Machine learning based intrusion detection system: an experimental comparison

Hidayat, Imran; Ali, Muhammad Zulfiqar; Arshad, Arshad

Published in:
Journal of Computational and Cognitive Engineering

Publication date:
2022

Document Version
Author accepted manuscript

Link to publication in ResearchOnline

Citation for published version (Harvard):
Hidayat, I, Ali, MZ & Arshad, A 2022, 'Machine learning based intrusion detection system: an experimental comparison', *Journal of Computational and Cognitive Engineering*.

Machine Learning Based Intrusion Detection System: An Experimental Comparison

Imran Hidayat [1], Muhammad Zulfiqar Ali [2] and Arshad [2,*]

1 School of Computing Edinburgh Napier University, UK.
2 James Watt School of Engineering, University of Glasgow, UK.
3 School of Computing, Engineering and Built Environment, Glasgow Caledonian University, Glasgow, UK
Emails: 40457769@live.napier.ac.uk; muhammad.ali@glasgow.ac.uk

Abstract: Recently, networks are moving towards automation and getting more and more intelligent. With the advent of big data and cloud computing technologies, lots and lots of data is being produced on the internet. Every day petabytes of data are produced from websites, social media sites, or the internet. As more and more data are produced, there is a continuous threat of network attacks also growing. An intrusion Detection System(IDS) is used to detect such types of attacks in the network. IDS inspects packet headers and data and decides whether the traffic is anomalous or normal based on the contents of the packet. In this research, ML techniques are being used for intrusion detection purposes. Feature selection is also used for efficient and optimal feature selection. The research proposes a hybrid feature selection technique composed of the Pearson Correlation Coefficient and Random Forest Model. For the Machine Learning Decision tree, AdaBoost and KNN are trained and tested on the TON_IOT Dataset. The Dataset is new and contains new and recent attack types and features. For Deep Learning (DL), Multilayer Perceptron (MLP) and Long Short-Term Memory (LSTM) are trained and tested. Evaluation is done on the basis of accuracy, precision, and recall. It is concluded from the results that the Decision tree for ML and MLP for DL provides optimal accuracy with fewer false positive and negative rates. It is also concluded from the results that the ML techniques are effective for detecting intrusion in the networks.

Keywords: MLP, LSTM, KNN, IDS, machine learning.

1. Introduction

A network intrusion detection system is used to detect unwanted or malicious traffic in the network. An intrusion detection system detects anomalies or attacks in real-time. Now a days mostly applications are moving to the cloud. Due to rapid and fast growth of network devices, security risks got increased. For that reason, the security of cloud infrastructure and network resources is the main priority in the modern world. Therefore, IDS should be accurate, error-free and efficient. Due to the advent of cloud computing, the Internet of Things (IoT) and quantum computing huge amount of data is being created every day, known as big data. This big data also helps in training the machine learning model for security purposes. Network security is a challenging field nowadays. IDS provides promising results in determining the intrusion in the network.

IDS mainly consists of two types anomaly-based and signature-based. Anomaly-based IDS used machine learning or Deep Learning techniques to detect data patterns. Signature-based IDS works on predefined attacks and rules. IDS detects threats by monitoring traffic data in computer networks and issues alert after detecting threats. IDS can be passive or active depending upon its alert system.

Today many researchers are working in the field of IDS. ML and DL techniques are used to detect network anomalies by using historical data and standard datasets. Natasha et al. [1] proposed machine learning algorithms for intrusion detection purposes. Naïve Bayes is used in the research. The results obtained from Naïve Bayes are compared with SVM.peiyang et al. [2] used SVM and a genetic algorithm for attack detection. Detection accuracy is increased by optimizing the selection parameters and weights. Gisung et al.[3] used KNN and K-Means for intrusion detection purposes. The detection accuracy got increased in this research. Shapoorifsrd et al.[4] proposed a novel technique to detect the attacks. Firstly data is segmented into smaller clusters using C 4.5 algorithm, and then multiple SVM models are created from the subset of the data. This technique reduces the time complexity of the model. .zhao et al.[5] proposes a work based on Deep Belief Networks (DBN). The dimensionality of data is reduced by using probabilistic models. The probabilistic neural network is used for the classification of data.

However, several problems exist in the IDS domain like low accuracy, high false-positive rates and relevant feature selection problem. The main contributions of this paper are given below:

1. Provide a machine learning algorithm for the purpose of detecting intrusion in the network.
2. Efficient and effective feature selection technique based on the correlation among features.
3. Comparative analysis with other machine learning techniques.
4. Increase the detection accuracy of the machine learning model.

The paper is organized as follows: section 2 presents the literature review and related work, section 3 is dedicated to methodology while section 4 and 5 presents results and conclusion respectively.

2. Related Work

According to Bashir et al. [6], organizations are facing security threats every day in the form of malware and cyber-attacks. IDS and Intrusion Prevention System (IPS) detect and prevents the network from these malwares. Raghunath et al. [7] proposes a Network Intrusion Detection System (NIDS) that detects the attacks in the network by using Machine Learning(ML) techniques and associated pattern analysis technique to detect anomaly in the network.

Gadze et al. [8] proposes IDS for Software Defined Networks (SDN) and detecting DDoS attack in the network. Deep Learning-based CNN and LSTM model is presented and evaluated. Overall, 89.63% accuracy is achieved in this research. The performance of the model is also compared with other state-of-the-art Machine Learning algorithms.Maseer et al. [9] use the CICIDS 2017 dataset for making of IDS. This is one of the new and flow-based Dataset with new attack categories. The authors utilized DL and proposed a new technique, namely AIDS. The researcher in this study evaluates the performance by using true positive and negative rates. KNN-AIDS and DT-AIDS obtain the best results in this research.

Mingzheng et al.[10] uses the NSL-KDD Dataset for IDS. Several Machine Learning algorithms are used in the study. A new framework named SHAP is proposed in the research. This algorithm combines local and global explanations for IDS. Vinaykumar et al.[11] uses the Deep Learning approach along with kdd cup 99 datasets for the making of IDS.in this research, 1,000 epochs are set for each experiment. This model is also applied to different datasets like NSL-KDD, UNSW NB 15 and CICIDS 2017 to measure the performance. In this research, high dimensional features are also learned by the model. This model also provides optimal accuracy.

Rajagopal et al.[12] uses Azure Machine Learning platform for IDS. Meta-classification approach is used for both binary and multi-classification purposes. Three datasets are used in the research which is UNBSW, CICIDS and CICDOS. 99.8% accuracy is achieved on UNSW, whereas 99% on CICIDS and 98% on CICDOS. Train and test split is 40:60 used in the research.Ahmed et al.[13] The Deep Learning technique is used for IDS. UNSW NB 15 dataset is used for training and testing purposes. In this research, CNN is used with regularized MLP instead of fully connected layers. Keras library is used for development purposes. The model is trained on GPU. Early stopping is also used to prevent the model from overfitting.

Saranyaa et al.[14] uses KDD CUP 99 dataset with several Machine Learning algorithms like LDA, Cart and Random Forest. Random Forest achieves the highest accuracy with 99.8%, LDA with 98% and CART with 98.1%. zhang et al.[15] uses Deep Learning for IDS. CNN algorithm is proposed in this research along with Google Net inception to detect network packets binary problem. Overall, 99.63% accuracy is achieved.Gao et al. [16] use NSL-KDD Dataset. A new Machine Learning model called the adaptive ensemble learning model is proposed. Multitree algorithm is proposed to increase the overall performance of the algorithm. 84.2% accuracy is achieved in this research.

Ring et al.[17] Host-based IDS is proposed. In this research, the Deep Learning model is presented. A new algorithm called ALAD is also proposed in this research. This new model detects application-level attacks in the network. Optimal accuracy is achieved through this model. This model is also compared against other state-of-the-art algorithms.Devarakonda et al. [18] use the NSL-KDD Dataset in the research. In this Deep Learning model, the auto encoder is proposed. Both Network IDS and Host-based IDS are proposed in this research.

The application of DL is widely used in the field of IDS. Deep Learning provides promising results when there is a huge amount of data which needs to be processed. In the Security field, an enormous amount of data is sometimes received from different sources and there is a need to process that data quickly or efficiently.

Ashiku et al.[19] proposes Deep Learning-based IDS to detect network attacks. They developed a flexible IDS which also detects zero-day attacks. UNSW-NB 15 dataset is used for this purpose. Overall, 95.4% accuracy is achieved in this research.Tang et al. [20] used IDS 2018 dataset for research purposes. In their study, a novel attention-based CNN-LSTM model is proposed which is based on Deep Learning. Several experiments are performed, and optimal accuracy is achieved in this research

Vladimir et al.[21] used NSL-KDD and UNSW-NB 15 datasets are used for training. The deep Reinforcement Learning approach is used. The new type of network traffic attack is detected automatically. The proposed model can process a million records of network traffic. Paper et al.[22] proposes a new Deep Learning Technique called self-taught learning (STL). This technique learns features automatically from the data and feeds them to the model. They used NSL-KDD Dataset for training. Optimal accuracy is achieved in this research.

Faker et al.[23] uses three classifiers to detect anomalies in the network. One is Deep Feed Forward Neural Network (DNN), and the other is an ensemble technique based on Random Forest and Gradient Boosting. UNSW-NB and CICIDS 2017 dataset is used. Five cross-fold validation is also used for evaluation purposes. Experimentation is done using the spark library. 99.16% accuracy is achieved on the UNSW-NB Dataset and 99.99% on the CICIDS dataset. Park et al.[24] proposes a technique called HIIDS, which is hybrid intelligent IDS. This technique learns important and most relevant features from the Dataset. LSTM and Autoencoder is used. ISCX-UNB Dataset is used for training. 97.52% accuracy is achieved in this research.

Ishtiaque et al.[25] uses kddcup 99 datasets for training. Fifteen features are used along with the MLP algorithm. 95% accuracy is achieved in this research.m95% accuracy is achieved in this research. [1] uses the RNN model, which is based on the sequence model. They used their own generated Dataset. AUC value of greater than 0.8 is achieved in this research.

Yanfang et al [26] used Machine Learning techniques for the intrusion detection system. Information gain and gain ratios are used for the selection of features.IoTID20 and NSL-KDD datasets is used in the research. Several machine learning algorithms like MLP, J48, IBK and Bagging are used in the research. 99% accuracy is achieved in the research. Tang et al [27] used Deep Learning in the research. NSL-KDD Dataset is used in the research. Stacking-based model is used in the study which is the combination of various classification models to improve the accuracy. 86.8% accuracy is achieved in the research. The results of the research were also compared with four ML algorithms. This technique improves the overall detection accuracy of the detection model. Saif et al [28] used Deep Learning to improve the accuracy of the intrusion detection model. CIC-IDS, CIC-DOS and CSE-CIC-IDS 2018 datasets are used in the research. LSTM and GRU are used in the research. Overall, 99% accuracy is achieved in the research. Antunes et al [29] proposed an ML-based model to detect zero-day attacks in the network. A Deep Learning-based model consisting of CNN, Autoencoder and LSTM is proposed. CSE-CIC-IDS 2018 dataset is used in the research. The principal component analysis technique is used select relevant features from the Dataset. Better accuracy is achieved in the research. Halbouni et al [30] proposed an ML and DL model for intrusion detection purposes. The authors reviewed several approaches used for intrusion detection purposes. Recent ML and DL algorithms are also discussed by the authors which are used for IDS purposes. Kim et al [31] proposed an ML model for intrusion detection purposes. Several algorithms are used, like AdaBoost, Random Forest, ELM, DNN, CNN and XGBoost. 95% accuracy is achieved in the research.

3. Materials

Research methodology is discussed in this section. The effectiveness of using the Machine Learning technique is also discussed in this section.

3.1. Proposed Framework

In Machine Learning, we have a block or model diagram for development purposes. The Methodology proposed for making an Intrusion Detection system is discussed below:

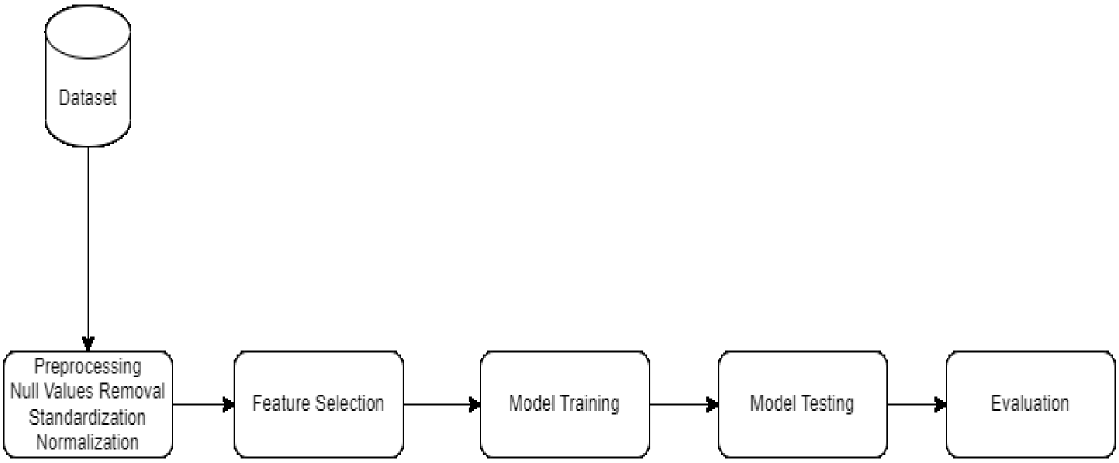


Figure 1. General framework for IDS

Figure 1 represents the proposed Methodology of the Machine Learning model. In Machine Learning, firstly, we take data from a source and then apply preprocessing techniques to that data. The data collected from different sources is not clean and sometimes may include null values, so in preprocessing, we remove null values and replace them with suitable ones. After null values removal, data needs standardization and normalization. The standardization and normalization technique puts data in the range between 0 and 1. Step by step discussion about the model is discussed below.

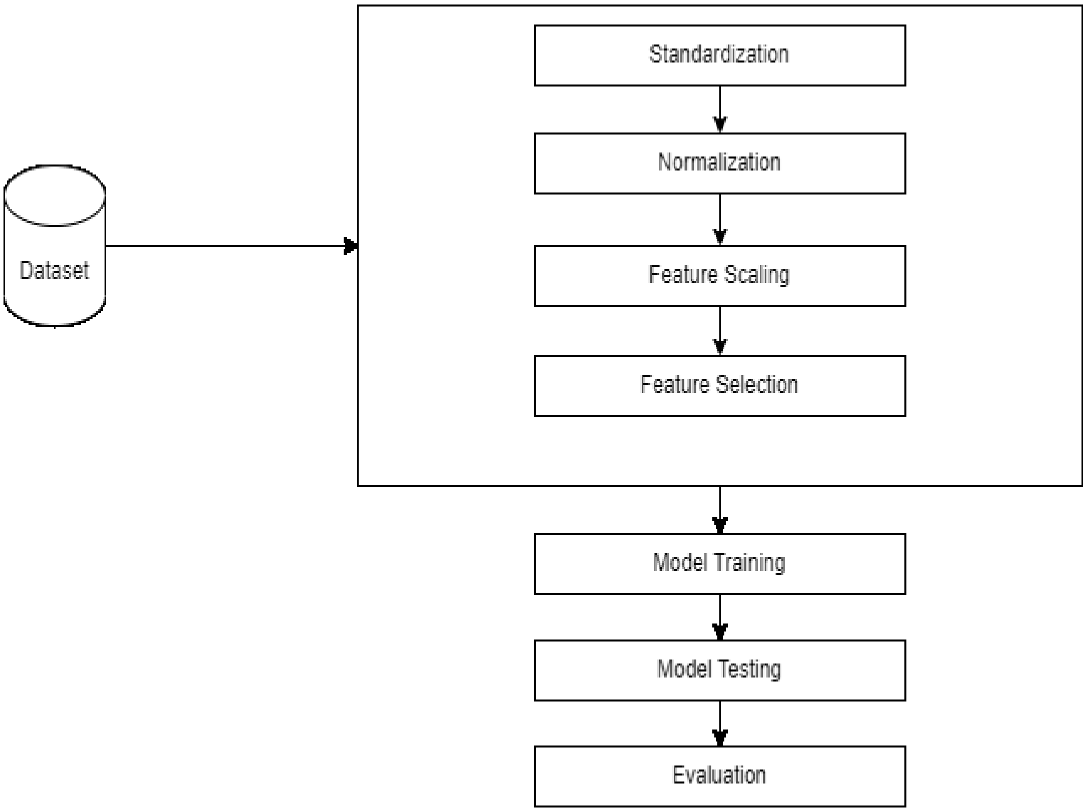


Figure 2. Proposed Methodology

3.1.1. Dataset

In research. The TON-IoT Dataset is used in the study are new Dataset and includes all the latest network attacks. TON_IOT contains features related to the IoT traffic.

Table 1. Dataset Features

Feature	Description
Ts	Timestamp
Date	Date of logging sensor data
Time	Time of logging sensor data
Fridge_temperature	Fridge sensor temperature measurement

Temp_condition	Fridge sensor temperature condition
Label	Normal or attack traffic data
Type	Normal or attack traffic type like DoS or DDoS.
Src_ip	Ip address of source
Src_port	Port number of source computer
Dst_ip	Ip address of destination
Dst_port	Port number of destination
Proto	Protocol either TCP or UDP
Duration	Connection duration
Src_bytes	Bytes send by a source computer
Dst_bytes	Bytes received by a destination computer
Conn_state	State of connection
Missed_bytes	Bytes missed by destination
Src_pkts	Packets send by a source
Src_ip_bytes	Number of ip bytes by a source
Dst_pkts	Destination packets
Dst_ip_bytes	Destination ip bytes
Dns_query	Type od dns query
http_response	Response generated by http
http_response	Response generated by http
http_status_code	Status code of http
http_version	Version of http
Weird_name	Whether a TCP is bad or not

These are all the features which are present in the TON_IoT Dataset. Not all these features are used for training purposes because not all are necessary for predicting attacks. For that purpose, a feature selection technique is used to select relevant features from the data. In feature selection techniques, relevant and most important features are selected, and the rest of the features are removed for training the model.

3.2 PreProcessing:

The data which is collected for model training and testing purpose contains outliers and null values. These values need to be removed for the efficient working of the ML model.

The data contains categorical values and numerical values. The data is collected from real-time environments and saved as a CSV to use for model-building purposes. Pre-processing involves several steps like normalization, standardization, label encoding, one-hot encoding and feature scaling. All these steps are necessary for the development of the Machine Learning model. The details about these steps are described below

3.3 Data Standardization:

Data standardization is one of the most important parts of preprocessing. Standardization rescales the data so that its standard deviation becomes one and the mean becomes 0. Standardization brings down all features of the data to the common scale. The Dataset which we used in Machine Learning mostly has many features. The value of these features lies on a different scale. Consider an example of house price prediction in which the area of the house is 200 square meters, and the number of rooms is 1, 2 or 3. If we use this data without scaling, then Machine Learning gives more importance to the features with high values. Machine Learning models will learn faster when the data is on the same scale. One solution in Machine Learning for this problem is standardization. In standardization mean value of the column is subtracted from each value and then divided by the standard deviation. In this way, data is normally distributed. In our Work, we also do standardization. The resultant data obtained by standardization is shown below.

$$\mathbf{X= x\text{-}\mu/\sigma} \qquad (1)$$

Equation 1 is the standard equation of standardization. Where μ is the mean of the data and σ is the standard deviation of the data.

3.4 Data Normalization:

Normalization is the second step in the process. The main purpose of normalization is to transform data in such a manner that the data is either dimensionless or similar distribution. Due to normalization, equal weight is given to each of the variables in the Dataset.

$$\mathbf{X[:,i]=x[:,i]\text{-}min(x[:,i])/max(x[:,i])\text{-}min(x[:,i])} \qquad (2)$$

In equation 2 min is the value of **a** which is minimum absolute, whereas max is the maximum absolute value of a.

3.5 Label Encoding:

The label encoder technique is used to convert categorical features to numerical. This technique converts each and every categorical value present in Dataset to a number.

3.6 Data Classes:

After label encoding, we need to prepare our target column. For that purpose, we assign our data label to normal or abnormal for binary classification, and for multi-class classification, all the attacks are defined.

3.7 Data Distribution:

Data distribution plays a very important role in Machine Learning model training and testing purpose. If our data is imbalanced, then the results of Machine Learning might not be good. So balanced data distribution plays a vital role in Machine Learning. If Dataset is not balanced, then we do synthetic minority oversampling technique (smote) technique to balance our dataset classes. In our case, our Dataset is balanced, so we don't need any smote technique.

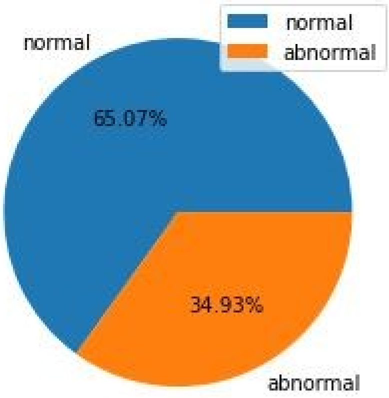


Figure 3. Data Distribution

Figure 3 represents the data distribution of the TON_IOT Dataset. It is evident from the above figure that the data is balanced in the target class. 65% of normal data is present, along with 35% of attack data. Normal distribution of data is mandatory for obtaining high accuracy because all the classes need to participate equally in model training.

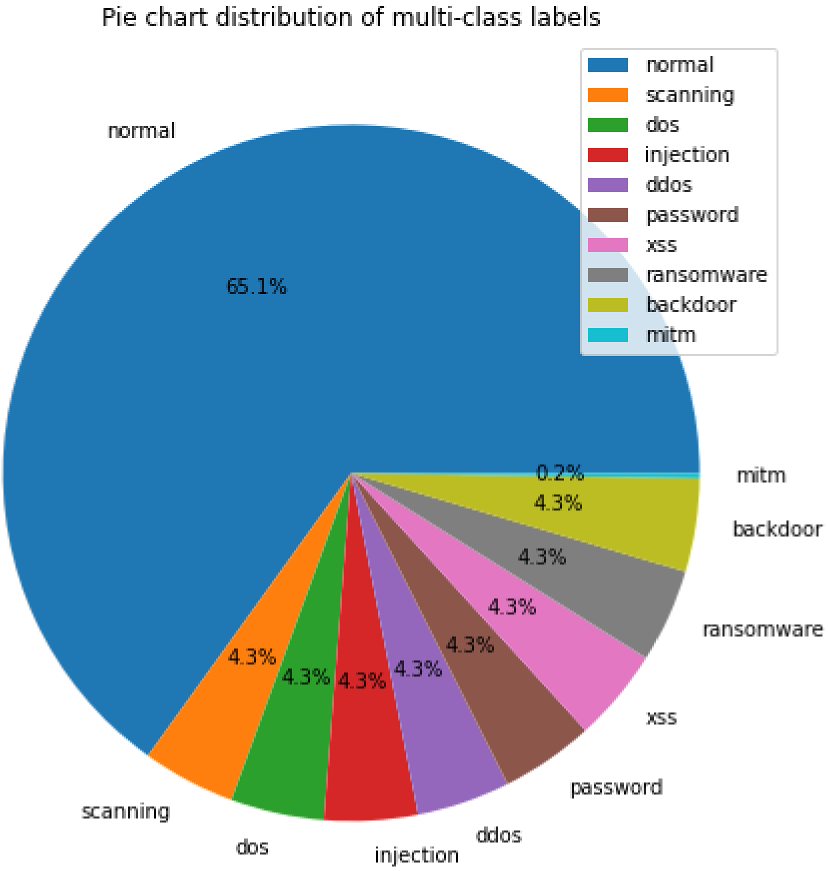


Figure 4. Overall attack distribution

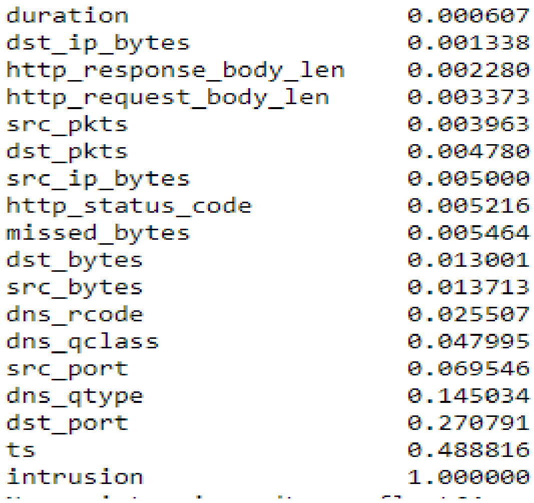
Figure 4 represents the class distribution of multi-class data. In our target class, we have ten types of attacks. Scanning, Denial of Service (DoS), Distributed Denial of Service (DDoS) and man in the middle (MITM) attack. These all are network attacks. ML models classify the traffic on the basis of these attacks. Scanning, DoS, MITM, injection, ransomware, backdoor, and XSS are all types of network attacks. The Machine Learning model is trained on data features to effectively and efficiently detect these attacks.

3.8 Feature Selection:

Feature selection is one of the most important tasks in the Machine Learning domain. Not all the features are used for model training. If so many features are present in the Dataset, then they may increase the training time and complexity of the model. Sometimes obtained data is very high dimensional, and we need to convert it to the lower dimension for efficiency and effective attack detection. So efficient feature selection technique is necessary to cope with this problem. Also, relevant feature selection is very important in Machine Learning. Sometimes, we remove some most important features and may get low accuracy. We need to cope with all these problems. Various feature selection techniques are used in Machine Learning like Recursive Feature Elimination, Chi-square or Backward feature selection techniques. These techniques are used based on datasets, dimensionality and correlation. In our case, we use the Pearson correlation coefficient technique for feature selection. This technique works based on the correlation among variables.

3.9 Pearson Correlation Coefficient:

This technique works based on correlation. This technique depicts the linear relationship among the variables in the Dataset. This technique can take a range of values between -1 and +1. A value of 0 indicates no relationship among variables, whereas -1 indicates a negative relationship and +1 indicates a positive relationship among the variables. If the relationship between two values is stronger, the correlation is close to +1.



```
duration                  0.000607
dst_ip_bytes              0.001338
http_response_body_len    0.002280
http_request_body_len     0.003373
src_pkts                  0.003963
dst_pkts                  0.004780
src_ip_bytes              0.005000
http_status_code          0.005216
missed_bytes              0.005464
dst_bytes                 0.013001
src_bytes                 0.013713
dns_rcode                 0.025507
dns_qclass                0.047995
src_port                  0.069546
dns_qtype                 0.145034
dst_port                  0.270791
ts                        0.488816
intrusion                 1.000000
```

Figure 5. Selected Features

Figure 5 represents the features selected on the basis of the correlation score in our Dataset. These are the final features which are selected for model training purposes. These features are further joined with one hot encoded variable to form complete data.

3.10 Random Forest Feature Scoring:

In the research, Random Forest feature scoring is also utilized. The features obtained from the Pearson correlation technique are given to the Random Forest for further selection. The Random Forest model selects features on the basis of their importance. Seventeen features are selected with the Pearson correlation technique. These features are further reduced to 14 after utilizing the Random Forest technique. The features which are obtained from the RandomForest model are shown below.

Table 2 Random Forest Feature Selection

Feature	selected
Duration	True

Dst_ip_bytes	True
http_response_body_len	True
http_request_body_len	True
Src_pkts	True
Dst_pkts	True
Src_ip_bytes	True
Missed_bytes	True
Src_bytes	True
Dst_bytes	True
Dns_rcode	True
Src_port	True
Dst_port	True
ts	True

Table 2 represents features that the Random Forest model selects. The features which are selected by the Random Forest model are referred to as true, and the features which are not selected are referred to as false.

3.10 Machine Learning Training:

After all the preprocessing is done, then we have the model training phase. The features which are obtained from preprocessing stage is given to the Machine Learning model for training purpose. In the training phase, the data or features which are obtained from preprocessing stage are given to the model, and the model starts training on these features. Some Machine Learning algorithms take so much time for training, while some require less time. Sometimes hyper parameter tuning is required if desired results are not obtained.

3.11 Model Evaluation:

After the testing phase, we need to evaluate our Machine Learning model on the basis of some parameters. For classification problems, confusion matrix, Accuracy and Receiver Operating Characteristic (RoC) curve are used to measure the model's performance, whereas Root Mean Square Error is used for regression problems. Accuracy and precision are considered the benchmark in binary classification problems. If our mode obtains accuracy greater than 95% with less false positive rate, then we conclude that the performance of our model is good and it is ready for a real-time production environment. Sometimes we also consider precision, recall and accuracy.

3.11.1 RoC curve:

ROC curve is also used to measure the effectiveness of the binary classification problem. ROC curve plots two parameters, true positive and false positive. ROC curve is also appropriate when the class data is balanced, whereas, for imbalanced data, precision, recall and f score is feasible.

4 Results:

In this section, the results obtained from each model are described. Models are evaluated on the basis of accuracy, detection time and testing time. The confusion matrix and ROC curve are used to evaluate model performance. Machine Learning models are tested on preprocessed Dataset, and accuracy is calculated.

4.1 Decision Tree Results:

After data preprocessing Machine Learning model is applied to the preprocessed data. Data is trained and tested in the ratio of 75:25. 75% data is used for training and 25% is used for testing the model.

The accuracy obtained from the Decision Tree model is 99.6%, which is optimal. Accuracy means how our model is accurate, and its predictions are correct. If the Machine Learning model achieves an accuracy of 90% or greater, then we conclude that its performance is considered as good.

A Decision Tree is ideal for classification problems because it mostly gives maximum accuracy if data preprocessing is done properly. The Decision Tree is composed of nodes and branches, and besides feature selection, it automatically makes feature selection on each node when splitting occurs. So Decision Tree often is used in classification-related problems.

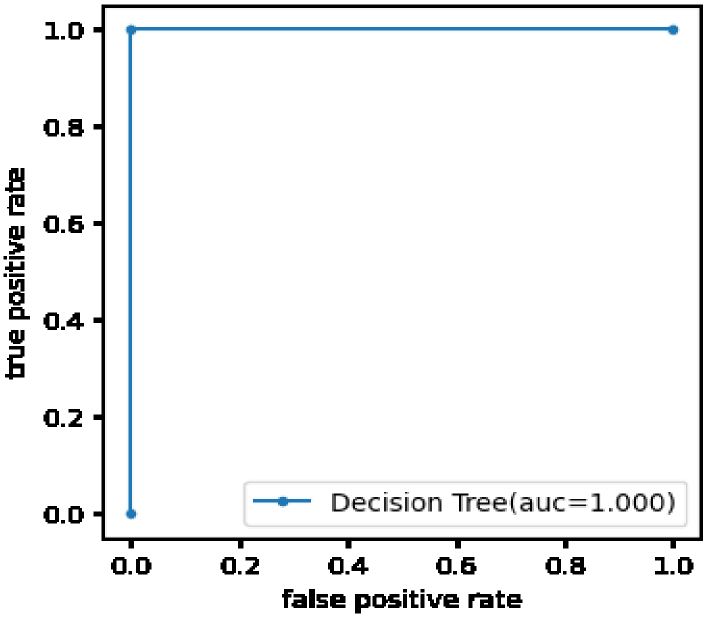


Figure 6. ROC Curve Decision Tree

Figure 6 is the ROC curve obtained from Decision Tree results. ROC curve illustrates the capability of a binary classifier. A higher AUC value tells us that the model performance is better. The ROC curve is plotted against the model's true and false-positive rates. From the classification and ROC curve it is concluded that the performance and accuracy of Decision Tree is optimal and accurate. So, Decision Tree can be used for intrusion detection system in real network environments.

4.3 KNN Results:

K-Nearesrt Neighbor (KNN) algorithm is also used in our research. The results obtained from this model is described below.

The accuracy obtained from KNN algorithm is 99%. Although the accuracy obtained from KNN model is good but it takes so much time to train or test the model making it inappropriate for deployment purpose. When dealing with IDS, we also consider the time taken by the model to train or test the algorithm.

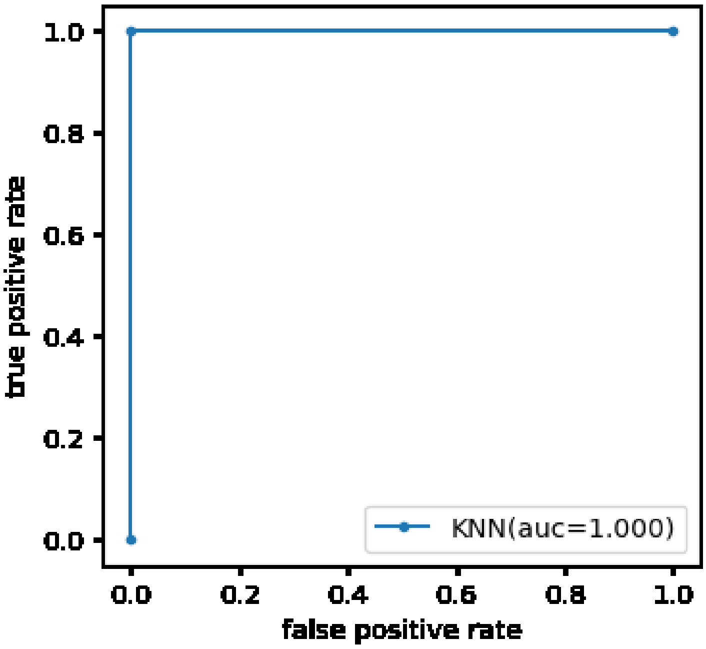


Figure 7. ROC Curve KNN

4.4 AdaBoost Accuracy:

The accuracy achieved by the AdaBoost algorithm is 99.8%. The results obtained from this model are shown below. The accuracy obtained from the AdaBoost algorithm is 99.8%. Although the accuracy obtained from the AdaBoost model is good, it takes so much time to train or test the model making it inappropriate for deployment purposes.

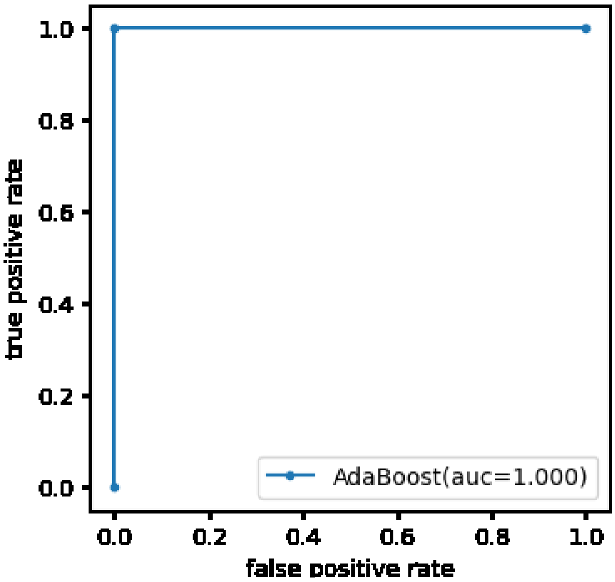


Figure 8. ROC curve AdaBoost

The ROC area is 1.00 in the case of AdaBoost. This shows that the model performs well in terms of positive predictions. Auc area is not relevant to accuracy it only refers to the positive predictions of the model.

ROC and AUC curves determine the model's efficiency in case of true positive and false-positive predictions. The area between the true positive and false positive rate is being determined by the ROC curve in the case of binary classification problems; however, in the case of multi-class classification problems, other parameters are considered.

Table 4 Accuracy

Algorithm	Accuracy	Precision	Recall	F1 Score
Decision Tree	99.6%	99%	98%	99.8%
KNN	99%	99.2%	99.4%	99%
AdaBoost	99.8%	99.8%	99.2%	99.9%

From table 2, it is evident that the accuracy of the Machine Learning models is good, but the testing and training time of the Decision Tree is less than the other algorithms making it more appropriate for real-time detection and deployment.

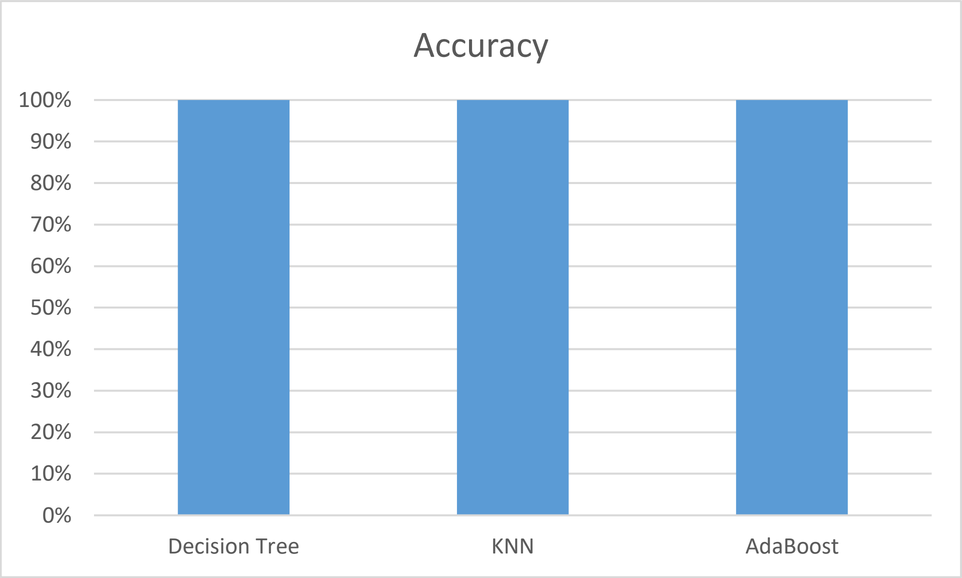


Figure 9. Comparison Graph

4.5 Deep Learning Results:

The results of the Deep Learning model are evaluated on the basis of the ROC Curve and model loss and accuracy curves. The results obtained from Deep Learning models on TON_IOT Dataset are discussed below.

4.5.1 MLP Results:

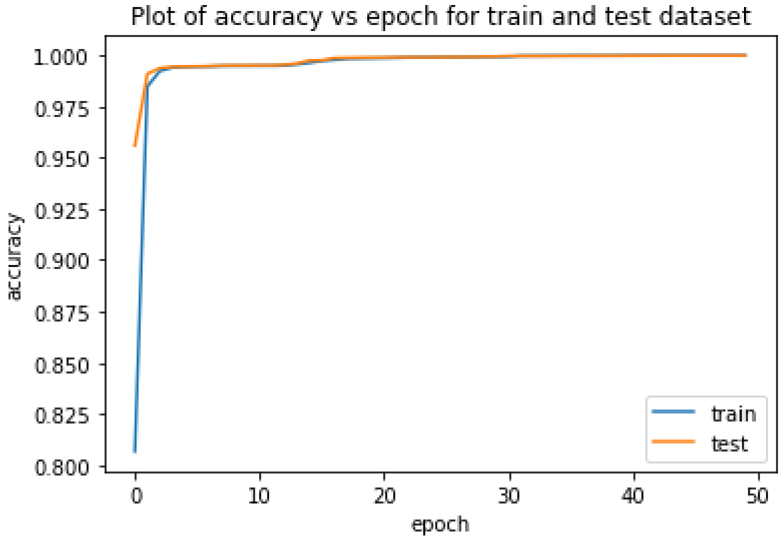


Figure 10. MLP accuracy curve

Figure 10 represents the accuracy curve for Multi-Layer Perceptron (MLP). This curve shows us that with the increase in the number of epochs, the accuracy of the model increases. At the start, the accuracy becomes low, but with the increase in epoch, accuracy increases.

Accuracy to epoch curve is the most important evaluation parameter for Deep Learning algorithms. With the increase in the number of epochs, the accuracy tends to be increased.

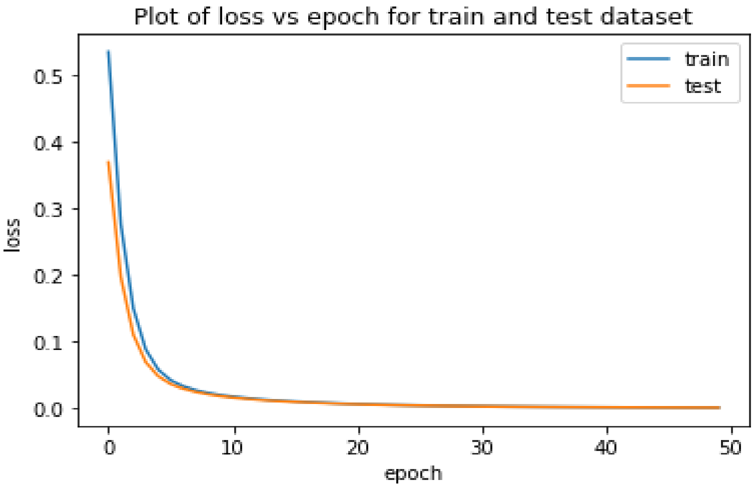


Figure 11. MLP loss curve

Figure 11 is the loss curve loss of the MLP model. It is evident from the figure that the loss of the model tends to be low with the increase in the number of epochs. So the number of epochs plays a significant role in determining the accuracy and reducing the loss of the model.

4.5.2 LSTM Results:

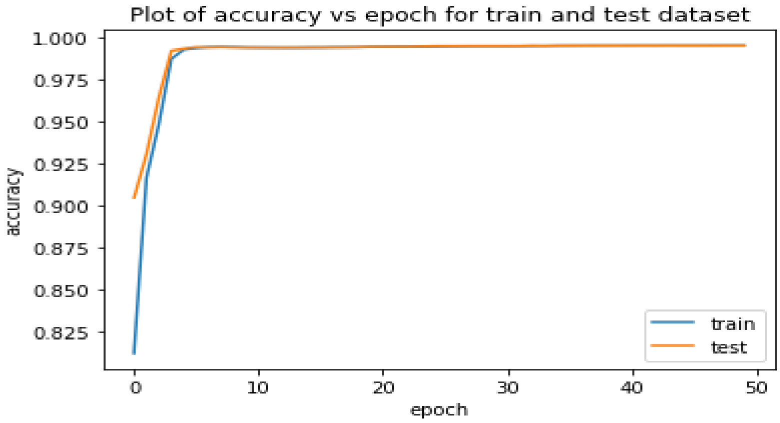


Figure 12 LSTM accuracy curve

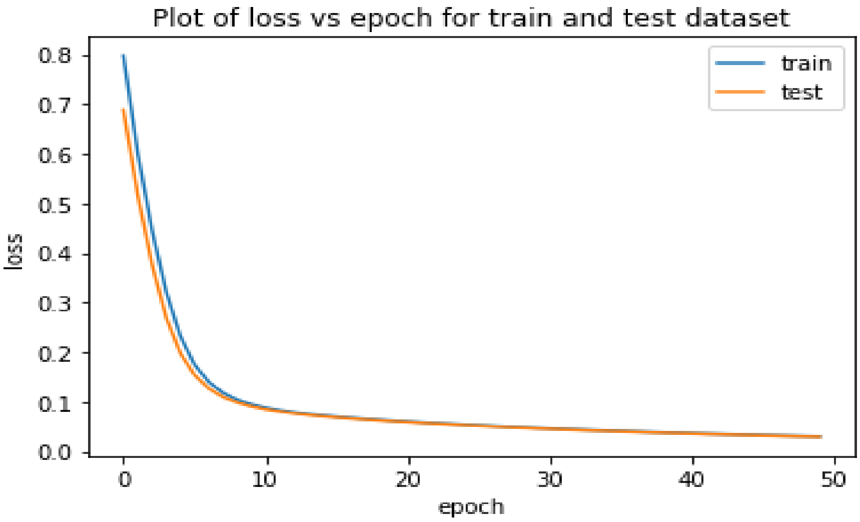


Figure 13. LSTM loss curve

Table 4. Deep Learning Accuracy

Algorithm	**Accuracy**	**Precision**	**Recall**	**F1 score**
MLP	99.2%	99%	99.4%	98%
LSTM	99%	99%	99.4%	99%

It is evident from the comparison table that the accuracy, precision and recall of the MLP model is better than the other algorithm in determining the attack in the network. The accuracy, precision, recall and f1 score of the MLP model provide optimal results as compared to the LSTM model. However, LSTM also provides good results. LSTM usually works well when sequential or temporal data, whereas MLP works well with numerical data.

5. Conclusions

In this research, Machine Learning techniques are used for the Detection of Intrusion in computer networks. TON_IOT Dataset is used in this research for intrusion detection systems. Pearson Correlation Coefficient feature selection technique is used for efficient feature selection technique. Several Machine Learning algorithms are applied to data like Decision Tree, AdaBoost, and KKN. These algorithms are evaluated on the basis of Accuracy, Precision, recall, and ROC curve. The accuracy achieved by Decision Tree on the TON_IOT Dataset is nearly 99.6% followed by AdaBoost which is also near 99.8% KNN achieves an accuracy of 99. From this research, we concluded that the use of Machine Learning algorithms for intrusion detection system is optimal and Machine Learning techniques provides accurate results with very less false-positive rates and false-negative rates.Deep Learning techniques are also being applied to the two datasets. The results obtained on the TON_IOT Dataset are optimal and accurate. MLP obtained an accuracy of nearly 99.2% on the TON_IOT Dataset whereas LSTM obtained 99% on TON_IOT .It is evident from the results that the Decision Tree for Machine Learning and MLP and LSTM for Deep Learning provides accurate optimal results.

Machine Learning provides efficient and accurate techniques for detecting intrusion in the network. The algorithms like Decision Tree, KNN, and MLP provide good results along with accuracy. So we can say that Machine Learning provides a good basis for intrusion detection in the network. Moreover proposed model could be implemented for the detection of unknown attacks in the network in real-time.

Supplementary Materials: The following supporting information can be downloaded at: www.mdpi.com/xxx/s1, Figure S1: title; Table S1: title; Video S1: title.

Author Contributions: For research articles with several authors, a short paragraph specifying their individual contributions must be provided. The following statements should be used "Conceptualization, X.X. and Y.Y.; methodology, X.X.; software, X.X.; validation, X.X., Y.Y. and Z.Z.; formal analysis, X.X.; investigation, X.X.; resources, X.X.; data curation, X.X.; writing—original draft preparation, X.X.; writing—review and editing, X.X.; visualization, X.X.; supervision, X.X.; project administration, X.X.; funding acquisition, Y.Y. All authors have read and agreed to the published version of the manuscript." Please turn to the CRediT taxonomy for the term explanation. Authorship must be limited to those who have contributed substantially to the Work reported.

Funding: Please add: "This research received no external funding" or "This research was funded by NAME OF FUNDER, grant number XXX" and "The APC was funded by XXX". Check carefully that the details given are accurate and use the standard spelling of funding agency names at https://search.crossref.org/funding. Any errors may affect your future funding.

Institutional Review Board Statement: In this section, you should add the Institutional Review Board Statement and approval number, if relevant to your study. You might choose to exclude this statement if the study did not require ethical approval. Please note that the Editorial Office might ask you for further information. Please add "The study was conducted in accordance with the Declaration of Helsinki, and approved by the Institutional Review Board (or Ethics Committee) of NAME OF INSTITUTE (protocol code XXX and date of approval)." for studies involving humans. OR "The animal study protocol was approved by the Institutional Review Board (or Ethics Committee) of NAME OF INSTITUTE (protocol code XXX and date of approval)." for studies involving animals. OR "Ethical review and approval were waived for this study due to REASON (please provide a detailed justification)." OR "Not applicable" for studies not involving humans or animals.

Informed Consent Statement: Any research article describing a study involving humans should contain this statement. Please add "Informed consent was obtained from all subjects involved in the study." OR "Patient consent was waived due to REASON (please provide a detailed justification)." OR "Not applicable." for studies not involving humans. You might also choose to exclude this statement if the study did not involve humans.

Written informed consent for publication must be obtained from participating patients who can be identified (including by the patients themselves). Please state "Written informed consent has been obtained from the patient(s) to publish this paper" if applicable.

Data Availability Statement: In this section, please provide details regarding where data supporting reported results can be found, including links to publicly archived datasets analyzed or generated during the study. Please refer to suggested Data Availability Statements in section "MDPI Research Data Policies" at https://www.mdpi.com/ethics. If the study did not report any data, you might add "Not applicable" here.

Acknowledgments: In this section, you can acknowledge any support given which is not covered by the author contribution or funding sections. This may include administrative and technical support, or donations in kind (e.g., materials used for experiments).

Conflicts of Interest: Declare conflicts of interest or state "The authors declare no conflict of interest." Authors must identify and declare any personal circumstances or interest that may be perceived as inappropriately influencing the representation or interpretation of reported research results. Any role of the funders in the design of the study; in the collection, analyses or interpretation of data; in the writing of the manuscript, or in the decision to publish the results must be declared in this section. If there is no role, please state "The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results".

Appendix A

The appendix is an optional section that can contain details and data supplemental to the main text—for example, explanations of experimental details that would disrupt the flow of the main text but nonetheless remain crucial to understanding and reproducing the research shown; figures of replicates for experiments of which representative data is shown in the main text can be added here if brief, or as Supplementary data. Mathematical proofs of results not central to the paper can be added as an appendix.

Appendix B

All appendix sections must be cited in the main text. In the appendices, Figures, Tables, etc. should be labeled starting with "A"—e.g., Figure A1, Figure A2, etc.

References

[1] N. Alkhatib, H. Ghauch, and J.-L. Danger, "SOME/IP Intrusion Detection using Deep Learning-based Sequential Models in Automotive Ethernet Networks," pp. 0954–0962, 2021, doi: 10.1109/iemcon53756.2021.9623129.

[2] P. Tao, Z. H. E. Sun, and Z. Sun, "An Improved Intrusion Detection Algorithm Based on GA and SVM," *IEEE Access*, vol. 6, pp. 13624–13631, 2018, doi: 10.1109/ACCESS.2018.2810198.

[3] G. Kim, S. Lee, and S. Kim, "Expert Systems with Applications A novel hybrid intrusion detection method integrating anomaly detection with misuse detection," *Expert Syst. Appl.*, vol. 41, no. 4, pp. 1690–1700, 2014, doi: 10.1016/j.eswa.2013.08.066.

[4] H. Shapoorifard, "Intrusion Detection using a Novel Hybrid Method Incorporating an Improved KNN," vol. 173, no. 1, pp. 5–9, 2017.

[5] G. Zhao, C. Zhang, and L. Zheng, "Intrusion Detection using Deep Belief Network and Probabilistic Neural Network," pp. 639–642, 2017, doi: 10.1109/CSE-EUC.2017.119.

[6] U. Bashir and M. Chachoo, "Intrusion detection and prevention system: Challenges & opportunities," *2014 Int. Conf. Comput. Sustain. Glob. Dev. INDIACom 2014*, pp. 806–809, 2014, doi: 10.1109/IndiaCom.2014.6828073.

[7] B. R. Raghunath and S. N. Mahadeo, "Network intrusion detection system (NIDS)," *Proc. - 1st Int. Conf. Emerg. Trends Eng. Technol. ICETET 2008*, pp. 1272–1277, 2008, doi: 10.1109/ICETET.2008.252.

[8] J. D. Gadze, A. A. Bamfo-Asante, J. O. Agyemang, H. Nunoo-Mensah, and K. A.-B. Opare, "An Investigation into the Application of Deep Learning in the Detection and Mitigation of DDOS Attack on SDN Controllers," *Technologies*, vol. 9, no. 1, p. 14, 2021, doi: 10.3390/technologies9010014.

[9] 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," *IEEE Access*, vol. 9, pp. 22351–22370, 2021, doi: 10.1109/ACCESS.2021.3056614.

[10] W. Mingzheng, "New Approach for Information Security Evaluation and Management," vol. 25, no. 6, pp. 689–699, 2020.

[11] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep Learning Approach for Intelligent Intrusion Detection System," *IEEE Access*, vol. 7, no. c, pp. 41525–41550, 2019, doi: 10.1109/ACCESS.2019.2895334.

[12] S. Rajagopal, P. P. Kundapur, and K. S. Hareesha, "Towards Effective Network Intrusion Detection: From Concept to Creation on Azure Cloud," *IEEE Access*, vol. 9, pp. 19723–19742, 2021, doi: 10.1109/ACCESS.2021.3054688.

[13] H. Ahmed, G. Elsayed, S. Chaffar, and S. B. Belhaouari, "A two-level deep learning approach for emotion recognition in Arabic news headlines," 2020, doi: 10.1080/1206212X.2020.1851501.

[14] T. Saranyaa *et al.*, "ScienceDirect ScienceDirect Performance Analysis of Machine Learning Algorithms in Intrusion Detection System : A Review Performance Analysis of Machine Learning Algorithms in Intrusion Detection System : A Review," vol. 00, no. 2019, 2020, doi: 10.1016/j.procs.2020.04.133.

[15] Y. Zhang and X. Ran, "A Step-Based Deep Learning Approach for Network Intrusion Detection," *Comput. Model. Eng. Sci.*, vol. 128, no. 3, pp. 1231–1245, 2021, doi: 10.32604/cmes.2021.016866.

[16] X. Gao, C. Shan, C. Hu, Z. Niu, and Z. Liu, "An Adaptive Ensemble Machine Learning Model for Intrusion Detection," *IEEE Access*, vol. 7, pp. 82512–82521, 2019, doi: 10.1109/ACCESS.2019.2923640.

[17] J. H. Ring, C. M. Van Oort, S. Durst, V. White, J. P. Near, and C. Skalka, "Methods for Host-based Intrusion Detection with Deep Learning," *Digit. Threat. Res. Pract.*, vol. 2, no. 4, pp. 1–29, 2021, doi: 10.1145/3461462.

[18] A. Devarakonda, N. Sharma, P. Saha, and S. Ramya, "Network intrusion detection : a comparative study of four classifiers using the NSL-KDD and KDD' 99 datasets Network intrusion detection : a comparative study of four classifiers using the NSL-KDD and KDD' 99 datasets," 2022, doi: 10.1088/1742-6596/2161/1/012043.

[19] L. Ashiku and C. Dagli, "ScienceDirect ScienceDirect Network Intrusion Detection System using Deep Learning Network Intrusion Lirim Detection System using Deep Learning," *Procedia Comput. Sci.*, vol. 185, no. June, pp. 239–247, 2021, [Online]. Available: https://doi.org/10.1016/j.procs.2021.05.025.

[20] C. Tang, N. Luktarhan, and Y. Zhao, "Saae-dnn: Deep learning method on intrusion detection," *Symmetry (Basel).*, vol. 12, no. 10, pp. 1–20, 2020, doi: 10.3390/sym12101695.

[21] V. F. Dr. Vladimir, "済無No Title No Title No Title," *Gastron. ecuatoriana y Tur. local.*, vol. 1, no. 69, pp. 5–24, 1967.

[22] C. Paper, A. Y. Javaid, and W. Sun, "A Deep Learning Approach for Network Intrusion Detection System Presented By :," no. January 2016, 2015, doi: 10.4108/eai.3-12-2015.2262516.

[23] O. Faker and E. Dogdu, "Intrusion detection using big data and deep learning techniques," *ACMSE 2019 - Proc. 2019 ACM Southeast Conf.*, pp. 86–93, 2019, doi: 10.1145/3299815.3314439.

[24] D. Park, S. Kim, and H. Kwon, "Host-Based Intrusion Detection Model Using Siamese Network," vol. 9, 2021, doi: 10.1109/ACCESS.2021.3082160.

[25] S. M. Istiaque, A. I. Khan, Z. Al Hassan, and S. Waheed, "Performance Evaluation of a Smart Intrusion Detection System (IDS) Model," *Eur. J. Eng. Technol. Res.*, vol. 6, no. 2, pp. 148–152, 2021, doi: 10.24018/ejers.2021.6.2.2371.

[26] I. Data, "A Deep Learning Model for Network Intrusion Detection with Imbalanced Data," pp. 1–13, 2022.

[27] Y. Tang, "Deep Stacking Network for Intrusion Detection," 2022.

[28] S. Ullah *et al.*, "HDL-IDS: A Hybrid Deep Learning Architecture for Intrusion Detection in the Internet of Vehicles," *Sensors*, vol. 22, no. 4, pp. 1–20, 2022, doi: 10.3390/s22041340.

[29] K. Albulayhi, Q. Abu Al-Haija, S. A. Alsuhibany, A. A. Jillepalli, M. Ashrafuzzaman, and F. T. Sheldon, "IoT Intrusion Detection Using Machine Learning with a Novel High Performing Feature Selection Method," *Appl. Sci.*, vol. 12, no. 10, p. 5015, 2022, doi: 10.3390/app12105015.

[30] A. Halbouni, T. S. Gunawan, M. H. Habaebi, M. Halbouni, M. Kartiwi, and R. Ahmad, "Machine Learning and Deep Learning Approaches for CyberSecurity: A Review," *IEEE Access*, vol. 10, no. January, pp. 19572–19585, 2022, doi: 10.1109/ACCESS.2022.3151248.

[31] T. Kim and W. Pak, "Robust Network Intrusion Detection System Based on Machine-Learning With Early Classification," *IEEE Access*, vol. 10, pp. 10754–10767, 2022, doi: 10.1109/ACCESS.2022.3145002.