Journal of Computing Research and Innovation (JCRINN) Vol. 7 No. 1 (2022) (pp124-137)
https://jcrinn.com : eISSN: 2600-8793 / 10.24191/jcrinn.v7i1.274
https://doi.org/10.24191/jcrinn.v7i1.274

An Analysis of Intrusion Detection Classification using Supervised Machine Learning Algorithms on NSL-KDD Dataset

Sarthak Rastogi[1], Archit Shrotriya[2], Mitul Kumar Singh[3], P. Raghu Vamsi[4*]
*[1,2,3, *4] Department of CSE, Jaypee Institute of Information Technology, Sector 62, NOIDA, India*

*Corresponding author: * prvonline@yahoo.co.in*
Received Date: 28 January 2022
Accepted Date: 5 February 2022
Revised Date: 10 February 2022
Published Date: 1 March 2022

HIGHLIGHTS

- Machine learning was used to predict the intrusions in the NSL-KDD dataset.
- Investigated the supervised learning algorithms applied on various benchmark network traffic datasets.
- Studied both the binary and multi class classification methods for intrusion detection.
- Studied the behaviour of the supervised learning algorithms in detecting and classifying the abnormal traffic in terms of accuracy.

ABSTRACT

From the past few years, Intrusion Detection Systems (IDS) are employed as a second line of defence and have shown to be a useful tool for enhancing security by detecting suspicious activity. Anomaly based intrusion detection is a type of intrusion detection system that identifies anomalies. Conventional IDS are less accurate in detecting anomalies because of the decision taking based on rules. The IDS with machine learning method improves the detection accuracy of the security attacks. To this end, this paper studies the classification analysis of intrusion detection using various supervised learning algorithms such as SVM, Naive Bayes, KNN, Random Forest, Logistic Regression and Decision tree on the NSL-KDD dataset. The findings reveal which method performed better in terms of accuracy and running time.

Keywords: *NSL-KDD, Intrusion Detection System, Machine Learning, Anomaly, SVM, Naive Bayes, KNN, Random Forest, Logistic Regression, Decision Tree.*

INTRODUCTION

It is unavoidable in today's world for a person to be subjected to a Cyber attack in some form or another. With the easy availability of Internet at low cost, the number of users exposed to Internet and intrusions has increased rapidly in recent years, requiring the creation of a system that monitors all activities and protects our sensitive information from any anomaly and the risk of it being exposed and falling into the wrong hands. While surfing the internet, a huge number of packets are received and transferred through the user device to the web server. The Intrusion Detection System (IDS) is placed in a network as a second level of defence and keeps track of these packets and network connections. The IDS is broadly classified into two categories based on its position in the network. If the IDS is placed at the network level, preferably at the

entrance level of the internal network, and monitors the suspicious activities then it is said to be Network based IDS (NIDS), on the other hand if the IDS has been installed on the computer and observe for suspicious activities in the system then it is said to be Host based IDS (HIDS).

Based on the nature of detecting the malicious activity IDS is classified into three categories: 1) Signature based IDS, 2) Anomaly based IDS, and 3) Hybrid IDS. Signature based IDSs analyses network traffic or system activities for suspicious behaviour and issues an alarm based on the rules or signature specified during the configuration. The anomaly based IDSs focus on protecting the normal behaviour of the system by identifying abnormalities. The identified anomalies or abnormalities are considered as the potential threats to the system and a symptom of security attack. It classifies every packet and activity into normal and abnormal behaviour (Binary classification), and then further classified into a specific type of intrusion refers to a sequence of actions aimed at compromising conventional security properties such as the integrity, confidentiality, or availability of any resource on a computing platform. Some examples of such intrusion attacks are Denial of Service (DoS), Probing attack (Probe), User to Root (U2R), and Remote to Local (R2L). The way of classification of the abnormal traffic to a specific category of attack is said to be Multi Class classification. These methods may result in a high false alarm rate if they are poorly designed. Finally, the hybrid IDSs combine the advantages of the both signature based system and anomaly based classification to improve the accuracy of detecting the abnormalities as compared to previous two IDS models.

The reset of the paper is organised as follows: After presenting methodology of the work, the literature review is presented. Descriptions of dataset and supervised algorithms considered for the study are presented in the later section followed by the results of the simulation study are presented. Conclusion and future work is presented in the final section.

METHODOLOGY

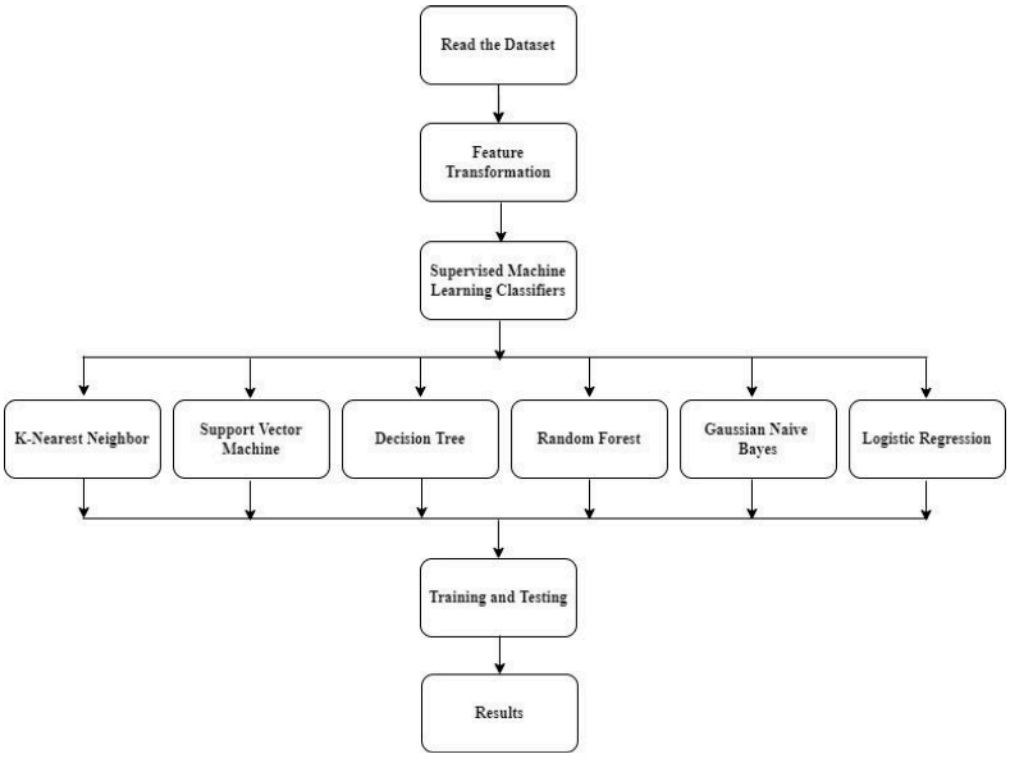

Figure 1: Methodology of the study

The main area of focus in this study is with anomaly based IDS. Intelligent algorithms are applied with the available data and the data extracted from the network traffic to extract the abnormal traffic patterns. This classification is performed in the literature with various machine learning and deep learning algorithms. The main objective of the anomaly based IDS is to classify abnormal traffic patterns from network traffic. To do this, the IDSs should be capable of detecting the abnormal traffic dynamically or with the help of predefined labels. The algorithms used for the former way of detection are said to be unsupervised learning algorithms and the later way of detection are said to be supervised learning algorithms. Supervised classification algorithms are most suited in-order to develop efficient hybrid IDS systems and to support the signature based methods with anomaly detection. To this end, the contributions of this paper are as follows:

- Investigating the supervised learning algorithms applied on various benchmark network traffic datasets in the literature.
- Studying the behaviour of the supervised learning algorithms in detecting and classifying the abnormal traffic with very low false alarm rate. In this, both the binary and multi-class classification mechanisms have been studied using supervised classification algorithms such as Support Vector Machine (SVM), Naive Bayes, K-Nearest Neighbour (K-NN), Random Forest, Logistic Regression and Decision tree using a well known benchmark network traffic NSL-KDD dataset.

Figure 1 depicts the methodology of the study. In the pre-processing part, the attack types are divided into different classes. The numeric values are normalised in the range [-1, 1] and converted the categorical data into numeric data by adding columns for each of the values in the categorical columns. Two data files were created out of the dataset in which the first file is for binary classification and the next file is for multiclass classification experimentation. In the binary classification file, two columns namely normal and abnormal are added with value 1 to indicate normal traffic and the value 0 to indicate attack. Similarly files are prepared for multiclass classification with 5 columns in which one column for normal traffic and one each for the attack class. After the files are prepared, they are imported to the main program and split into 75% for training and 25% for testing. The supervised classification algorithms are used on the trained and tested parts and calculated the training, testing time and accuracy score and plotted the graph for how different algorithms performed. This study considered the accuracy score for comparison and it is calculated as follows

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (1)$$

Where TP = True Positive, TN = True Negative, FN = False Negative, FP = False Positive.

LITERATURE REVIEW

Ravipati et al. (Ravipati & Abualkibash, 2019) studied in detail how the NSL-KDD changed from the original KDD dataset and states its advantages over the original. This work also conducted a survey of how different classification algorithms work for the dataset. Authors implemented J48, SVM and Naïve Bayes algorithms and used only 6 features of the many provided by the dataset. The analysis results on the NSL-KDD dataset show that it has improved significantly from the original KDD dataset and is a great candidate

data set to simulate and test the performance of IDS. Dhanabal et al. (Dhanabal & Shantharajah, 2015) implemented multiple types of machine learning algorithms against both KDD and NSL-KDD datasets and provided the results of the accuracy score and false alarm rate in tabular and graph form. In order to strengthen the network from illegal access the concept of IDS (Intrusion Detection System) is gaining popularity around the world. The applications of data mining in the computer security field improves the development of IDS in order to work on these applications it is essential to classify the degree of attacks in IDS and use it through data mining. Despite the use of IDS, we cannot be completely certain about its functioning and results of IDS use have been uncertain (Rashid, Siddique, & Ahmed, 2020; Park, et al., 2021; Kumar, Gupta, & Arora; Saha, 2021; Singhal, Gupta, Sharma, Sharma, & Rana, 2021; Haq, et al., 2015; Sekhar & Rao, 2019; Vamsi & Chahuan, 2020; Zamani & Movahedi, 2013). To plug the loopholes, we need to adjust the detection strategy according to the degree of attack activities to ensure error free results. The goal of a network intrusion detection system is to discover unauthorised access to a computer network by analysing traffic on the network for signs of malicious activity (Rashid, Siddique, & Ahmed, 2020; Heine, Laue, & Kleiner, 2020; Chauhan & Vamsi, 2019; Aziz & Abdulazeez, 2021; Ahmad, Shahid Khan, Wai Shiang, Abdullah, & Ahmad, 2021; Mahfouz, Venugopal, & Shiva, 2020; Liu & Lang, 2019; Dhanabal & Shantharajah, 2015). The intrusion detection task is to build a predictive model capable of distinguishing between intrusions or attacks, and normal network connections. Web application threats have become a prime concern for information security. IDS are one of the security mechanisms used to guard these applications against attacks. However, the methodology has been primarily used for monitoring the network-based attacks. Designing suitable IDS to prevent web-based attacks still needs more focus by the interest groups (Sharma, Gigras, Chhikara, & Dhull, 2019; Thomas & Pavithran, 2018; Negandhi, Trivedi, & Mangrulkar, 2019; Abrar, Ayub, Masoodi, & Bamhdi, 2020; Gurung, Ghose, & Subedi, 2019; Ding & Zhai, 2018; Ever, Sekeroglu, & Dimililer, 2019; Masoodi & others, 2021).

DESCRIPTION OF NSL-KDD DATASET

The NSL-KDD dataset from the Canadian Institute for Cyber security (the updated version of the original KDD Cup 1999 Data (KDD99) is used in this project. This study used the KDDTrain+ dataset for both training and testing by splitting it in 75% and 25% ratio. There exists 4 different classes of attacks in the dataset namely 1) Denial of Service (DoS), 2) Probe, 3) User to Root (U2R), and 4) Remote to Local (R2L). A brief description of each attack is as follows (Aziz & Abdulazeez, 2021):

- DoS is an attack that tries to shut down traffic flow to and from the target system. The IDS is flooded with an abnormal amount of traffic, as a result the system is unable to handle the requests and shutdown to protect itself. This prevents normal traffic from entering a network. This is the most common attack in the data set.
- An attack that attempts to gather information from a network is known as a probe or surveillance attack. The purpose of this assault is to impersonate an attacker and steal sensitive information such as customer personal information or financial data.
- U2R is an attack that starts with a regular user account and attempts to get super-user access to the system or network (root). The attacker tries to get root privileges or access to a system by exploiting vulnerabilities.
- R2L is a method of gaining local access to a distant machine. An attacker does not have local access to the system/network, and tries to "hack" their way into the network.

Table 1 shows the count of the cases per attack category available in the dataset. Figure 2 shows that the cases of attack class combine to 46.54% of the dataset and total normal cases are 53.46%. Figure 3 shows the distribution of multi-class labels in the dataset. The dataset is made up of 21 different attacks which come under the classes mentioned in Figure 3 are shown in Table 2.

Journal of Computing Research and Innovation (JCRINN) Vol. 7 No. 1 (2022) (pp124-137)
https://jcrinn.com : eISSN: 2600-8793 / 10.24191/jcrinn.v7i1.274
https://doi.org/10.24191/jcrinn.v7i1.274

Table 1 Count of cases per attack category

Normal	67343
DoS	45927
Probe	11656
R2L	995
U2R	52

Table 2 Different attacks in each attack class

DoS	Back, Land, Neptune, Pod, Smurf, Teardrop, Apache2, Udpstorm, Processtable, Worm
Probe	Satan, Ipsweep, Nmap, Portsweep, Mscan, Saint
R2L	Guess_Password, Ftp_write, Imap, Phf, Multihop, Warezmaster, Warezclient, Spy, Xlock, Xsnoop, Snmpguess, Snmpgetattack, Httptunnel, Sendmail, Named
U2R	Buffer_overflow, Loadmodule, Rootkit, Perl, Sqlattack, Xterm, Ps

Journal of Computing Research and Innovation (JCRINN) Vol. 7 No. 1 (2022) (pp124-137)
https://jcrinn.com : eISSN: 2600-8793 / 10.24191/jcrinn.v7i1.274
https://doi.org/10.24191/jcrinn.v7i1.274

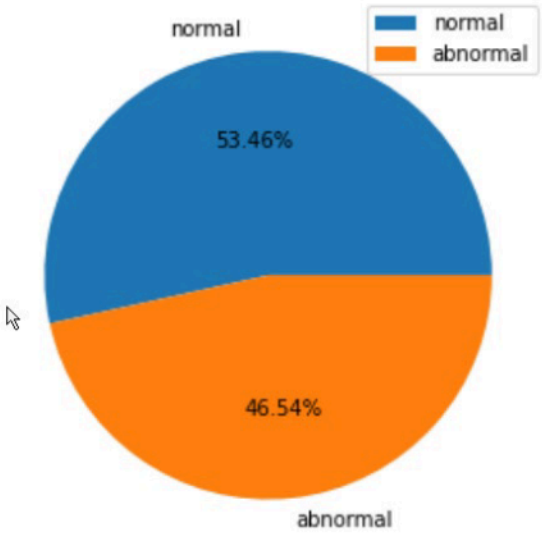

Figure 2: Distribution of normal and abnormal labels

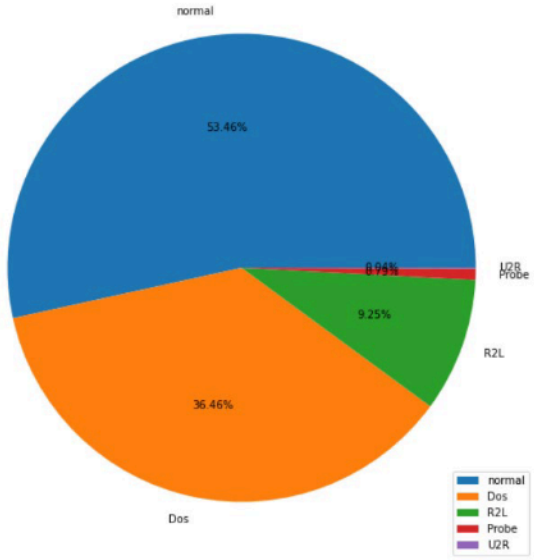

Figure 3: Distribution of multi-class labels

SUPERVISED LEARNING ALGORITHMS

K-Nearest Neighbour Algorithm (K-NN)

K-NN is a supervised machine learning classification algorithm. KNN algorithm assumes the similarity between the test data and available trained data and places the test point into the category that is most similar to the available categories. It is known as a lazy learner algorithm because it does not learn from the training set immediately rather it stores the dataset and when it gets the new data then it classifies that data according to the category similar to the new data. This study observes how long the testing time is compared to the training time. K-NN is a non-parametric algorithm, which means it does not make any assumption on underlying data. This algorithm works by calculating the distance between the test case and the trained data points and gets the k-closest points. K is a user defined value. Out of those points it counts how many points

belong to which category and assigns to the test case the category with the maximum count. The distance d is calculated using the Euclidean distance formula between two points $(x1, x2)$ and $(x2, y2)$ as shown below

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (2)$$

This study took a range of values of k to find one optimal value. To choose the optimal k value, a graph of k against the error rate is plotted and selected the one corresponding to the minimum error rate. It is observed that for the range 5 to 15, k=7 give the least error rate and hence have been selected for the comparison with other algorithms in case of binary and multi-class classification.

Logistic regression

Logistic regression is a Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables. Logistic regression predicts the output of a categorical dependent variable. Therefore, the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, True or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.

Consider the hypothesis function,

$$0 \le hx \le 1 \quad (3)$$

Here, $h_\theta(x) = g(\theta^T x)$ is the sigmoid function

$$g(z) = \frac{1}{1 + e^{-z}} \quad (4)$$

A threshold can be set to predict which class a data belongs to. Based upon the threshold, the obtained estimated probability is classified into classes. Decision boundaries can be linear or non-linear. Polynomial order can be increased to get complex decision boundaries. Figure 5 shows the graphical representation of the sigma function. For the multi-class classification, it uses the one vs all approach. It trains a logistic regression classifier for each class i to predict the probability that y=i. To make prediction on the test case x, pick the class i that maximises $h_\theta^i(x)$.

Support Vector Machine

Support Vector Machines are used for classification and regression. This study used SVM for classification. In the SVM based classification, this study plots each data item as a point in n-dimensional space (n=number of features) with the value of each feature being the value of a particular coordinate. Then, the classification is performed by finding the hyper-plane that differentiates the two classes very well i.e., Normal and Abnormal in Binary Classification and DoS, Probe, U2R, R2L in Multi-Class Classification. Support Vector Machines plots data points

on an n-dimensional plane where n is the number of features present in the dataset. Support Vectors that are created when algorithms run on the simply data points representing the individual value of each record. Important terminologies are

- Hyper-plane: There can be multiple lines/decision boundaries to segregate the classes in n-dimensional space.
- Support Vectors: The data points or vectors that are the closest to the hyper-plane and which affect the position of the hyper-plane are termed as Support Vector.
- Margin: The distance between the vectors and the hyper-plane is called the margin.

This study is trying to find the most optimal hyper-plane which classifies the NSL-KDD dataset with the maximum margin possible to get the best accuracy. The following statements show the working of SVM. In this g(x) is the objective function subject to the conditions are provided.

$$g(x) = w^T x + b$$

$$Maximize\ k\ such\ that\ :$$

$$-w^T x + b \geq k\ for\ d_i == 1$$

$$-w^T x + b \leq k\ for\ d_i == -1$$

$$Value\ of\ g(x)\ depends\ upon\ ||w||\ :$$

1) $Keep\ ||w|| = 1\ and\ maximize\ g(x)\ or,$

2) $g(x) \geq 1\ and\ minimize\ ||w||$

Random Forest

Random Forest (RF) is widely used for Classification and regression. It basically builds many random Decision Trees from the given data and takes the majority vote for classification and the average for regression. Taking a real-life example for understanding Random Forest will help us get the concept more clearly: A 10+2 Pass student has to decide which field of engineering he wants to go into so he decides to ask his relatives, friends, classmates and Teachers. Some people told him to choose core engineering, some told him to go into the more technical side and majority of them told him to take Computer Science as his career field. So, he chose computer science for his Bachelor's Degree.

Journal of Computing Research and Innovation (JCRINN) Vol. 7 No. 1 (2022) (pp124-137)
https://jcrinn.com : eISSN: 2600-8793 / 10.24191/jcrinn.v7i1.274
https://doi.org/10.24191/jcrinn.v7i1.274

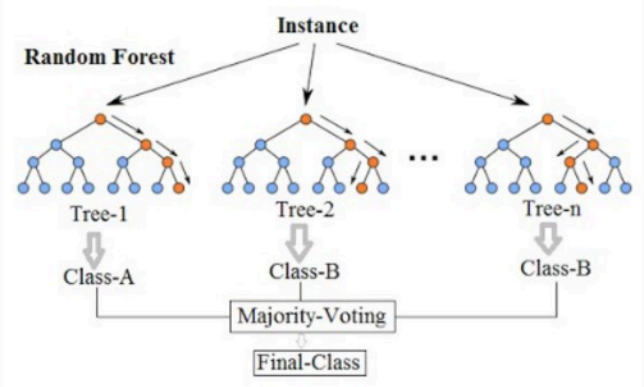

Figure 4: Simplified Random Forest

As shown in Figure 4, the RF algorithm uses an ensemble technique called Bagging in which the dataset is divided into many random decision trees and takes a majority vote for the outcome.

Algorithm for Random Forest is as follows

1. N numbers of random decision trees are made from the given dataset.
2. Individual trees are constructed for each sample.
3. Final output is considered on the basis of majority outcomes from the randomly generated n number of decision trees.

The following are important features of the RF:

1. Diverse Outcome- Not all attributes/variables/features are considered while making an individual tree, each tree is different.
2. Parallelization-Each tree is created independently out of different data and attributes. This means that we can make full use of the CPU to build random forests.
3. Train-Test split- In a random forest we don't have to segregate the data for train and test as there will always be 30% of the data which is not seen by the decision tree.

Gaussian Naive Bayes

It is based on applying Bayes' theorem with the "naive" assumption of conditional independence between every pair of features given the value of the class variable. Bayes' theorem states the following relationship, given class variable y and dependent feature vector x1 through xn as shown in (5).

$$P(y \mid x_1, \ldots, x_n) = \frac{P(y)P(x_1, \ldots, x_n \mid y)}{P(x_1, \ldots, x_n)} \quad (5)$$

Naive Bayes is a supervised classifying set of learning algorithms. The likelihood of the features is assumed to be Gaussian. The likelihood ratio is given in (6)

Journal of Computing Research and Innovation (JCRINN) Vol. 7 No. 1 (2022) (pp124-137)
https://jcrinn.com : eISSN: 2600-8793 / 10.24191/jcrinn.v7i1.274
https://doi.org/10.24191/jcrinn.v7i1.274

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right) \quad (6)$$

The parameters σ_y and μ_y are estimated using maximum likelihood. But this study considers the Gaussian Naive Bayes. In Gaussian Naive Bayes, continuous values associated with each feature are assumed to be distributed according to a Gaussian distribution. A Gaussian distribution is also called Normal distribution. Assumption of Naïve is that it is independent among the features. Hence, we split evidence into the independent parts as for any two independent events A and B the P(A,B) is as follows
P(A, B) = P(A).P(B)

Decision tree

A Decision tree is a flowchart like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label. The construction of a decision tree classifier does not require any domain knowledge or parameter setting, and therefore is appropriate for exploratory knowledge discovery. Decision trees can handle high dimensional data. Decision trees classify instances by sorting them down the tree from the root to some leaf node, which provides the classification of the instance. This process is then repeated for the sub-tree rooted at the new node. Consider the following example and Figure 5

(Outlook = Sunny ^ Humidity = Normal) v (Outlook = Overcast) v (Outlook = Rain ^ Wind = Weak)

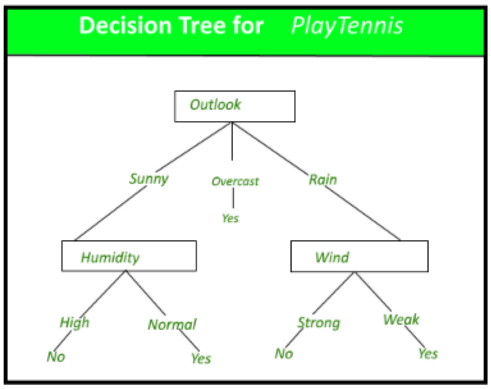

Figure 5: Decision Tree for the Play Tennis example

Figure 5 is a decision tree diagram for whether we can play tennis on that day or not, and the outcomes for different combinations of features would be different, but in a single diagram, we can observe different path situations which would produce different outcomes and thus would give different results.

SIMULATION RESULTS

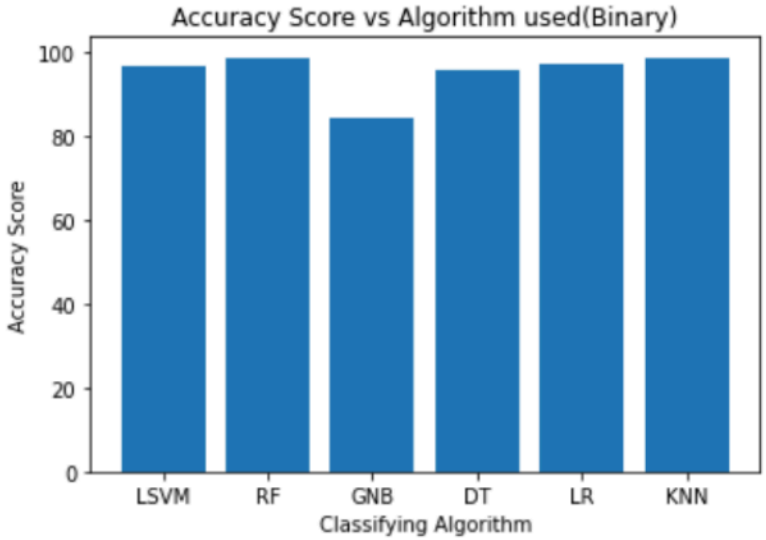

Figure 6: Accuracy score of algorithms for binary classification

Table 3: Performance of Supervised learning algorithms (in case of binary classification)

Algorithm	Training Time (s)	Testing Time (s)	Accuracy (%)
K-NN	0.030003309	140.0600479	98.59655807
SVM	232.3176758	11.44427609	96.6977837
DT	0.713073969	0.014960766	95.83412714
RF	3.51273632	0.127059221	98.70451515
GNB	0.194990873	0.085766792	84.32717343
LR	4.894774437	0.01561904	96.97085159

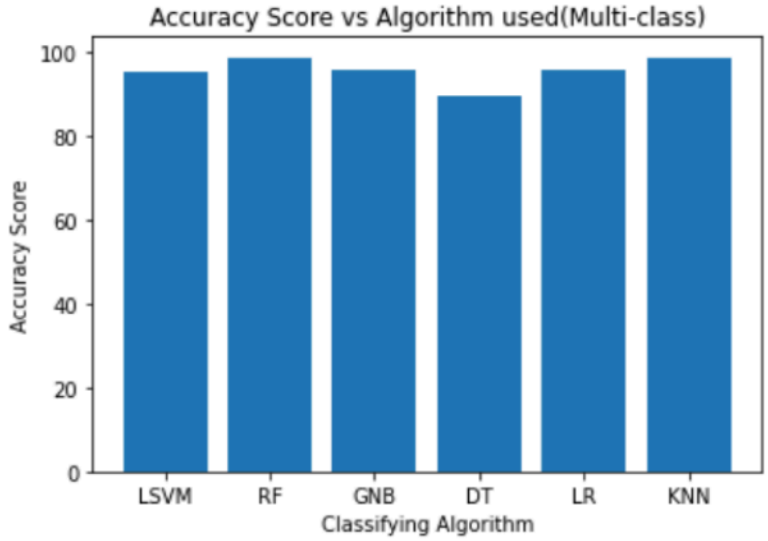

Figure 7: Accuracy score of algorithms for multi-class classification

Table 4: Performance of Supervised learning algorithms (in case of multi-class classification)

Algorithm	Training Time (s)	Testing Time (s)	Accuracy (%)
K-NN	0.015621901	72.06326985	98.28221248

SVM	201.615521	22.26730466	95.24988887
DT	0.591451645	0.013961315	89.59166825
RF	3.55549407	0.158385515	98.46954976
GNB	0.346791506	0.534326792	95.4213501
LR	27.36496043	0.01393342	95.4213501

Results summary

The results of the simulation have been presented in Figure 6 and 7, and Tables 3 and 4 respectively. It is observed that the random forest classifier shows the highest accuracy score and is faster compared to some of the other supervised algorithms for both binary and multi-class classification. Even though SVM shows a decent accuracy score, it was the slowest among all the algorithms employed. KNN showed the highest testing time, but it makes up for it with a high accuracy score. The Gaussian Naïve Bayes classifier resulted in the poor accuracy score for the binary classification whereas the Decision Tree classifier resulted in the poor accuracy score for the multi-class classification. Therefore, it is recommended that among the studied algorithms, implementing IDS should be done using the random forest algorithm.

CONCLUSION AND FUTURE WORK

In this paper, we have presented an overview of multiple supervised machine learning techniques for Intrusion Detection Systems (IDS) and distinct detection methodologies as well as classifiers for the NSL-KDD dataset. The ways it can detect the intrusion are provided based on a study of supervised machine learning techniques. The study has been conducted for both multi-class and binary classification. When compared to other supervised algorithms, the experiment results demonstrate that KNN has a high accuracy in detecting intrusion. It is recommended that among the studied algorithms, implementation of IDS should be done using the random forest algorithm. In the future, we attempt to conduct a survey with other types of machine learning algorithms and techniques to study an intrusion detection model having better accuracy.

ACKNOWLEDGEMENT

The authors appreciate the reviewers for their contributions towards improving the quality of this research.

CONFLICT OF INTEREST DISCLOSURE

All authors declare that they have no conflicts of interest to disclose.

REFERENCES

Abrar, I., Ayub, Z., Masoodi, F., & Bamhdi, A. M. (2020). A machine learning approach for intrusion detection system on NSL-KDD dataset. *2020 International Conference on Smart Electronics and Communication (ICOSEC)*, (pp. 919–924).

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*, e4150.

Aziz, Z. A., & Abdulazeez, A. M. (2021). Application of Machine Learning Approaches in Intrusion Detection System. *Journal of Soft Computing and Data Mining, 2*, 1–13.

Chauhan, A., & Vamsi, P. R. (2019). Anomalous Ozone Measurements Detection Using Unsupervised Machine Learning Methods. *2019 International Conference on Signal Processing and Communication (ICSC)*, (pp. 69–74).

Dhanabal, L., & Shantharajah, S. P. (2015). A study on NSL-KDD dataset for intrusion detection system based on classification algorithms. *International journal of advanced research in computer and communication engineering, 4*, 446–452.

Ding, Y., & Zhai, Y. (2018). Intrusion detection system for NSL-KDD dataset using convolutional neural networks. *Proceedings of the 2018 2nd International Conference on Computer Science and Artificial Intelligence*, (pp. 81–85).

Ever, Y. K., Sekeroglu, B., & Dimililer, K. (2019). Classification analysis of intrusion detection on NSL-KDD using machine learning algorithms. *International Conference on Mobile Web and Intelligent Information Systems*, (pp. 111–122).

Gurung, S., Ghose, M. K., & Subedi, A. (2019). Deep learning approach on network intrusion detection system using NSL-KDD dataset. *International Journal of Computer Network and Information Security, 11*, 8–14.

Haq, N. F., Onik, A. R., Hridoy, M. A., Rafni, M., Shah, F. M., & Farid, D. M. (2015). Application of machine learning approaches in intrusion detection system: a survey. *IJARAI-International Journal of Advanced Research in Artificial Intelligence, 4*, 9–18.

Heine, F., Laue, T., & Kleiner, C. (2020). On the Evaluation and Deployment of Machine Learning Approaches for Intrusion Detection. *2020 IEEE International Conference on Big Data (Big Data)*, (pp. 4594–4603).

Kumar, S., Gupta, S., & Arora, S. (n.d.). A comparative simulation of normalization methods for machine learning-based intrusion detection systems using KDD Cup99 dataset. *Journal of Intelligent & Fuzzy Systems*, 1–18.

Liu, H., & Lang, B. (2019). Machine learning and deep learning methods for intrusion detection systems: A survey. *applied sciences, 9*, 4396.

Mahfouz, A. M., Venugopal, D., & Shiva, S. G. (2020). Comparative analysis of ML classifiers for network intrusion detection. *Fourth international congress on information and communication technology*, (pp. 193–207).

Mahmood, R. A., Abdi, A. H., & Hussin, M. (2021). Performance Evaluation of Intrusion Detection System using Selected Features and Machine Learning Classifiers. *Baghdad Science Journal, 18*, 0884–0884.

Masoodi, F., & others. (2021). Machine Learning for Classification analysis of Intrusion Detection on NSL-KDD Dataset. *Turkish Journal of Computer and Mathematics Education (TURCOMAT), 12*, 2286–2293.

Negandhi, P., Trivedi, Y., & Mangrulkar, R. (2019). Intrusion detection system using random forest on the NSL-KDD dataset. In *Emerging Research in Computing, Information, Communication and Applications* (pp. 519–531). Springer.

Park, D., Ryu, K., Shin, D., Shin, D., Park, J., & Kim, J. (2021). A Comparative Study of Machine Learning Algorithms Using LID-DS DataSet. *KIPS Transactions on Software and Data Engineering, 10*, 91–98.

Rashid, A., Siddique, M. J., & Ahmed, S. M. (2020). Machine and deep learning based comparative analysis using hybrid approaches for intrusion detection system. *2020 3rd International Conference on Advancements in Computational Sciences (ICACS)*, (pp. 1–9).

Ravipati, R. D., & Abualkibash, M. (2019). Intrusion detection system classification using different machine learning algorithms on KDD-99 and NSL-KDD datasets-a review paper. *International Journal of Computer Science & Information Technology (IJCSIT) Vol, 11*.

Saha, B. (2021). Comparison Analysis of Classification Algorithms for Intrusion Detection.

Sekhar, C. H., & Rao, K. V. (2019). A Study: Machine Learning and Deep Learning Approaches for Intrusion Detection System. *International Conference on Computer Networks and Inventive Communication Technologies*, (pp. 845–849).

Sharma, S., Gigras, Y., Chhikara, R., & Dhull, A. (2019). Analysis of NSL KDD dataset using classification algorithms for intrusion detection system. *Recent Patents on Engineering, 13*, 142–147.

Singhal, A., Gupta, I., Sharma, U., Sharma, M., & Rana, A. (2021). Experimental Analysis of various Machine Learning approaches for Intrusion Detection. *2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO)*, (pp. 1–7).

Thomas, R., & Pavithran, D. (2018). A survey of intrusion detection models based on NSL-KDD data set. *2018 Fifth HCT Information Technology Trends (ITT)*, (pp. 286–291).

Vamsi, P. R., & Chahuan, A. (2020). Machine learning based hybrid model for fault detection in wireless sensors data. *EAI Endorsed Transactions on Scalable Information Systems, 7*.

Zamani, M., & Movahedi, M. (2013). Machine learning techniques for intrusion detection. *arXiv preprint arXiv:1312.2177*.