Intrusion Detection in the Era of Machine Learning: A Critical Survey of Algorithms and Evaluation Practices

Bhavika
Research Scholar
Department of Computer Engineering
J.C. Bose University of Science and Technology,
YMCA Faridabad, India

Neelam Duhan
Associate Professor
Department of Computer Engineering
J.C. Bose University of Science and Technology,
YMCA Faridabad, India

ABSTRACT

With the growing prominence and sophistication of cyber-attacks, IDS are now indispensable in securing computer networks. Traditional signature-based methods often fail to detect novel threats, prompting the adoption of ML and DL techniques into IDS. This review explores a range of ML algorithms: such as Decision Trees, Random Forest, Support Vector Machines, k-Nearest Neighbors, Naïve Bayes, and Logistic Regression—as well as DL models like Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN). It explains their use in anomaly detection with established datasets like NSL-KDD and UNSW-NB15, and emphasizes importance of data preprocessing, feature selection, and evaluation measures (precision, accuracy, recall, F1-score). The survey emphasizes the strengths as well as constraints of every method, indicating that ensemble & deep learning methods show improved detection accuracy. Finally, it outlines key challenges and proposes future research avenues for developing robust & intelligent IDS solutions.

General Terms

Intrusion Detection, Anomaly Detection.

Keywords

Intrusion Detection System (IDS); Machine Learning (ML); Deep Learning (DL); Anomaly Detection; NSL-KDD; UNSW-NB15.

1. INTRODUCTION

The rapid transformation of digital technology has made computer networks an essential component for modern communication, data exchange, and business operations. However, the increasing connectivity and reliance on these networks have also led to a sharp rise in cyber-attacks, comprising malware, phishing, Distributed Denial-of-Service (DDoS) attacks, and advanced persistent threats. IDSs are pivotal in protection of computer networks because of their capacity to identify and react to anomalous activity. Conventional IDS techniques, such as Signature-based and Rule-based systems, are incapable of identifying novel and emerging threats since they are reliant on predetermined attack patterns [1]. Therefore, ML-based IDS have appeared as a viable method for improving network security by spotting anomalies and potential threats in real time.

While there are some promising results, ML-based IDS is hindered by several challenges such as high false positive rates, adversarial attacks, and scalability limitations. It is essential to conduct a thorough analysis of the latest techniques, their effectiveness, and potential improvements [2]. This survey paper provides a detailed review of the architectures, features, evaluation metrics, and practicality with ML-based intrusion detection methods. Furthermore, it examines the challenges linked with ML-based IDS, like adversarial attacks, high execution costs, and the need for real-time threat detection in expansive networks [3].

1.1 Intrusion Detection

Intrusion takes place when an unwanted user tries to corrupt, modify, or steal information from a host system without proper authorization [4]. Intrusion detection is the process of tracking and analyzing user behavior, system activity, and network traffic to spot signs of unauthorized access or malicious activity within a computer network or system [5]. The primary aim of intrusion detection is to detect security breaches, intrusion attempts, or policy violations in real-time or near real-time, enabling prompt response & mitigation of potential threats. Intrusion detection is essential for preserving the security of computer networks and systems because it enables organizations to identify and address security events quickly, reducing the impact of possible threats and safeguarding crucial data and resources from misconduct or illegal access.

1.2 Architecture of Intrusion Detection

Below is the description of Intrusion detection architecture shown in Fig. 1 and its elements are discussed below: -

Data Collection – Gather the datasets on network traffic (e.g., UNSW-NB15, NSL-KDD, etc.).

Data Pre-processing – After data collection, check for its quality. Data preprocessing modifies and normalizes the dataset, which simplifies feature extraction, data cleaning, and instance selection [6]. This comprises data cleansing, normalization and the use of methods such as SMOTE (Synthetic Minority Over-Sampling Technique) to deal with unbalanced data.

Feature Selection and Engineering -- Identify significant attributes to enhance model accuracy and minimize computation time. Feature selection techniques like Recursive Feature Elimination (RFE) and Principal Component Analysis (PCA) which helps in reducing dimensionality.

Prediction Models for training - Train a variety of ML models, including supervised, unsupervised, and deep learning models. Models like Decision Trees, Random Forest, SVMs, CNNs, & LSTMs will be evaluated for their efficacy. Methods like stacking and boosting can be used to analyze detection rates in ensemble or hybrid models.

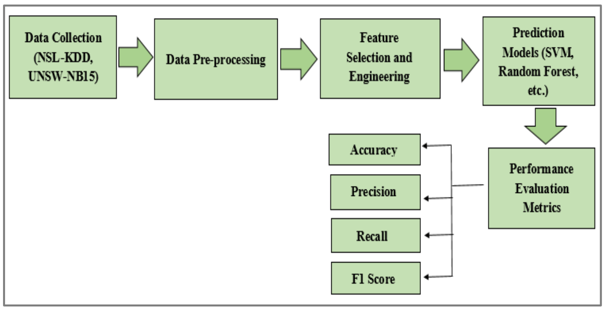
Data Collection (NSL-KDD, UNSW-NB15)
Data Pre-processing
Feature Selection and Engineering
Prediction Models (SVM, Random Forest, etc.)
Performance Evaluation Metrics
Accuracy
Precision
Recall
F1 Score

Figure 1: Architecture of Intrusion Detection

Performance Evaluation Metrics - The following parameters Precision, F1 Score, Recall & Accuracy [7] are employed to evaluate how well the model performs. The brief introduction about these parameters are given below:

Accuracy - It describes how many of the model's predictions were actually came true. The formula for the same is given below in *(1)*:

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

where;

TP: stands for True Positive
FP: stands for False Positive
TN: stands for True Negative
FN: stands for False Negative

Precision – Precision indicates the proportion of favorable forecasts that came true. It focuses on avoiding false alarms (false positives). The formula for the same is given below in *(2):*

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

Recall – It calculates how many true positive cases the model predicted correctly. It focuses on avoiding missed positives (false negatives). High recall ensures fewer missed attacks. The formula for the same is given below in *(3):*

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

F1- Score – It balances precision and recall by combining the two into a single score. It is the inverse of the average values of them. A decent classifier should have the value of 1 [8]. The formula for the same is given below in *(4):*

$$F1\,Score = \frac{2*Precision*Recall}{Precision+Recall} \quad (4)$$

1.3 Various Detection Techniques of IDSs

IDSs employ a number of detection methods to detect possible security risks, which are described as follows:

- ***Signature-based Detection****:* Signature-based detection is employed by matching network traffic or system events with an established database of known attack signatures or patterns [9]. When a known attack or intrusion attempt is traced, an alert is generated by the recognition of matched data.
- ***Anomaly-based Detection****:* Anomaly-based detection entails creating an average network or system behavior and marking those that deviate from the baseline as potential anomalies or suspicious activity. The aim of anomaly detection algorithms is to find out outliers or unusual activities that may indicate a security breach by analyzing traffic patterns, system performance metrics, and user behavior.
- ***Behavior-based Detection****:* Behavior-based detection is concerned with detecting malicious behaviors or activities that go against established security policies or expected norms. This approach involves monitoring user actions, application behaviors, and system interactions to detect unauthorized access, privilege escalation, or other suspicious activities.
- ***Heuristic-based Detection****:* Heuristic-based detection incorporates applying pre-defined rules or heuristics to look for possible security threats or vulnerabilities based on common attack techniques or known weaknesses [9]. These heuristics aim to signal any suspicious activities that may indicate an ongoing attack or exploitation attempt.

In this paper, the analysis of some studies have been performed to figure out the efficient Machine Learning or Deep Learning algorithms that yields better accuracy and other evaluation parameters applied on the Datasets for the Intrusion Detection. This paper is structured as follows: In Section 2, it discusses various ML algorithms employed for IDSs providing the overview of it highlighting their strengths & relevance. Section 3 provides the information of widely used benchmark datasets for intrusion detection, outlining their features, importance and description of relevance in evaluating IDS performance. Section 4 presents a comprehensive literature survey of recent works in the field, summarizing key contributions, methodologies, and performance trends. Section 5 provides a comparative analysis of the surveyed algorithms and datasets, focusing on accuracy, efficiency and applicability to real-time systems. Finally, Section 6 concludes the paper with key insights, observed research gaps, & discussions for future work in the domain of intrusion detection.

2. ALGORITHMS USED

Many researchers have applied Machine Learning techniques for intrusion detection in computer networks. Some of the prevalent techniques are being surveyed in this section.

2.1 Decision Trees

Decision trees are widely used in IDSs due to their ability to be interpreted and can accommodate both categorical and numerical data. C4.5, ID3, and possibly CART algorithms can be used to construct decision trees by using recursive split criteria for creating the feature space. Hence, this leads to the classification of network traffic as either malicious or benign. This algorithm helps to capture hierarchical relationships in network traffic and also it facilitates fast inference and explainability for real-time IDS.

2.2. Random Forest

Random Forest is an ensemble learning technique that combines the predictions of numerous decision trees to enhance the accuracy of classification and reduce the chances of overfitting [10]. This approach is frequently employed to tackle classification problems [11]. The final classification outcome is obtained by building multiple decision trees and collecting the results of their predictions. Random Forests are renowned for their consistency as well as their ability to handle high-dimensional data [12].

2.3 Support Vector Machines (SVM)

SVM is a supervised learning algorithm and non-parametric classifier that uses a linear vector to divide classes for regression and classification [13, 21]. SVMs are efficient for binary classification problems and can be used with IDS for discriminating normal from malicious network traffic. SVMs are also useful for detecting anomalies in network traffic. SVMs aim to find a hyperplane that efficiently separates the data points belonging to each class. By employing kernel functions, they can solve linear as well as non-linear separations [14].

2.4 Neural Networks

In neural networks, deep learning models such as convolutional neural networks (CNNs) and recurrent neural network (RNNs), have shown a lot of promise in the field of IDS. CNNs are suitable for extracting spatial relationships in network traffic data, while RNNs are better suited for processing sequential data, making them suitable for analyzing sequences of network packets or log entries. Both types of neural networks can be employed together [15].

2.5 Logistic Regression

In recent years, deep learning has seen rapid advancements. However, it requires extensive datasets and high-performance computing hardware. For simpler binary classification problems, traditional machine learning techniques remain valuable. Among these, logistic regression is widely used as an effective method for binary classification and predictive analysis [16].

Logistic regression is a standard statistical classification method, particularly applied to binary-classification problems [22]. It works by assigning data points to one of two categories. Each input feature is multiplied by its corresponding weight, and the weighted sum is computed as described in (5), yielding a z-value. This z-value is then processed through the sigmoid function to generate an output. If the output is below 0.5, the data point is classified as 0, while values above 0.5 are classified as 1. Through this process, logistic regression effectively models the relationship between input features and class labels, enabling accurate predictions.

$$Z = W_0X_0 + W_1X_1 + W_2X_2 + \cdots + W_nX_n \qquad (5)$$

where,

- Z : The weighted sum or linear combination of the input features.
- W_i (for i = 0,1, 2,...,n) : The weight coefficient associated with the input feature X_i. These weights determine the influence of each feature on the output prediction and are learned during model training.
- X_i (for i = 0,1,2,...,n) : The input feature values for a given instance.
  - X_0 is typically set to 1 and is used to represent the bias term.
  - The remaining X_1 to X_n are the actual feature values.
- W_0X_0 : Represents the bias term in the model. Since X_0 = 1, this simplifies to just W_0, which allows the model to shift the decision boundary.

2.6 k-Nearest Neighbor

The k-Nearest Neighbor (k–NN) algorithm is one of the most commonly used, simple and non-parametric algorithms for classification [17, 18]. It classifies data points based on the majority class among their k closest neighbors, determined by a distance metric such as Euclidean distance. k-NN is known for its effectiveness and ease of implementation but can be computationally intensive for large datasets. In intrusion detection, k-NN offers competitive accuracy and serves as a strong baseline model, though it is sensitive to irrelevant features and data scaling.

2.7 Naïve Bayes

Naive Bayes is a probability-based classifier based on Bayes' Theorem, assuming high statistical independence among features. Despite this simplification, it performs remarkably well in various applications, particularly in text classification and spam detection. The algorithm estimates the posterior probability of every class for the input features and returns the highest-probability class [19]. Naive Bayes is highly efficient, scalable, and works well with high-dimensional data. Intrusion detection systems benefit from its fast and accurate performance, making it suitable for real-time deployment, but feature independence assumptions may be significantly violated, leading to deterioration in performance. The Naïve Bayes algorithm is based on the Bayes formula:

$$P(A|B) = \frac{P(B|A).P(A)}{P(B)} \qquad (6)$$

This formula states that the probability of variable B being A if class A is known and the likelihood of variables B having A presence based on knowledge is determined by this function. The probability class that follows this formula is the decision class [20].

3. DATASET USED

The NSL-KDD dataset is an improved version of the KDD'99 dataset, which was the first widely adopted benchmark for IDS research. It addresses prime issues in KDD'99 such as duplicate records and imbalance, making evaluations more reliable. It contains 41 features and labels that represent normal and various attack types (e.g., DoS, Probe, U2R, R2L), which are suitable for both binary and multi-class classification tasks. The simplicity and structure of NSL-KDD make it ideal for evaluating classical machine learning algorithms like SVM, Decision Trees, and Naive Bayes. Although it's outdated in terms of modern network traffic, it remains useful for comparative studies and initial prototyping of IDS algorithms. Table 1 below represents the basic information of the NSL-KDD dataset.

Table 1: Description of NSL-KDD Dataset

Attributes	NSL-KDD Dataset
Total Records	148,517
Training Data	125,973 (KDDTrain+)
Testing Data	22,544 (KDDTest+)
Number of Features	41
Types of Attacks	4 Main Categories (22 specific attacks)
DoS (Denial of Service)	Smurf, Neptune, Teardrop, Pod, Land, Back, etc.
Probe (Surveillance & Probing)	Satan, Ipsweep, Nmap, Portsweep, etc.
R2L (Remote to Local)	Guess_Password, Warezclient, Warezmaster, Imap, etc.
U2R (User to Root)	Buffer_overflow, Loadmodule, Rootkit, Perl, etc.

The cyber security research team of the Australian Cyber

Security Centre came up with a new dataset, UNSW-NB15, in 2015 to meet with the problems encountered by the KDDCup 99 & NSLKDD datasets. This dataset captures more realistic and recent network behaviors, using real-time emulated environments with tools like IXIA PerfectStorm tool. The IXIA traffic generation tool used two servers where one server generated normal events and the other server generated malicious events in the network. It includes nine different modern attack categories (e.g., Fuzzers, Backdoors, Exploits, Worms, Shellcode), providing a broader testing ground for IDS performance. It comprises 49 features including basic and behavioral characteristics, which better reflect current cybersecurity challenges. The complexity and richness of UNSW-NB15 make it suitable for testing advanced machine learning and deep learning models. Table 2 below represents the basic information of the UNSW-NB15 dataset.

Table 2: Description of UNSW-NB15 Dataset

Attributes	UNSW-NB15 Dataset
Total Records	257,673
Training Data	175,341
Testing Data	82,332
Number of Features	49
Types of Attacks	9 Attack Categories
Fuzzers	Random input testing to find vulnerabilities
Analysis	Scanning and traffic analysis attacks
Backdoor	Unauthorized remote access attacks
DoS (Denial of Service)	Service-disrupting attack
Exploits	Targeting system vulnerability
Generic	Cryptographic attacks against ciphers
Reconnaissance	Information-gathering and scanning
Shellcode	Injecting malicious shellcode into memory
Worms	Self-replicating malicious programs

4. COMPREHENSIVE LITERATURE SURVEY

G. Yedukondalu et al. [21] discusses the development of an IDS using ML techniques to improve cybersecurity. Primary objective is to apply and compare Support Vector Machine (SVM) & Artificial Neural Networks (ANN) algorithms for intrusion detection. The study utilizes the NSL-KDD dataset, a widely used benchmark for IDS research.

The research employs Correlation-Based & Chi-Squared feature selection techniques to preprocesses the dataset by removing redundant data. The dataset is then trained and tested using SVM and ANN models to evaluate their efficiency. The results indicate that the ANN model significantly outperforms SVM, attaining an accuracy of 97% as compared to 48% for SVM. It highlights importance of machine learning in cybersecurity, emphasizing that ANN-based IDS provides superior accuracy in detecting unauthorized access and malicious activities. The study concludes that while ANN performs well, future research should explore alternative models that balance high accuracy with computational efficiency for real-time applications.

A. Mohamed et al. [23] explores the use of machine learning (ML) to improve the effectiveness of intrusion detection systems (IDS). Traditional rule-based IDS often fail to detect evolving cyber threats, which drives the shift toward intelligent, data-driven methods. The study evaluates several ML models—decision trees, support vector machines (SVM), random forests, and deep learning—on the UNSW-NB15 dataset. The process includes data preprocessing, feature engineering, training, and model evaluation using metrics such as accuracy, precision, recall, and F1-score. Results show that neural networks perform best (98.3%), followed by random forests (97.6%), decision trees (95.6%), and SVMs (95.2%). The study emphasizes the importance of feature selection and highlights the advantages of ensemble learning in enhancing IDS robustness. It concludes that ML-based IDS significantly improve detection capabilities and recommends future work on real-time implementation, efficiency improvement, and resilience against adversarial threats.

A. Y. Kalayci & U. Hacizade [24] presents an anomaly-based IDS utilizing various ML techniques to enhance network security. The study evaluates multiple classifiers, including Logistic Regression, K-Nearest Neighbor (KNN), Support Vector Machines (SVM), Naive Bayes, Decision Tree, and Random Forest, using the UNSW-NB15 dataset. The dataset includes both normal and attack network traffic, making it suitable for intrusion detection research.

The methodology involves data preprocessing, feature transformation, and normalization before training machine learning models. The Random Forest and Decision Tree classifiers achieved the highest accuracy (99.99%), while Logistic Regression performed the worst, with an accuracy of 79.04%.

The findings emphasize that ensemble techniques like Random Forest outperform traditional classifiers in intrusion detection owing to their capacity to handle intricate intrusion sequences effectively. The study concludes that while machine learning-based IDS can significantly enhance cybersecurity, challenges such as dataset accuracy, feature selection, and false alarm rates remain critical areas for future research.

F. Guo et al. [25] presents a machine learning-driven approach to enhance network intrusion detection. Addressing the limitations of traditional IDS—such as poor accuracy and limited real-time capability—the authors combine support vector machines (SVM), deep learning (DL), and reinforcement learning to design a multi-module IDS framework. This includes components for data collection, feature extraction, model training, real-time detection, and logging. The system utilizes preprocessing methods and feature optimization techniques like chi-square tests and principal component analysis (PCA), which lead to improved detection accuracy. Comparative analysis of various classifiers shows that neural networks perform best, achieving a 92% accuracy rate. Hyperparameter tuning through grid and random search further enhances model performance. Real-time tests demonstrate an average detection time of 50 milliseconds. The research concludes that ML-based IDS offers significant improvements in speed and adaptability, with future work focusing on algorithmic refinement and real-time efficiency. Table 3 described the consolidated overview of the above analysis is on Page no. 7.

5. COMPARATIVE ANALYSIS

The ML algorithms provides an edge when integrated with the IDS. The performance evaluation comparison of the studies has been described in Table 4 below:

Table 4: Comparison of Performance Evaluation Metrics

Papers	ML / DL Algorithms	Performance Evaluation Metrics (in %)			
		Accuracy	Precision	Recall	F1 Score
[21]	SVM ANN	48 97	NIL NIL	NIL NIL	NIL NIL
[23]	Decision Tree	95.6	92.3	81.2	78.9
	SVM	95.2	95.6	87.5	82.5
	Random Forest	97.6	93.5	85.6	80.6
	CNN	98.3	97.8	89.6	83.6
[24]	Decision Tree	99.99	99.99	99.99	99.99
	SVM	93.76	86.63	99.4	92.58
	Random Forest	99.99	100	99.99	99.99
	KNN	99.41	98.88	99.81	99.34
	Logistic Regression	79.04	54.19	98.5	69.91
	Naïve Bayes	95.13	89.16	99.99	94.26
[25]	Decision Tree	85	83	80	82
	SVM	88	86	83	85
	Random Forest	90	88	85	87
	Neural Network	92	91	88	90

G. Yedukondalu et al. [21], the ANN produce better accuracy than SVM. The representation is illustrated in Fig. 2 below:

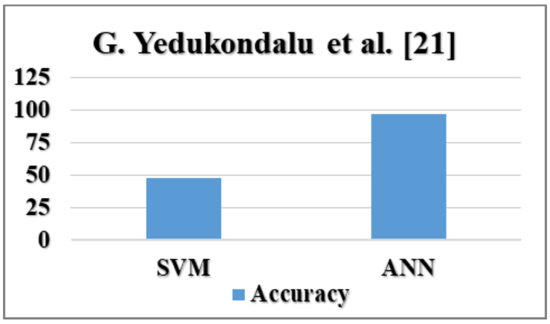

Figure 2: Output graph of Accuracy of ML of [21]

This shows that ANN outperforms in producing the accuracy on NSL-KDD dataset.

In the study of A. Mohamed et al. [23], there are several algorithms employed for Intrusion Detection on the benchmark dataset. The outputs of their performance evaluations are illustrated below in Fig. 3.

In this study, CNN performed well among the other algorithms in every performance parameter. This shows that the deep learning methods are better at finding the intrusions than traditional machine learning methods. Now here comes another study of A. Y. Kalayci & U. Hacizade [24] in which numerous algorithms are used to find out which method works well in obtaining the Intrusion Detection on the benchmark dataset.

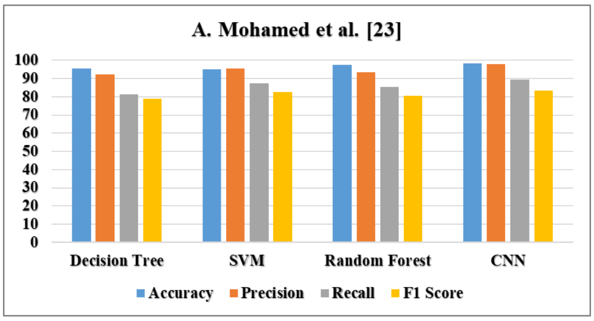

Figure 3: Output graph of Performance metrics of ML of [23]

The output of every method with their performance evaluation metrics is described below in Fig. 4:

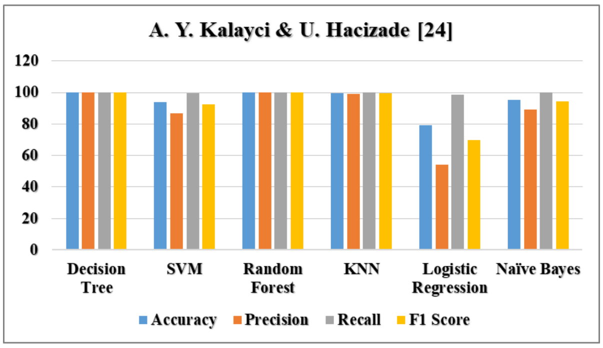

Figure 4: Output graph of Performance metrics of ML of [24]

In [24], Decision Tree and Random Forest generated outstanding result in performance metrics. Then after that goes KNN which also produced promising result. The least scorer among all the algorithms is Logistic Regression. This shows that these two algorithms are working well for performing the anomaly intrusion detection. The last study of F. Guo et al. is being performed on the real-time network traffic. In [25], four methods are employed and their respective performance metrics are shown below in Fig. 5:

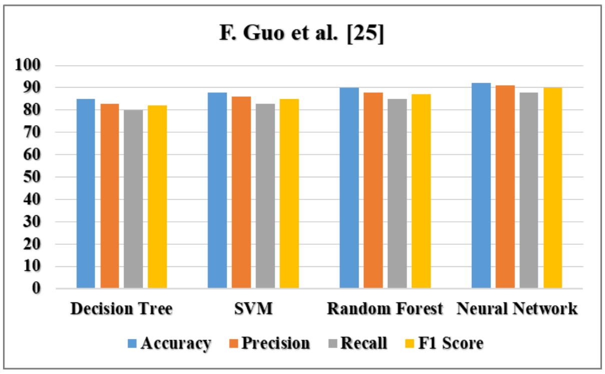

Figure 5: Output graph of Performance metrics of ML of [25]

After looking upon the figure, it is clearly visible that the Neural Network generated promising outputs than the other methods. It generates better precision, accuracy, recall and F1 score.

The method that performed not-so-good is Decision Tree. It generated good result but when compared with other methods, it falls short. Now, looking forward for analysis of methods that are in common among three or four studies. Table 5 below represents the comparison of performance evaluation metrics of Decision Tree employed in the above studies.

Table 5: Comparison of Perf. Metrics of Decision Tree (in %)

Decision Tree	Accuracy	Precision	Recall	F1 Score
[23]	95.6	92.3	81.2	78.9
[24]	99.99	99.99	99.99	99.99
[25]	85	83	80	82

After looking the above data, it can be seen that the respective method works well in the benchmark dataset of [24] (with better accuracy, precision, recall and F1 score than other studies). And the representation of it is shown in Fig. 6 below:

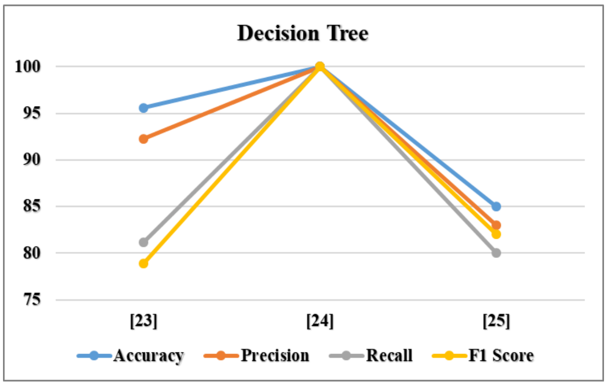

Figure 6: Output graph of Performance Metrics of Decision Tree

Now moving ahead is the below Table 6 represents the performance evaluation metrics comparison of Random Forest that is used in the above studies.

Table 6: Comparison of Perf. Metrics of Random Forest (in %)

Random Forest	Accuracy	Precision	Recall	F1 Score
[23]	97.6	93.5	85.6	80.6
[24]	99.99	100	99.99	99.99
[25]	90	88	85	87

After reflecting on the above data, it is drawn out that the above mentioned method works well in the benchmark dataset of [24]. And the representation of it is shown below in Fig. 7:

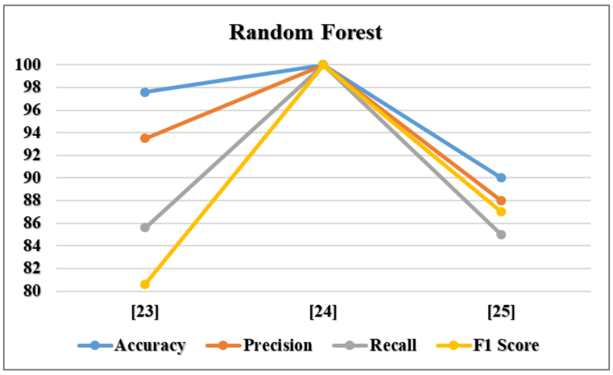

Figure 7: Output graph of Performance Metrics of Random Forest

Now, below is the comparison table of SVM that is used in the above studies:

Table 7: Comparison of Perf. Metrics of SVM (in %)

SVM	Accuracy	Precision	Recall	F1 Score
[21]	48	NIL	NIL	NIL
[23]	95.2	95.6	87.5	82.5
[24]	93.76	86.63	99.4	92.58
[25]	88	86	83	85

In the above Table 7, it can be seen that the accuracy and precision of [23] is more than the other studies i.e. 95.3 & 95.6. This may describe that the method is performing well on majority (benign or frequent attack) classes and SVM is not misclassifying benign traffic or common attacks, which increases precision.

But the recall and F1 score of [24] is more than the other studies i.e. 99.4 and 92.58. This may describe that the method is better at detecting rare or unknown attacks, even at the cost of more false positives (lower precision). As the task likely focuses on detecting rare intrusions, and SVM is more aggressive in labeling anomalies, which improves recall.

Also higher recall is particularly valued in security, where missing an attack is more critical than a false alert. The representation is shown in Fig. 8:

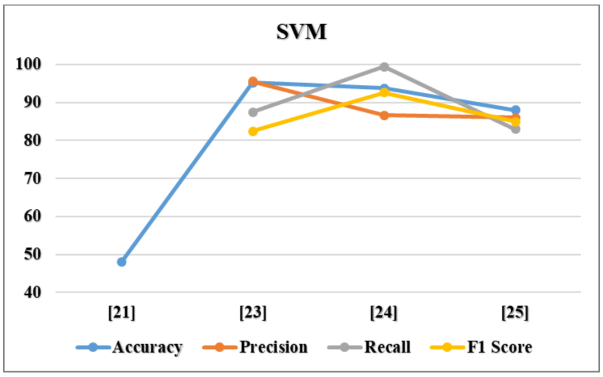

Figure 8: Output graph of Performance Metrics of SVM

6. CONCLUSION

This study is being done to address the efficient ML methods that shall be taken up to perform intrusion detection in the CN. The various studies have been analyzed and the comprehensive survey is done. The methods are being assessed on the basis of their performance evaluation parameters such as precision, accuracy, recall and F1 score. These parameters assist in better understanding of the practical application of the ML methods in intrusion detection area. The study also found that Random Forests performed well, particularly in terms of recall, implying that they are skilled at detecting occurrences of positive class. These results shed insight on the relative benefits of various ML algorithms, allowing ID system designers to make wiser decisions. This study has the potential to make significant advances to network security using DL approach. Deep Learning models may help organizations discover and stop undesirable activities in real time. This could improve network security, minimize the severity of intrusions, and protect sensitive information. Future scope may include designing Real-time IDS that balances accuracy with low latency for deployment in high-speed networks and Adversarial resilience, focusing on models that can withstand evasion techniques and poisoned data.

Table 3: Comprehensive analysis of research paper

Research Paper	Study Title	ML/ DL Algorithms used	Dataset	Key Findings	Advantages	Limitations
G. Yedukondalu et al. [21]	Intrusion Detection System Framework using Machine Learning	SVM, ANN	NSL-KDD	ANN algorithm is working efficiently on this dataset.	The algorithms have employed Correlation-Based and Chi-Squared Based feature selection algorithms to decrease the dataset by removing the unnecessary data	Due to absence of confusion matrix, unable to find the other parameters.
A. Mohamed et al. [23]	Machine Learning-Based Intrusion Detection Systems for Enhancing Cyber Security	Decision Tree, SVM, Random Forest, CNN	UNSW-NB15	CNN outperforms among the mentioned algorithms.	The algorithms are performing effectively on the desired Dataset.	--------
A. Y. Kalayci & U. Hacizade [24]	Anomaly-Based Intrusion Detection System Using Machine Learning Methods	Logistic Regression, Random Forest, KNN, SVM, Naïve Bayes, Decision Tree	UNSW-NB15	Random Forest produces output accurately and precisely among the given dataset.	The ensemble approach like Random Forest outperform traditional classifiers.	In this, Random Forest and Decision tree produces the nearly ideal result due to over-fitting factor.
F. Guo et al. [25]	Information Security Network Intrusion Detection System based on Machine Learning	Decision Tree, Random Forest, SVM, Neural Network	Not Mentioned	Neural Network work well in the network intrusion detection.	It focuses on feature selection optimization using chi-square test and PCA that significantly improved accuracy .	--------

7. REFERENCES

[1] S. Axelsson, "The base-rate fallacy and its implications for the difficulty of intrusion detection," ACM Trans. Inf. Syst. Secur., vol. 3, no. 3, pp. 186–205, Aug. 2000. doi: 10.1145/357830.357849.

[2] A. L. Buczak and E. Guven, "A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection," in IEEE Communications Surveys & Tutorials, vol. 18, no. 2, pp. 1153-1176, Secondquarter 2016, doi: 10.1109/COMST.2015.2494502.

[3] M. Gharib, R. Wehbe, A. Habrard, and C. Leroux, "Adversarial machine learning in network intrusion detection: Taxonomy, challenges, and future trends," IEEE Access, vol. 8, pp. 71366–71384, 2020. doi: 10.1109/ACCESS.2020.2987075.

[4] "Intrusion Detection: A Survey," White Paper, Chapter 2, 2002.

[5] D. E. Denning, "An Intrusion-Detection Model," IEEE Transactions on Software Engineering, vol. SE-13, no. 2, pp. 222–232, Feb. 1987. doi: 10.1109/TSE.1987.232894.

[6] Murat oguz, Ihsan Omur buck, “A Behavior Based Intrusion Detection System Using Machine Learning Algorithms,” International Journal of Artificial Intelligence and Expert Systems,vol. 7, pp.22-26, 2016.

[7] P. Venkateswari, E. Jebitha Steffy, N. Muthukumaran, 'License Plate cognizance by Ocular Character Perception', International Research Journal of Engineering and Technology, Vol. 5, No. 2, pp. 536-542, February 2018.

[8] Al Mehedi Hasan, Mohammed Nasser, Shamim Ahmad, “Intrusion Detection System using Feature selection and Machine Learning methods,” Proceedings of 1996 IEEE Symposium on Computer Security and Privacy, pp.120-128, 1996.

[9] R. Bace and P. Mell, "Intrusion Detection Systems," NIST Special Publication 800-31, National Institute of Standards and Technology, 2001.

[10] L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001. doi: 10.1023/A:1010933404324.

[11] Reis, I., Baron, D., & Shahaf, S. Probabilistic random forest: A machine learning algorithm for noisy data sets. The Astronomical Journal, 2018, vol. 157, no. 1, pp., 16.

[12] M. A. Khan, N. Javaid, A. Majid, M. Imran, and M. Alnuem, “Dual sink efficient balanced energy technique for underwater acoustic sensor networks,” Proc. - IEEE 30th Int. Conf. Adv. Inf. Netw. Appl. Work. WAINA 2016, pp. 551–556, 2016, doi: 10.1109/WAINA.2016.156.

[13] H. P. Singh and M. Sharma, "Intrusion detection using feature selection and machine learning algorithm with misuse detection," Int. J. Comput. Sci. Inf. Technol., vol. 8, no. 1, pp. 145–152, 2016.

[14] K. Wang and S. J. Stolfo, "Anomalous payload-based network intrusion detection," in Recent Advances in Intrusion Detection, Springer, 2004, pp. 203–222.

[15] A. Kannappan and R. M. Bommi, “Energy-Efficient Routing using the Hybrid Bilevel- Litechenbery-Optimization Algorithm in Comparison with Ant-colony Optimization,” ICDCS 2022 - 2022 6th Int. Conf.

Devices, Circuits Syst., no. April, pp. 464– 466, 2022, doi: 10.1109/ICDCS54290.2022.9780826.

[16] Rongheng, S. Applied Mathematical Statistics (3rd Edition), CA: Science Press, 2014.

[17] Bishop, C.M. Neural networks for pattern recognition. England Oxford University, 1995.

[18] Manocha, S., and Girolami, M.A. An empirical analysis of the probabilistic K-nearest neighbour classifier. Pattern Recognition Letters, 2007, vol.28, pp.1818–1824.

[19] T. Mitchell, Machine Learning, McGraw-Hill, 1997.

[20] Solmaz, R., Günay, M., Alkan, A. Use of naive bayes classifier in the diagnosis of functional thyroid disease. Academic Informatics Conference, 2014, Mersin, Türkiye, pp. 891-897.

[21] G. Yedukondalu, G. H. Bindu, J. Pavan, G. Venkatesh, and A. Sai Teja, "Intrusion Detection System Framework Using Machine Learning," in Proc. 2021 3rd Int. Conf. Inventive Res. Comput. Appl. (ICIRCA), Coimbatore, India, 2021, pp. 437–442. doi: 10.1109/ICIRCA51532.2021.9544522.

[22] H. L. Gururaj, F. Flammini, V. R. Ravikumar, and N. S. Prema, Recent Trends in Healthcare Innovation. Boca Raton, FL, USA: CRC Press, 2025.

[23] A. Mohamed, J. Heilala and N. S. Madonsela, "Machine Learning-Based Intrusion Detection Systems for Enhancing Cybersecurity," 2023 Second International Conference On Smart Technologies For Smart Nation (SmartTechCon), Singapore, Singapore, 2023, pp. 366-370, doi: 10.1109/SmartTechCon57526.2023.10391626.

[24] A. Y. Kalayci and U. Hacizade, "Anomaly-Based Intrusion Detection System Design Using Machine Learning Methods," 2024 XXXIII International Scientific Conference Electronics (ET), Sozopol, Bulgaria, 2024, pp. 1-6, doi: 10.1109/ET63133.2024.10721523.

[25] F. Guo, H. Jiao, X. Zhang, Y. Zhou and H. Feng, "Information Security Network Intrusion Detection System Based on Machine Learning," 2024 International Conference on Data Science and Network Security (ICDSNS), Tiptur, India, 2024, pp. 01-04, doi: 10.1109/ICDSNS62112.2024.10691041.