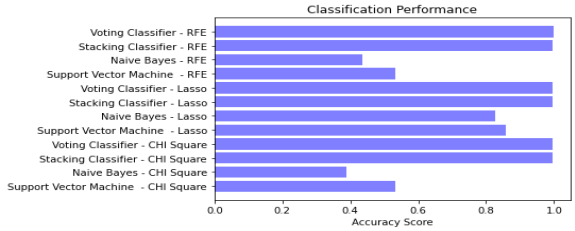

Fig 5 Accuracy graph

F1 Score: The F1 Score is the harmonic mean of precision and recall, offering a balanced measure that considers both false positives and false negatives, making it suitable for imbalanced datasets.

$$\text{F1 Score} = 2 * \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} * 100$$

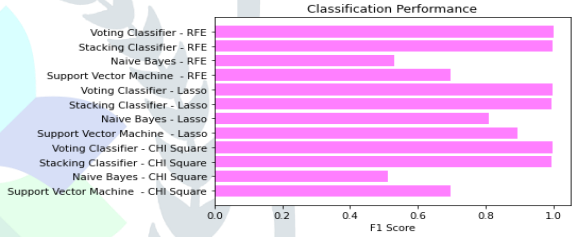

Fig 6 F1Score

ML Model	Accuracy	Precision	Recall	F1-Score
Support Vector Machine - CHI Square	0.533	0.999	0.533	0.695
Naive Bayes - CHI Square	0.390	0.909	0.390	0.512
Extension Stacking Classifier - CHI Square	0.995	0.996	0.995	0.995
Extension Voting Classifier - CHI Square	0.997	0.997	0.997	0.997
Support Vector Machine - Lasso	0.857	0.947	0.857	0.895
Naive Bayes - Lasso	0.828	0.854	0.828	0.809
Extension Stacking Classifier - Lasso	0.995	0.995	0.995	0.995
Extension Voting Classifier - Lasso	0.997	0.997	0.997	0.997
Support Vector Machine - RFE	0.533	0.999	0.533	0.695
Naive Bayes - RFE	0.435	0.904	0.435	0.531
Extension Stacking Classifier - RFE	0.996	0.997	0.996	0.997
Extension Voting Classifier - RFE	0.998	0.999	0.998	0.999

Fig 7 Performance Evaluation

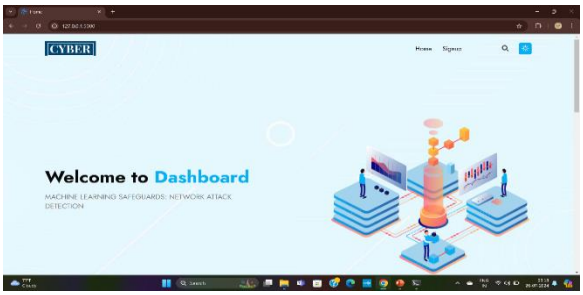

Fig 8 Home page

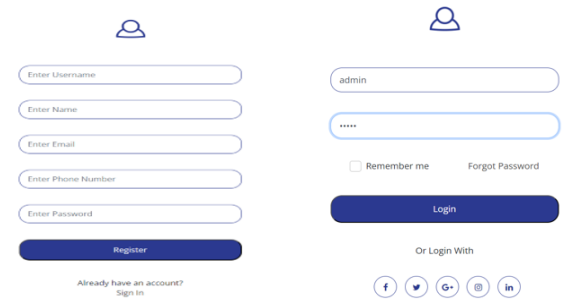

Fig 9 Sign in and Login pages

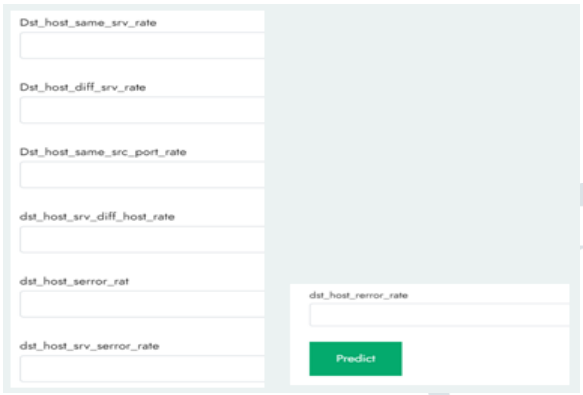

Fig 10 User input

Result: There is an Attack Detected, Attack Type is DDoS!

Fig 11 Predict result for given input

5. CONCLUSION

The project successfully implemented a hybrid feature extraction method using machine learning to detect network attacks, achieving 99% accuracy with the NSL-KDD dataset. By employing a Voting Classifier (RF + AdaBoost), the system demonstrated exceptional performance in predicting normal and attack types. The model was tested through a user-friendly interface, proving its practical effectiveness. The project highlighted the importance of robust network traffic analysis and proactive Intrusion Detection Systems to enhance network security against potential threats.

6. FUTURE SCOPE

To learn the parameters, the retrieved features are used to train the SVM and Naive Bayesian machine learning models [11, 14]. The models undergo testing on a dataset following the training phase in order to assess their machine learning performance. Whether a network attack will transpire during the testing phase is predicted. We investigate this predicted result's accuracy and error rate. For subsequent analysis based on prediction accuracy, we employ the model with the highest score.

REFERENCES

[1] F. Z. Belgrana, N. Benamrane, M. A. Hamaida, A. Mohamed Chaabani and A. Taleb-Ahmed, "Network Intrusion Detection System Using Neural Network and Condensed Nearest Neighbors with Selection of NSL-KDD Influencing Features," 2020 IEEE International Conference on Internet of Things and Intelligence System (IoTaIS), 2021, pp. 23-29, doi: 10.1109/IoTaIS50849.2021.9359689.

[2] M. B. Shahbaz, Xianbin Wang, A. Behnad and J. Samarabandu, "On efficiency enhancement of the correlation-based feature selection for intrusion detection systems," 2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON),2016,pp.1-7,doi: 10.1109/IEMCON.2016.7746286.

[3] T. Yu, Z. Liu, Y. Liu, H. Wang and N. Adilov, "A New Feature Selection Method for Intrusion Detection System Dataset – TSDR method," 2020 16th International Conference on Computational Intelligence and Security (CIS),2020,pp.362-365,doi:10.1109/CIS52066.2020.00083.

[4] Ali Hussein Shamman Al-Safi , Zaid Ibrahim Rasool Hani, , Musaddak M. Abdul Zahra,"Using A Hybrid Algorithm and Feature Selection for Network Anomaly Intrusion Detection", Journal of Mechanical Engineering Research and Developments, ISSN: 1024-1752, CODEN: JERDFO, Vol. 44, No. 4, pp. 253-262. Published Year 2021.

[5] P. S. Bhattacharjee, A. K. Md Fujail and S. A. Begum, "A Comparison of Intrusion Detection by K-Means and Fuzzy C-Means Clustering Algorithm Over the NSL-KDD Dataset," 2017 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), 2017, pp. 1-6, doi: 10.1109/ICCIC.2017.8524401.

[6] Y. J. Haranwala, S. Haidri, T. S. Tricco, V. P. da Fonseca and A. Soares, "A Dashboard Tool for Mobility Data Mining Preprocessing Tasks," 2022 23rd IEEE International Conference on Mobile Data Management (MDM), 2022, pp. 278-281, doi: 10.1109/MDM55031.2022.00059.

[7] Z. Wang et al., "Image Noise Level Estimation by Employing Chi-Square Distribution," 2021 IEEE 21st International Conference on Communication Technology (ICCT), 2021, pp. 1158-1161, doi: 10.1109/ICCT52962.2021.9657946.

[8] M. S. S. Sumi and A. Narayanan, "Improving Classification Accuracy Using Combined Filter+Wrapper Feature Selection Technique," 2019 IEEE International Conference on Electrical,

Computer and Communication Technologies (ICECCT), 2019, pp. 1-6, doi: 10.1109/ICECCT.2019.8869518.

[9] M. Zhu, X. Huang and H. Pham, "A Random-Field-Environment-Based Multidimensional Time-Dependent Resilience Modeling of Complex Systems," in IEEE Transactions on Computational Social Systems, vol. 8, no. 6, pp. 1427-1437, Dec. 2021, doi: 10.1109/TCSS.2021.3083515.

[10] Y. Kim, J. Hao, T. Mallavarapu, J. Park and M. Kang, "Hi-LASSO: High-Dimensional LASSO," in IEEE Access, vol. 7, pp. 44562-44573, 2019, doi: 10.1109/ACCESS.2019.2909071.

[11] N. S. Rahmi, N. W. S. Wardhani, M. B. Mitakda, R. S. Fauztina and I. Salsabila, "SMOTE Classification and Random Oversampling Naive Bayes in Imbalanced Data : (Case Study of Early Detection of Cervical Cancer in Indonesia)," 2022 IEEE 7th International Conference on Information Technology and Digital Applications (ICITDA), 2022, pp. 1-6, doi: 10.1109/ICITDA55840.2022.9971421.

[12] L. Zeyang, "Research on Intelligent Acceleration Algorithm for Big Data Mining in Communication Network Based on Support Vector Machine," 2021 IEEE 4th International Conference on Automation, Electronics and Electrical Engineering (AUTEEE), 2021, pp. 479-483, doi: 10.1109/AUTEEE52864.2021.9668793.

[13] G. R. Kini and C. Thrampoulidis, "Phase Transitions for One-Vs-One and One-Vs-All Linear Separability in Multiclass Gaussian Mixtures," ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2021, pp. 4020-4024, doi: 10.1109/ICASSP39728.2021.9414099

[14] J. Sewall, S. J. Pennycook, D. Jacobsen, T. Deakin and a. S. McIntosh-Smith, "Interpreting and Visualizing Performance Portability Metrics," 2020 IEEE/ACM International Workshop on Performance, Portability and Productivity in HPC (P3HPC), 2020, pp. 14-24, doi: 10.1109/P3HPC51967.2020.00007.

[15] T. Wisanwanichthan and M. Thammawichai, "A Double-Layered Hybrid Approach for Network Intrusion Detection System Using Combined Naive Bayes and SVM," in IEEE Access, vol. 9, pp. 138432-138450, 2021, doi: 10.1109/ACCESS.2021.3118573.

[16] R. Wang, H. Jiang and G. Shi, "A Multi-Layer Hybrid Intrusion Detection Method Based on Nb And SVM," 2022 IEEE 4th International Conference on Civil Aviation Safety and Information Technology (ICCASIT), 2022, pp. 1384-1388, doi: 10.1109/ICCASIT55263.2022.9986813.

[17] Z. K. Maseer, R. Yusof, N. Bahaman, S. A. Mostafa and C. F. M. Foozy, "Benchmarking of Machine Learning for Anomaly Based Intrusion Detection Systems in the CICIDS2017 Dataset," in IEEE Access, vol. 9, pp. 22351-22370, 2021, doi: 10.1109/ACCESS.2021.3056614.