International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

MLIDS: A Machine Learning-Based Intrusion Detection System Using the NSLKDD Data

Dr. Bere Sachin Sukhadeo[1], Dr. Ratna Nitin Patil[2], Dr. Reshma Atole[3], Dr. Yogita Deepak Sinkar[4], Dr. Uday Chandrakant Patkar[5], Dr. Rupali Chopade[6]

Submitted: 07/09/2023 **Revised**: 20/10/2023 **Accepted**: 06/11/2023

Abstract: In order to protect computer networks from malicious activity, intrusion detection systems (IDS) are essential. Traditional rule-based IDS frequently experience significant false positive rates and struggle to keep up with changing attack techniques. This paper, utilizing the NSL-KDD dataset, suggests a machine learning strategy for intrusion detection to overcome these issues. The suggested method makes use of the capabilities of machine learning algorithms to efficiently identify and categorise network intrusions. The proposed model is trained on and tested against the NSL-KDD dataset, a benchmark dataset for intrusion detection research. The data was split into four feature subsets that were taken from the NSL-KDD dataset in order to evaluate the performance of these classifiers. Preprocessing techniques were used to remove pointless attributes from the dataset because an IDS's performance is influenced by the dimensions of the data. In this study, multiple machine learning (ML) classifiers were used to categorise data in an intrusion detection system (IDS) as either normal or invasive. The recommended machine learning-based IDS works brilliantly in terms of various accuracy parameters, according to testing results. When compared to traditional rule-based systems, the suggested method has improved detection rates and fewer false positives, improving the overall security of computer networks.

Keywords: *Intrusion detection system, Machine Learning, Classification, SVM, NB, LR, DT*

I. Introduction

Computer networks are continuously at risk of criminal activity and intrusions in today's connected society. These invasions may cause service interruptions, data breaches, and financial losses. The evolution of attacks makes it difficult for traditional rule-based IDS to stay up, which results in a lot of false positives and poor detection precision. More advanced and intelligent intrusion detection techniques are therefore becoming increasingly necessary. Network security is only one of the many industries where machine learning (ML) techniques have shown to be useful tools. The ability of ML algorithms to learn patterns and behaviours from data allows them to detect intrusions and find anomalies that traditional IDS may overlook. By exploiting the capabilities of ML algorithms, our goal is to increase the accuracy and effectiveness of intrusion detection systems in order to increase the overall security of data networks. Every machine learning system must include a data preparation phase in order to ensure the accuracy and suitability of the data used to train the models. Preprocessing in the context of intrusion detection entails reducing noise and unnecessary data from the NSL-KDD dataset. A more targeted and effective feature set is produced by removing unrelated qualities that do not aid in the identification of intrusions.

Then, the preprocessed dataset is employed with attribute choice algorithms to select the most enlightening and discriminating characteristics. By reducing the number of data dimensions, feature selection enhances the performance of ML

[1]*Associate Professor, HOD Computer Engineering Department, Dattakala Group of Institutions Faculty of Engineering, Bhigwan.*
[2]*Associate Professor, Vishwakarma Institute of Information Technology, Pune, Maharashtra, India*
[3]*Associate Professor, SVPM's College Of Engineering, Baramati, Maharashtra, India*
[4]*HOD & Associate Professor, SVPM's College Of Engineering, Baramati, Pune, Maharashtra, India*
[5]*HOD, Department of Computer Engineering, Bharati Vidyapeeth's College of Engineering Lavale, Pune, Maharashtra, India.*
[6]*Associate Professor, Marathwada Mitra Mandal's College of Engineering, Pune, Maharashtra, India*
sachinbere@gmail.com, ratna.nitin.patil@gmail.com, reshma174@gmail.com, gtsinkar186@gmail.com, patkarudayc@gmail.com, rupalimchopade@mmcoe.edu.in

algorithms. By choosing characteristics with care, it is possible to improve the IDS's precision, efficiency, and computing complexity. The correctness and relevance of the data used to train the models are ensured by data preparation, which is a crucial step in every machine learning (ML) system. In order to perform intrusion detection, the NSL-KDD dataset must be preprocessed to remove noise and redundant data. Unrelated qualities that are ineffective for identifying intrusions are deleted to provide a feature set that is more focused and efficient. The preprocessed dataset's most informative and discriminative characteristics are subsequently chosen utilising feature selection algorithms. Feature selection lowers the degree of dimensionality of the data, which helps enhance the efficiency of ML algorithms. By selecting features that are relevant to the task at hand, we can improve the IDS's precision and efficiency while decreasing its computing cost.

In this study, a variety of ML classifiers are used to categorise network traffic data as either normal or intrusive. These metrics provide insight into how effectively the IDS detects and classifies intrusions.By contrasting the output from various classifiers, we can determine which intrusion detection model on the NSL-KDD dataset is the most accurate and efficient. The study's conclusions advance our understanding of ML-based intrusion detection systems. We want to develop more robust and intelligent IDS solutions that can swiftly recognise and address network intrusions using ML algorithms and the NSL-KDD dataset.

II. Related Work

The topic of malware detection systems (IDS) has seen significant improvements due to the plethora of studies focusing on the identification and avoidance of different network assaults. A thorough analysis of the significant research in this area is offered, emphasising how machine learning (ML) methods are used to create efficient IDS. As they can effectively handle vast amounts of data, ML methods have become very popular in IDS. Network administrators may use ML approaches to analyse and process enormous volumes of network traffic data, which enables them to recognise and stop possible threats in their tracks. Anomalies and suspicious behaviours that can be signs of an intrusion can be found more easily because to the ability of ML algorithms to learn patterns and behaviours from this data.

Network administrators can prevent attacks by acting quickly and appropriately thanks to the use of ML in IDS. ML algorithms can spot comparable patterns in real-time network traffic by examining the patterns and traits of known assaults, warning managers of potential hazards. Organisations may deploy security measures quickly and effectively thanks to this proactive strategy, which lowers the likelihood of successful invasions. ML methods also provide the benefits of scalability and adaptability. ML algorithms can be trained and updated to recognise and react to these changing threats as network environments change and new attack tactics appear. This versatility guarantees that IDS continue to be efficient in identifying both well-known and new attack patterns, enhancing network security. Additionally, the promise for more precise and reliable intrusion detection is provided by ML approaches. Traditional rule-based IDS frequently experience significant false positive rates and missed detections as a result of their inability to keep up with the constantly evolving attack landscape. On the other side, ML algorithms have a higher ability to recognise minor anomalies and previously unidentified attack patterns, which reduces false positives and raises the total detection rate.

The incorporation of ML into IDS paves the way for the creation of intelligent systems that can analyse network traffic in real-time and quickly detect and stop assaults. Real-time data streams can be used by ML-based IDS to undertake continuous analysis, which enables quick threat identification and prompt action. The Support Vector Machine (SVM) has several uses in the intrusion detection industry. In a study by [14], SVM was used for intrusion detection, and the UNSW dataset was used to assess its performance. With various classifiers like Random Forest (RF), RepTree, and Multi-layer Perceptron (MLP) in comparison, the study similarly achieved an amazing model accuracy of 94%.

Three adapted variants of SVM remainedcharity for intrusion detection in a different study by [15]. The study's extraordinary accuracy of 99.86% was achieved using an enhanced whale optimised SVM approach. On the dataset utilised in the study, the SVM algorithm's changes allowed for extremely accurate intrusion detection. Furthermore, [16]

investigated the use of different ML classifiers for developing and evaluating intrusion detection models. In this investigation, network data packets were captured using the wireshark tool and then fed into ML classifiers as input. The researchers analysed the performance of multiple classifiers to decide which intrusion detection technique worked best in their specific scenario. These experiments show how SVM and other ML classifiers perform well in the intrusion detection space. Researchers were successful in categorising and detecting network intrusions with high accuracy by utilising the characteristics of SVM. The SVM algorithm's enhancements in [15] show that it has the potential to perform intrusion detection jobs even better.

Table 1: Different Intrusion detection machine learning Method for Internet of Things

Paper	The Dataset	Method Used	Algorithm	Result
[21]	KDD Cup99	Metaheuristics method	Linear regression	Accuracy: 71.2, Precision: 64/8, Recall: 62, F1-score: 66
[22]	Self-generated	ML based method	decision Tree	Accuracy: 92.45, Precision: 79.19, Recall: 88, F1-score: 87
[23]	KDD Cup99, NSL	Novel model using ML	2 Dimension Reduction and 2 Classification (TDTC)	Accuracy: 83.56, Precision: 82.57, Recall: 74, F1-score: 74
[24]	Malicia	ML based malware detection system	Advanced malware analysis	Accuracy: 96.95
[25]	ISCX	ML based	K-mean Clustering, DT	Accuracy: 84, Precision: 82, Recall: 81, F1-score: 83
[26]	Self-generated	ML based	(LR) Logistic Regression	Accuracy: 97.3, Precision: 95.49, Recall: 93.10, F1-score: 93.10
[27]	NSL, KDD	Machine learning-based framework (DFEL)	-	Accuracy: 98.86, Precision: 96, Recall: 95.32, F1-score: 93
[17]	MQTT Dataset	DL and ML based	ROC-Curves, the confusion matrix, and the testing time are all missing.	Recall: 95.67, F1-score: 95.67, Accuracy: 99.37, Precision: 96.
[18]	Kaggle and Vx Heaven datasets	ML based Method	Fast and Fuzzy Pattern Tree with Fuzzies	Precision: 99, accuracy: 99.01 F1 score: 99.10, recall: 99.10
[19]	SCADA network traffic	ML and DL based	DBN and SVM	Accuracy: 94.65

Additionally, [16] illustrates the usefulness of ML classifiers in practical network security scenarios by using Wireshark as a tool for capturing network data packets. The researchers gained important insights into the functionality and applicability of several ML classifiers for intrusion detection by developing and evaluating the models using actual network data. The results of these investigations add to the corpus of knowledge on ML-based intrusion detection systems. They show the capability of SVM and other ML classifiers in accurately detecting intrusions and enhancing computer network security. The high accuracy levels attained in these investigations offer encouraging outcomes, demonstrating the potency of ML approaches in addressing the difficulties brought on by network attacks.

III. Dataset Description

The testing, analysis, and evaluation datasets have a big influence on how effectively the detection system works. Numerous investigations have used the well-known NSL-KDD datasets, an improved version of the KDD CUP 99 database. The NSL-KDD dataset addresses the main issues with the KDD CUP 99 dataset by removing duplicate items and choosing data based on relevance.

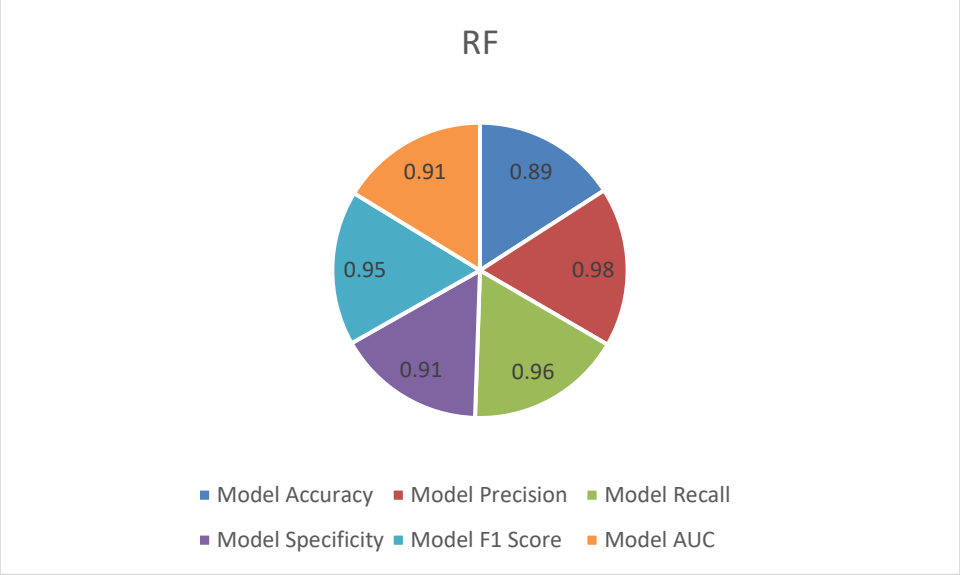

Fig 1: Representation of Dataset NSL- KDD 99

After pre-processing, the NSL-KDD dataset, which comprises 148,517 records, has become a key resource for intrusion detection systems (IDSs). Regular cases and attacks, broken down into Probing, DoS, User to Root (U2R), Remote to Local (R2L), and Normal classes, make up its two main categories. Due to its age and the changing threat landscape, however, its reputation as a go-to dataset for IDSs has been somewhat damaged. The NSL-KDD dataset might no longer accurately represent the real-world scenarios and attack patterns that modern IDSs must deal with as cyber threats continue to grow at a rapid rate. In order to assure the efficiency of their intrusion detection models in modern cybersecurity contexts, researchers and practitioners are increasingly looking for more up-to-current and relevant datasets. As a result, the dataset is now regarded to be slightly out of date.

Table 2: Description of Dataset

Dataset	No of Records (After Pre-Processing)	Attributes	No of Attacks	Classes	Normal	Anomaly
KDD CUP99	148,517	41	4	2	77054	71460

IV. Proposed Methodology

1. Pre-Processing Data:

At stage 1: Sensitive [30] data from the original dataset, like IP addresses and port numbers, must be removed in order to ensure impartial detection and avoid overfitting. The classifier can concentrate on learning from the inherent properties of the packets themselves by omitting this information. By using this strategy, the classifier can find patterns and characteristics that are not dependent on particular socket information. The detection technique can be made more universal by removing IP addresses and port numbers. The classifier can learn to identify

patterns in the packet data that are shared by various hosts rather than depending on the individual network hosts or their socket information. This makes it possible for the classifier to recognise and exclude any hosts that have comparable packet characteristics, independent of their unique socket information.

At stage2: The [27] dataset's multi-class labels, which correspond to the names of attacks, must be encoded into numeric values in order to speed up learning. This encoding enables the classifier to assign a unique class number to each tuple. The binary labels, on the other hand, are already represented as zero-one values and do not call for additional encoding, which is a crucial point to make. The classifier cannot comprehend and learn from the multi-class labels without a transformation of string-based attack names into numerical values. The classifier can distinguish between various sorts of assaults and provide accurate predictions based on the encoded data by giving numerical values to each attack name.

2. Feature Selection:

The suggested method relies heavily on feature selection to reduce the dataset's dimensionality. Based on predetermined evaluation criteria, it entails choosing the original feature set's most pertinent features. By reducing superfluous or duplicate features, redundancy is to be eliminated. Assume we have a set of features called N that has n features total ($f1, f2, f3,...,f_k$). It entails a number of procedures, including the creation of subsets, their assessment, and the use of controls and validation techniques.

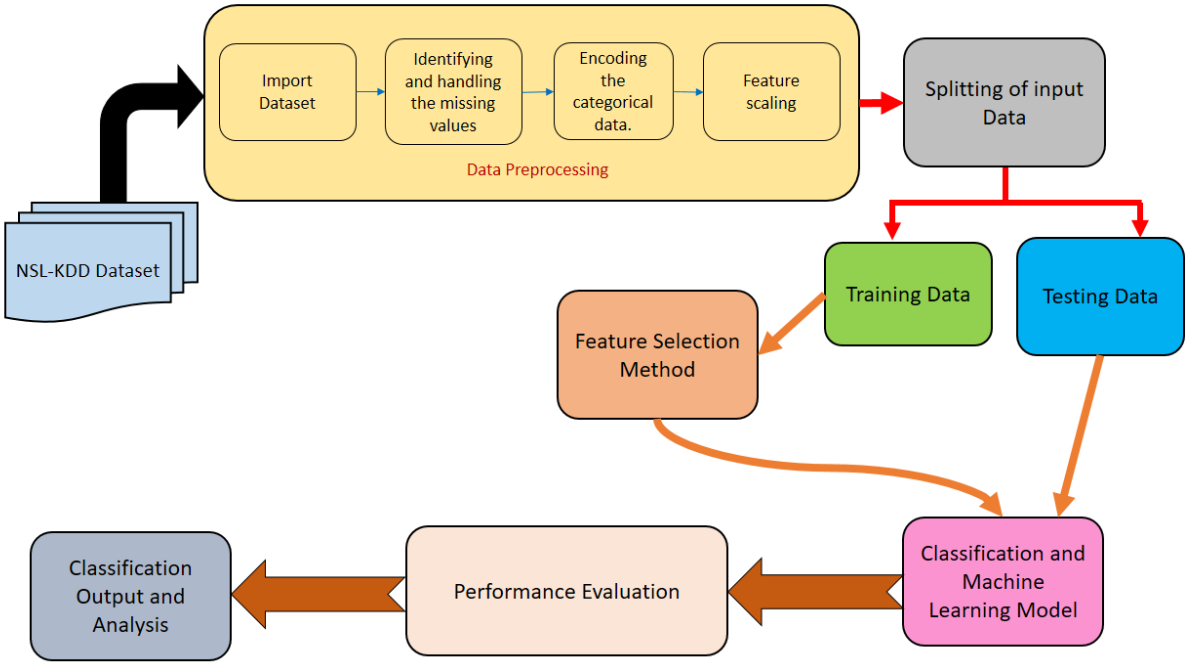

Fig 2: Proposed Intrusion Detection System architecture

The process of dividing the initial feature set into several subsets of features is referred to as "subset generation." To determine their applicability and contribution to the predictive model, these subsets are evaluated based on a set of criteria or assessment measures. The evaluation procedure aids in determining the significance and use of each aspect in relation to the particular situation. To regulate the development and evaluation of the subset during the feature selection process, proper controls and stopping criteria are used. These precautions make sure that the search process ends at the best subset of features. Validation processes are also utilised to assess how well the selected feature subset improved the model's predictive capabilities. Different techniques, such as the traditional method of linear correlation or an alternative technique based on information theory, can be used to determine the correlation between two properties. For each pair of (xt, yt) coordinates, a coefficient is determined using the traditional linear correlation method. The correlation coefficient is the name given to this ratio.

$$r = \frac{\left(\Sigma\left((xt - \bar{x}t)(yt - \bar{y}t)\right)\right)}{\left(\sqrt{\Sigma(xt - \bar{x}t)^2}\sqrt{\Sigma(yt - \bar{y}t)^2}\right)}$$

In this equation, x and y stand for the values of the two variables, x and y for the values of their respective means, and by for the sign of the sum. By dividing the covariance of the variables by the total of their standard deviations, the correlation coefficient is determined.

V. Proposed System

1. K-Nearest Neighbours (KNN):

The k-NN (k-Nearest Neighbours) methodology is the simplest algorithm when it comes to machine learning methods. The training dataset, which serves as the reference data, must be kept in order to create the model. When generating an estimate for a new statistics point, the method searches the training dataset for the "nearest neighbours," or closest data points.

The mathematical summary of the k-NN algorithm is as follows:

• Every data point in the training dataset should be compared to the new data point (X).

• Select the k nearest neighbours based on the estimated distances.

• When completing classification tasks, select the dominant class among the k nearest neighbours and set it as the projected class for the recently generated data point.

•Use the average or weighted mean of the goal values of the k nearest neighbours as the forecast value for the new data point when doing regression activities.

Depending on the type of information and the problem at hand, various proximity metrics, such as the distance from Manhattan or the distance estimated by Euclid, can be employed to establish the separation between two data locations.

2. Decision Tree (DT):

The decision tree is a directed learning technique that forecasts categorised variables for continuously categorised input and exit variables. It is also referred to as the classification tree and regression tree. Human interpretation is uncomplicated and aids decision-making due to its visual portrayal.

In order to provide the most effective data separation, a splitting rule, sometimes referred to as feature collection measure, is a heuristic used to select the best criterion for data partitioning. It makes it easier to locate the tuple breakpoints at a certain node. The attribute selecting measure assigns a rank or score to each feature (or attribute) based on how effectively it can explain the supplied dataset. The attribute with the highest score is chosen as the splitting attribute. For characteristics with continuous values, split points are also selected in order to define each branch.

- The focus of the algorithm is S.
- The algorithm examines each S non-used group attribute and computes the entropy (H) and information gain (IG) that go along with it.

$$Entropy\ E(H) = \sum_{k=1}^{d} - Pj\ log_2 Pj$$

Algoon with it Information gain is represent as:

$$Information\ Gain\ (InGn) = Entrophy\ (before) - \sum_{k=1}^{d} Entrophy(j\backslash k, later)$$

- The property with the largest information gain or lowest entropy is chosen by the algorithm.
- The given S is divided based on the selected attribute.
- The method then applies iteratively to each subassembly, concentrating exclusively on properties that weren't previously selected. Using the Ginni Index:

$$Ginni\ (GI) = 1 - \sum_{j=1}^{jk} (Pl)^2$$

- The algorithm does this by gradually segmenting the data base into different subgroups based on entropy or information quantity, and it continues to do so until a pause condition is satisfied.
- To calculate the Variance:

$$Variance\ (V) = \frac{\sum \sqrt{(X - Xi)}}{n}$$

3. Naive Bayes (NB):

The Naive Bayes automatic learning method, which is based on the Bayes theorem, is used to address a range of categorization problems. To clear up any confusion, we'll go into more detail about the Naive Bayes approach in this article.

Rebuild and Gather Data: assemble a training dataset with input vectors and class labels.

Preliminary probabilities for each class, P(Yi), should be calculated using the occurrences from the formation table. Use the following formula to get a class's preliminary probability.

Determine the residual probabilities: Applying the Bayes theorem, determine the posterior probability of the entry vector P(Y|X) for each distinct class.

$$\text{Prob(Zi|Xj)} = \frac{\big(\text{Prob(Xj|Zi)} * \text{Prob(Zi)}\big)}{\text{Prob(Xj)}}$$

To anticipate the name of the class for an upcoming and unknown vector of entry, the class with the highest possibility of returning must be selected.

4. Support Vector Machine (SVM):

It is well known for its ability to manage large datasets with ease and excels at handling both linear and nonlinear classification problems.SVM has also been used in a range of domains, including Wireless Sensor Networks (WSNs), to address a number of issues, including routing, localization, fault detection, congestion control, and communication issues.

It is necessary to use SVM to locate a hyperplane with the highest margin of separation between the two classes. This hyperplane can be represented as:

$$\text{Wi} * \text{Xi} + \text{C} = 0$$

The SVM Decision function is given as:

$$\text{DF(x)} = \text{sign(Wi} * \text{Xi} + \text{C)}$$

SVM maximises the margin between the classes while minimising the classification error. The optimisation problem is then stated as follows.

$$\text{minimize: } ½\|\text{Xi}\|^2 + \text{D}\Sigma\text{j } \xi\text{j,}$$

$$\text{subject to: Zi(wi} * \text{xj} + \text{c)} \geq 1 - \xi\text{j,}$$

5. Logistic Regression (LR):

A logical function is used in the statistical model of logistic regression to create a binary-dependent variable. The dependent variable in this approach is categorical. The mathematical form of logistic regression is as follows:

$$G(\text{Y} = 1|\text{X}) = \frac{1}{(1 + \text{e}^{-(\beta 0 + \beta 1\text{X}1 + \beta 2\text{X}2 + \ldots + \beta n\text{X}n)})}$$

where G(Y=1|X) is the likelihood that the dependent variable Y will equal 1 in the context of the independent variables X. The coefficients for each independent variable, X1, X2,...,Xn, are 0, 1, 2,..., n. The logistic function (represented by the exponential term) converts a probability value from a linear combination of coefficients and independent variables.

6. Random Forest (RF):

The outputs of various decision trees are combined by the potent ensemble method known as Random Forest to produce a single outcome. Decision trees are the primary learners in Random Forest, and row sampling and column sampling methods are used. The variation is decreased by increasing the base learner population, or vice versa. K is an optional

parameter that can be used for cross-validation. A significant bagging technique is Random Forest, which combines DTs with replacement row sampling, feature bagging through column sampling, and aggregation using techniques like mean/median or majority vote.

7. X G Boost (XGB):

Gradient boosted decision trees are implemented by consecutively building decision trees. The distribution of weights among the various variables in this approach is an essential component. The decision tree results are then obtained using these weights. This weighted technique is used to calculate each decision tree's prediction scores.

$$yi\hat{} = \sum k = 1Kfk\epsilon F$$

In this instance, Gn(X), which stands in for the actual target, is used in place of y1'. It can be stated mathematically as follows:

$$\mathrm{Gn} + 1(\mathrm{X}) = \mathrm{Gn}(\mathrm{X}) + \gamma \mathrm{nH1}(\mathrm{x}, \mathrm{en})$$

$$L1 = (\mathrm{Y} - \mathrm{Gn}(\mathrm{x}))2$$

The loss function used in this model is denoted by the expression (y1 - y1')2, where y1 denotes the actual value and y1' is the model's final predicted value.

VI. Results And Discussion

As intrusion analysis engines in this study, a number of machine learning techniques were used to categorise data as either normal or intrusive. Support Vector Machine (SVM), K-Nearest Neighbour (KNN), Logistic Regression (LR), Naive Bayes (NB), Random Forest (RF), XGBoost(XGB), and Decision Tree (DT) are a few of the methods in this list.While KNN used Euclidean distance and K value for classification, SVM handled linear and non-linear data by locating an ideal hyperplane to separate classes. LR determined target variable probabilities using uniform weights and a maximum iteration value of 100. For independent classification, Naive Bayes used the Gaussian distribution and the Bayes theorem. Using 100 neurons on a hidden layer and a rectified linear unit function, the MLP neural network translated inputs to outputs. While XGBoost employed a different splitting method, DT built tree-like structures using information gain or the Gini index, and RF merged the results of decision trees. 50 trees made up RF, while 100 made up XGBoost. With encouraging outcomes, these algorithms were used for intrusion detection, demonstrating their potency in managing various kinds of data.

Table 3: Model Accuracy, Precision and Recall of Different method

Method	Model Accuracy	Model Precision	Model Recall
KNN	0.91	0.81	0.93
SVM	0.92	0.96	0.91
LR	0.93	0.94	0.89
NB	0.92	0.96	0.92
RF	0.89	0.98	0.96
DT	0.96	0.97	0.98
XGBoost	0.97	0.99	0.98

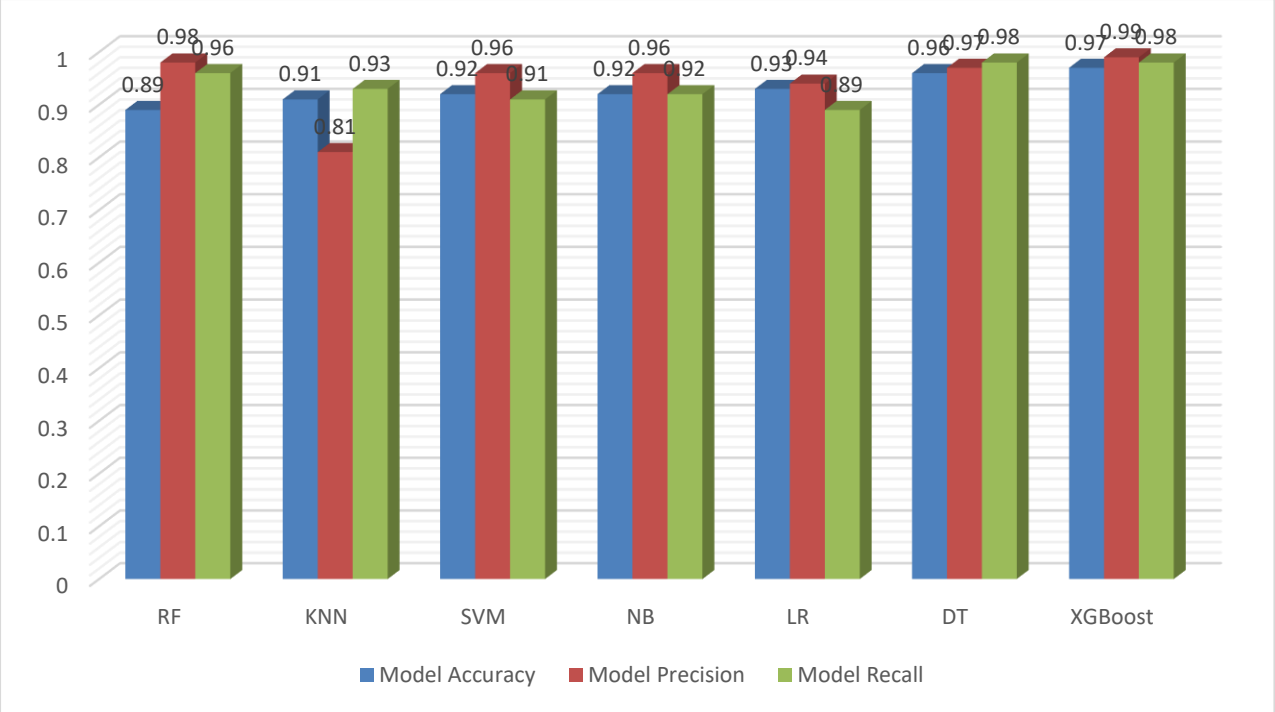

Fig 3: Representation of Model Accuracy, Precision and Recall of Different method

The performance characteristics for various machine learning models used for intrusion detection are shown in the table 3. The models are assessed using their scores for recall, accuracy, and precision.KNN identified 91% of the occurrences correctly, with an accuracy of 0.91. The precision score of 0.81 indicates that 81% of the time, KNN correctly anticipated an intrusion. 93% of the real incursions were correctly detected by KNN, according to the recall score of 0.93.With an accuracy of 0.92, SVM fared marginally better. With a high precision score of 0.96, it showed that SVM correctly identified intrusions 96% of the time. The recall score of 0.91 indicates that 91% of the incursions in the dataset were correctly recognised by SVM.

Table 4: Model Specificity, F1 Score and AUC of Different method

Method	**Model Specificity**	**Model F1 Score**	**Model AUC**
KNN	0.82	0.91	0.93
SVM	0.85	0.93	0.91
LR	0.86	0.94	0.96
NB	0.89	0.94	0.92
RF	0.91	0.95	0.91
DT	0.92	0.98	0.92
XGBoost	0.95	0.97	0.98

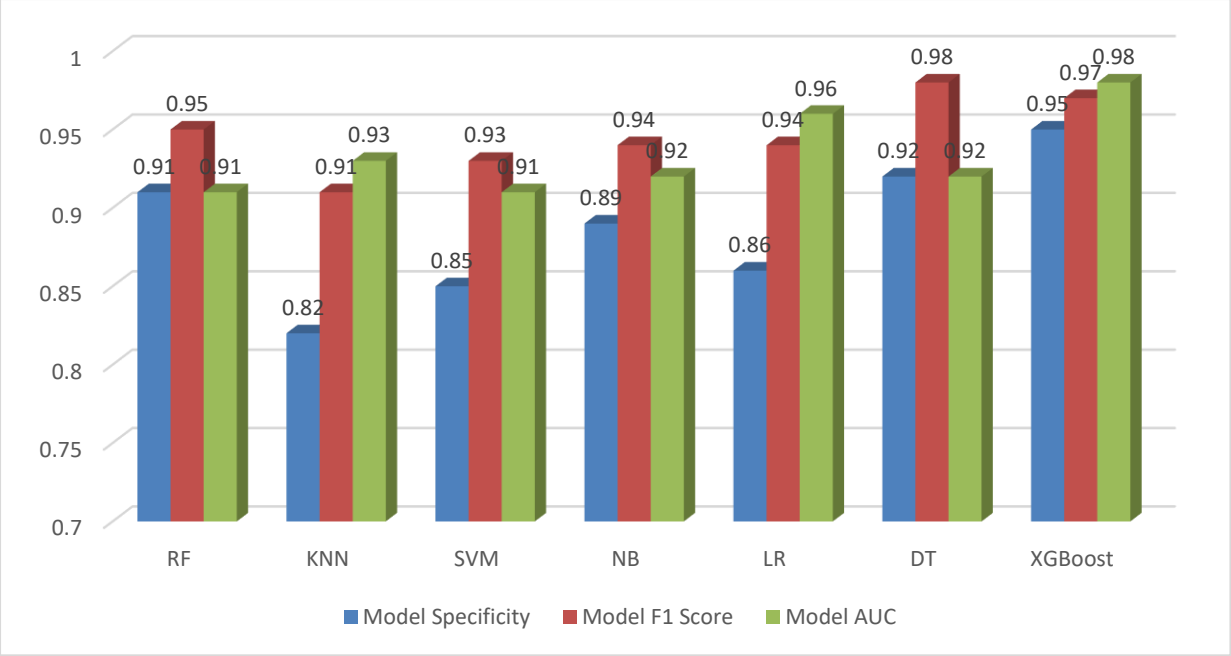

Fig 4: Representation of Model Specificity, F1 Score and AUC of Different method

LR's accuracy score of 0.93 shows that its classifications are generally very accurate. The precision score of 0.94 indicates that LR predicted incursions with an accuracy rate of 94%. 89% of the real incursions were effectively identified by LR, according to the recall score of 0.89.The accuracy of 0.92 obtained by NB further demonstrates its efficacy in intrusion detection. With a precision score of 0.96, NB demonstrated good intrusion prediction accuracy rates. 92% of the actual invasions were successfully captured by NB, as evidenced by the recall score of 0.92. 89% of the cases were properly recognised by RF, which had an accuracy of 0.89. With a precision rating of 0.98, RF has a good rate of success in anticipating intrusions. According to the recall score of 0.96. With an accuracy of 0.96, DT performed extremely well, displaying a high level of categorization accuracy. DT demonstrated a high accuracy rate in both anticipating invasions and capturing actual incursions, with precision and recall scores of 0.97 and 0.98, respectively. With an accuracy score of 0.97, XGBoost achieved the highest mark and proved to be the most successful intrusion detection system.

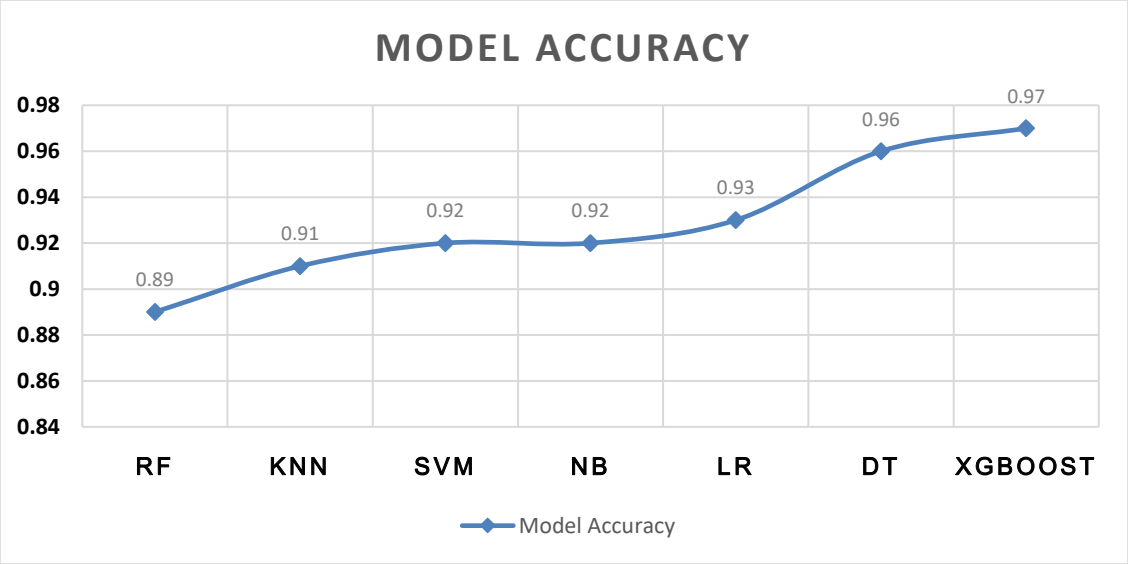

Fig 5: Machine learning method accuracy comparison graph

Random Forest (RF), one of the classification algorithms tested, has the highest accuracy rating (0.89), accurately identifying 89% of the cases in the sample. K-Nearest Neighbor (KNN) did somewhat better, properly classifying 91% of the instances with an accuracy of 0.91. With a score of 0.92, the Support Vector Machine (SVM) showed increased accuracy by accurately identifying 92% of the instances. With an accuracy score of 0.92, the Naive Bayes (NB) algorithm also shown success in detecting intrusions. With an accuracy score of 0.93, Logistic Regression (LR) correctly identified 93% of the occurrences. With an accuracy of 0.96, Decision Tree (DT) surpassed earlier techniques, showing a better level of scenario recognition accuracy. The gradient boosting technique XGBoost, however, received the highest accuracy score, displaying its remarkable performance in this context with an amazing score of 0.97. It showed superior efficacy in intrusion detection.

VII. Conclusion

The objective of this project was to improve the precision and efficacy of intrusion detection systems (IDS) by developing a machine learning framework for intrusion detection using the NSL-KDD dataset. The results of the experiment indicate that machine learning has the ability to greatly improve computer network security. The NSL-KDD dataset was used to evaluate various artificial intelligence classifiers, including SVM, KNN, LR, NB, MLP, RF, ETC, and DT. Recall, precision, and accuracy were some of the measures used to assess the performance of the classifiers. When used on the NSL-KDD dataset, the suggested machine learning technique produced encouraging results, accurately identifying and classifying network intrusions. The capacity of machine learning approaches to accurately identify network intrusions was proven. Notably, integrating cutting-edge methods like DT and XGBoost produced very precise and reliable intrusion detection. Each algorithm demonstrated a different level of performance in these criteria, with some models outperforming others. Notably, the best algorithms for correctly identifying conditions were determined to be DT and XGBoost. These results advance intrusion detection systems by shedding light on how machine learning techniques might be used to network security. The suggested strategy may be improved and expanded upon to handle new network intrusion detection difficulties and to aid in the creation of more advanced and effective IDS products.

References

[1] L. Sun, A. Ho, Z. Xia, J. Chen, M. Zhang, "Development of an Early Warning System for Network Intrusion Detection using Benford's Law Features", in Security and Privacy in Social networks and big data SocialSec. Singapore: Communications in computer and information science, Springer, 2019,vol.1095, pp. 57–73, 2019.

[2] L. Lv, W. Wang, Z. Zhang, X. Liu, " A novel intrusion detection system based on optimal hybrid kernel extreme learning machine", Knowledgebased systems, vol. 195, pp. 1-17, 2020.

[3] M. Darkaie, R. Tavoli, “Providing a method to reduce the false alarm rate in network intrusion detection systems using the multilayer Perceptron technique and backpropagation algorithm”, in 5th Conference on Knowledge-Based Engineering and Innovation, Tehran, Iran, 2019, pp.1-6

[4] W.Wang, Y. Li, X. Wang, J. Liu, X. Zhang, “ Detecting android malicious apps and categorizing benign apps with ensemble of classifiers”, Future Generation Computing System, vol. 78, pp. 987–94, 2018.

[5] H. Alazzam, A. Sharieh, K.E. Sabri, "A feature selection algorithm for intrusion detection system based on pigeon inspired optimizer", Expert systems with applications, vol. 148, pp.1-14, 2020.

[6] M. Safaldin, M. Otair, L. Abualigah, “Improved binary gray wolf optimizer and SVM for intrusion detection system in wireless sensor networks”, Journal of ambient intelligence and humanized computing, pp. 1-18, 2020.

[7] K. Chisholm, C. Yakopcic, M.S.Alam, T.M. Taha, “Multilayer perceptron algorithms for network intrusion detection on portable low power hardware”, in 10th annual computing and communication workshop and conference (CCWC), Las Vegas, USA, 2020, pp. 901- 906.

[8] M.A. Ferrag, L. Maglaras, A. Ahmim, M. Derdour, H. Janicke, "RDTIDS: Rules and decision tree-based intrusion detection

system for internet of things networks", Future internet,vol. 12, no.3, pp.1-14, 2020.

[9] T.B.Prasad, P.S.Prasad, K.P.Kumar, " An intrusion detection system software program using KNN nearest neighbors approach", International journal of science research and innovation engineering (IJSRIE), vol. 1, pp.1-6, 2020.

[10] S.Rajagopal, P.P. Kundapur, K.S.Hareesha, "A stacking ensemble for network intrusion detection using heterogeneous datasets", Secure Communication Networks, pp.1–9, 2020.

[11] F. Salo, A.B.Nassif, A. Essex, "Dimensionality reduction with IG-PCA and ensemble classifier for network intrusion detection", Computer Networks, vol. 148, pp. 164-175, 2019.

[12] H. Sarker, Y.B. Abushark, F. Alsohami, A.I. Khan, "IntruDTree: A machine learning based cyber-security intrusion detection model", Symmetry, vol.12, no.5, pp. 1-15, 2020.

[13] C. Ambikavathi, S.K.Srivatsa, "Predictor selection and attack classification using random forest for intrusion detection", Journal of scientific and industrial research, vol. 79, pp. 365-68, 2020.

[14] Bachar, N. E. Makhfi, O.E. Bannay, "Towards a behavioral network intrusion detection system based on the SVM model", in 2020 1st international conference on innovation research in applied science, engineering and technology (IRASET), Meknes, Morocco, 2020, pp. 1- 7.

[15] D. Wang, G.Xu, "Research on the Detection of Network Intrusion Prevention with SVM Based Optimization Algorithm", Informatica, vol. 44, pp. 269-273, 2020.

[16] S. Thaseen, B. Poorva, P. S. Ushasree, " Network Intrusion Detection using Machine Learning Techniques", in 2020 International Conference on Emerging Trends in Information Technology and Engineering (icETITE), Vellore, India, 2020, pp.1-7.

[17] H. Alazzam, A. Sharieh, K. Sabri, " A feature selection algorithm for intrusion detection system based on Pigeon Inspired Optimizer", Expert Systems with Applications, vol. 148, pp. 1-14, 2020.

[18] N. Kunhare, R. Tiwari and J. Dhar, "Particle swarm optimization and feature selection for intrusion detection system", Sadhana, vol. 45, no. 109, pp.1-14, 2020.

[19] M.Sarnovsky, J. Paralic, "Hierarchical Intrusion Detection Using Machine Learning and Knowledge Model", Symmetry, vol. 12, no. 203, pp.1-14, 2020.

[20] R. A. Ghazy, E. S. M. E. Rabaie, M. I. Dessouky, N. A. E. Fishawy · Fathi E. Abd E. Samie, Feature Selection Ranking and SubsetĞ Based Techniques with Different Classifiers for Intrusion Detection, "Wireless Personal Communications", vol. 111, pp. 375–393, 2020.

[21] S. Waskle, L. Parashar, "Intrusion Detection System Using PCA with Random Forest Approach", in Proceedings of the International Conference on Electronics and Sustainable Communication Systems (ICESC 2020), Coimbatore, India, 2020, pp. 803-808.

[22] M. A. Ferrag, L. Maglaras, A. Ahmim, M. Derdour, H. Janicke, "RDTIDS: Rules and Decision Tree-Based Intrusion Detection System for Internet-of-Things Networks", Future Internet, vol. 12, no. 44, pp. 1- 14, 2020.

[23] P. Nancy, S. Muthurajkumar, S. Ganapathy, S.V.N. Santhosh Kumar, M. Selvi, K. Arputharaj, " Intrusion detection using dynamic feature selection and fuzzy temporal decision tree classification for wireless sensor networks", IET Commununications, vol. 14, no. 5, pp. 888-895, 2020.

[24] Ahmad, M. Basheri, M.J.Iqbal, A.Rahim, "Performance Comparison of Support Vector Machine, Random Forest, and Extreme Learning Machine for Intrusion Detection", IEEE Access, vol.6, pp. 33789–33795, 2018.

[25] W. Li, P. Yi, Y. Wu, L. Pan, J. Li, "A new intrusion detection system based on KNN classification algorithm in wireless sensor network", Journal of Electrical and Computer Engineering, pp. 1–8, 2014.

[26] Y. Yao, W. Yang, F. Gao, Ge. Yu, "Anomaly intrusion detection using hybrid MLP/CNN neural network", in Proceedings of the sixth international conference on intelligent system design and applications (ISDA'06), Jinan, 2006, pp.1095-1102.

[27] S.Peddabachigari, A. Abraham, C. Grosan, J. Thomas, “Modeling intrusion detection system using hybrid intelligent systems”, Journal of Network and Comput Applications, vol. 30, pp. 114–132, 2007.

[28] Q. Zhoua, H. Zhoua, T. Lib, “Cost -sensitive feature selection using random forest: Selecting low-cost subsets of informative features”, Knowledge-based systems, vol. 95, pp. 1-11, 2016.

[29] Andrew Hernandez, Stephen Wright, Yosef Ben-David, Rodrigo Costa, David Botha. Intelligent Decision Making: Applications of Machine Learning in Decision Science. Kuwait Journal of Machine Learning, 2(3). Retrieved from http://kuwaitjournals.com/index.php/kjml/article/view/197

[30] Reddy V, S. ., Madhav, V. ., M, B. ., Krishna A, A. ., A, K. ., Inthiyaz, S. ., & Ahammad, S. H. . (2023). Hybrid Autonomous Vehicle (Aerial and Grounded). International Journal on Recent and Innovation Trends in Computing and Communication, 11(1), 103–109. https://doi.org/10.17762/ijritcc.v11i1.6056

[31] Dhabliya, D., Sharma, R. Cloud computing based mobile devices for distributed computing (2019) International Journal of Control and Automation, 12 (6 Special Issue), pp. 1-4.