Empirical Analysis of Conventional ML Techniques

Geeta Kocher[1] and Gulshan Kumar[2]

[1]*Research Scholar*
Department of Computational Sciences, MRSPTU, Bathinda (Punjab), India
[2]*Associate Professor*
Department of Computer Applications, SBSSU, Ferozepur, (Punjab), India

Abstract. With the rapid growth of internet technology, the prevalence of threats has been increasing at an exponential rate. To mitigate the effect of these threats, researchers have suggested numerous intrusion detection solutions. However, many machine learning classifiers in the literature are trained on outdated datasets, which constrains their detection accuracy. Therefore, it is essential to train machine learning classifiers on more up-to-date datasets. In this paper, an empirical evaluation of several conventional machine learning classifiers, including Naïve Bayes, Stochastic Gradient Descent, Logistic Regression, k-Nearest Neighbours, and, Decision Trees using benchmark datasets, NSL-KDD, UNSW-NB15, and CICIDS. Performance is analyzed in terms of recall, precision, and FPR for a comprehensive comparison. The results emphasize the relative strengths and weaknesses of each model, providing insights into their fitness for different types of prediction tasks. Subsequently, the paper presents the results and discussions of the experiments, culminating in a summary of the key findings.

KEYWORDS. Machine Learning, Logistic Regression, Intrusion Detection System, False Positive Rate, Detection Rate

1. Introduction

The rapid growth of Internet technologies has led to a major increase in cybersecurity threats, making effective intrusion detection systems (IDS) essential to protecting sensitive information. Intrusion detection, which involves identifying unauthorized access or anomalous behavior in computer systems, has become a critical research area in the field of cybersecurity. Traditional methods of intrusion detection rely heavily on pattern-based techniques or signature matching, but these approaches often fail to detect novel attacks and adapt to evolving threat landscapes.

With the rise of machine learning (ML), a variety of data-driven methods have been suggested to enhance the accuracy and adaptability of IDS. Conventional ML classifiers, such as Naïve Bayes (NB), Stochastic Gradient Descent (SGD), Logistic Regression (LR), k-Nearest Neighbors (KNN), and Decision Trees (DT), have been widely employed in the detection of cyber-attacks due to their ability to learn from data and make predictions. Despite their success in many domains, the performance of these classifiers often depends on factors like the quality of the training datasets, the selection of features, and the selection of evaluation metrics [1].

To assess and compare the effectiveness of these conventional ML classifiers, it is important to conduct empirical studies using widely recognized benchmark datasets. In this research, the performance of several traditional ML algorithms is evaluated on three prominent datasets: NSL-KDD, UNSW-NB15, and CICIDS. These datasets are commonly used in intrusion detection studies, as they contain labeled attack data that helps train and evaluate the models [2]. By applying a standard set of performance metrics—including Recall, Precision and False Positive Rate (FPR)—the aim is to provide a comprehensive analysis of the strengths and weakness of these ML classifiers in the context of intrusion detection.

The results of this study will be incorporated into the ongoing discourse about the suitability of conventional ML models for modern IDS, highlighting areas for improvement and potential directions for future research in cybersecurity.

2. Related Work

This section provides an overview of the existing literature on ML classifiers, with a focus on their application in intrusion detection. The intent of this section is to give an outline of the significant research conducted in this domain. A thorough examination of the literature reveals that researchers have dedicated considerable effort to the growth and evaluation of ML classifiers, and several key contributions in this area are summarized below:

Using the NSL-KDD dataset, Kim et al. (2014) [3] presented a hybrid ID approach that combines anomaly detection and misuse detection. The findings showed improved detection rate (DR), reduced temporal complexity, and lower FPR. But the method's time efficiency might be improved, thus future studies will focus on improving the C4.5 DT algorithm.

Belavagi and Muniyal (2016) [4] designed a Network IDS using supervised ML classifiers like NB, LR, Support Vector Machines(SVM), and random forest (RF). To validate the performance, these classifiers were tested on NSL-KDD datasets. It was reported that the RF outperformed other classifiers with an accuracy of 99%. This work was tested on older dataset and limited to binary classification.

A hybrid approach was proposed by Guo et al. (2016) [5] to attain a high DR with a low FPR. The system was built using a two-tier hybrid methodology that consists of a misuse detection component in addition to two anomaly detection components. The experimental findings demonstrated that, using the KDD'99 dataset, this method could successfully identify network abnormalities with a low FPR.

Ashfaq et al. (2017) [6] proposed a semi-supervised learning (SSL) approach that incorporates novel uncertainties to improve the performance of the classifier. This approach utilizes both labeled and unlabeled samples in conjunction with a supervised learning algorithm. The NSL-KDD dataset was used to evaluate the model. However, a key limitation of this study was that the performance of the model was only evaluated for the binary classification task.

In 2019, Çavusoglu [7] used a variety of machine learning techniques. The NSL-KDD dataset was employed for both testing and training. The suggested approach demonstrated low FPR and good accuracy across all attack types. But the limitation was that older dataset was used for evaluation.

Table 1 depicts the comparison of above mentioned studies.

Table 1. Summary of the Literature Review

Study	Dataset	Methodology	Key Findings	Limitations
Kim et al. (2014)	NSL-KDD	Hybrid ID combining anomaly and misuse detection.	Improved detection rate (DR), reduced temporal complexity, lower FPR.	Time efficiency could be improved; focused on improving C4.5 decision tree.
Belavagi & Muniyal (2016)	NSL-KDD	Supervised ML classifiers: LR, SVM, NB, RF.	RF achieved highest accuracy of 99%.	Tested on older dataset; limited to binary classification.
Guo et al. (2016)	KDD'99	Two-tier hybrid methodology (misuse + anomaly detection).	High DR and low FPR in detecting network abnormalities.	Older dataset used; limited exploration of novel attack patterns.
Ashfaq et al. (2017)	NSL-KDD	Semi-supervised learning with fuzziness.	Enhanced classifier performance using labeled and unlabeled samples.	Performance assessed only for binary classification tasks.
Çavusoglu (2019)	NSL-KDD	Various ML techniques tested.	Good accuracy and low FPR across all attack types.	Used an older dataset; imbalanced and lacking novel attack patterns.

The literature review shows that the majority of existing research has been validated using older data sets which often lack novel attack patterns and suffer from unbalanced network audit data and limited to binary classification.

Such non-uniform data distribution can lead to biased training in ML algorithms, a challenge that remains unresolved in many studies. To address this, along with the older datasets, two newer datasets, like the UNSW-NB15 dataset and CIC-IDS are used that offer potential for improved detection. While some researchers have explored the CIC-IDS and UNSW-NB15 dataset, it remains underutilized and warrants further investigation.

3. Benchmark Datasets

Benchmark datasets play a vital role in the development and evaluation of IDSs. They provide standardized datasets, enabling researchers to systematically compare the performance of various algorithms and methodologies. These datasets are typically comprised of labeled network traffic, where each instance is annotated to specify whether it represents normal activity or a particular type of intrusion. A prominent benchmark dataset in intrusion detection research is the NSL-KDD dataset, an improved version of the KDD Cup 1999 dataset [1]. By addressing key limitations of the original dataset, such as redundancy and class imbalance, NSL-KDD offers a more reliable and effective framework for assessing IDS performance. It encompasses a diverse array of network traffic instances, including normal traffic as well as various types of attacks such as DoS, probing, and R2L attacks.

The UNSW-NB15 dataset [2] is another widely used benchmark, specifically designed for network-based intrusion detection research. It provides a rich set of features derived from network traffic collected in realistic environments, encompassing both normal and malicious activities.

In this paper, we utilized three well-established benchmark datasets—NSL-KDD, UNSW-NB15, and CIC-IDS—to evaluate the performance of various machine learning techniques. These datasets include a diverse mix of data types, such as integers, symbols, and categorical attributes, making them ideal for both binary and multiclass intrusion classification tasks. A detailed statistical summary of these datasets is presented in Table 2.

Table 2. Statistical Information of the Selected Datasets

Dataset	Features	Classes	Attack Types	Dataset Type
NSL-KDD	42	4	Dos, Probe, R2L,U2R	Multi Class
UNSW-NB15	45	9	Generic, Fuzzers, Exploits, Reconnaissance, Dos, Analysis, Worms, Backdoor, Shellcode	Multi Class
CIC-IDS	78	14	DoS Hulk, Web Attack, Brute force, PortScan, Bot, Infiltration, Dos GoldenEye, DDos, SSH-Patator, FTP-Patator, DoS slowloris, DoS Slowhttptest, Web AttackXSS, Web Attack Sql Injection, Heartbleed	Multi Class

4. Algorithms

Numerous studies have investigated the effectiveness of AI techniques in intrusion detection. These methods cover a wide range, including DT, NB, Bayesian Networks, NN, SVMs, Nearest Neighbor techniques, RF, and more. Each approach presents distinct benefits and limitations regarding accuracy, computational efficiency, and resilience to various cyber threats. A brief introduction of these techniques is given below.

4.1 Logistic Regression

LR is a statistical method used for binary classification tasks. It’s a type of regression analysis that’s particularly well-suited for situations where the dependent variable is categorical and has two possible outcomes, often labelled as 0 and 1. The LR model estimates the probability that a given input belongs to one of the two categories. It uses a threshold value (often 0.5) to determine whether the predicted probability corresponds to class 1 or class 0. If the predicted probability is greater than the threshold, the input is classified as an intrusion; otherwise, it is classified as normal activity.

4.2 Stochastic Gradient Descent

SGD is an optimization algorithm for training ML classifiers, especially in scenarios involving large datasets and deep learning(DL). It’s a variant of gradient descent, an algorithm that minimizes a loss function by iteratively adjusting the model’s parameters in the direction that reduces the error. Traditional gradient descent updates parameters based on the loss gradient across the entire dataset, which makes it computationally expensive for large datasets. Instead of calculating the gradient from the entire data set, SGD updates the parameters at each iteration based on a single randomly selected data point. SGD is a versatile optimization algorithm that can be applied to train ML models for intrusion detection tasks, providing efficient convergence and scalability for large scale datasets [8].

4.3 k-Nearest Neighbors

The kNN algorithm is a simple, intuitive, and effective ML method primarily used for classification and regression tasks. It works by identifying the k data points that are most similar to a given input and making predictions based on the properties of those neighbors. Overall, KNN offers a straightforward and intuitive approach to intrusion detection, making it a popular choice in both research and practical applications. However, its performance can be sensitive to the choice of k and the distance metric, and it may suffer from computational inefficiency when dealing with large datasets [9].

4.4 Decision Tree

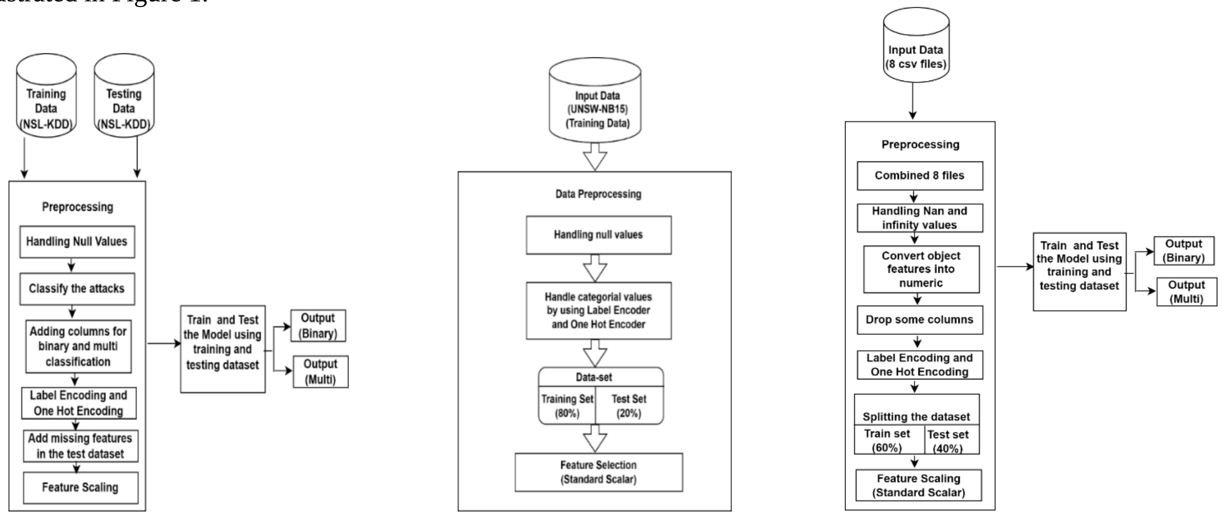
A DT is a popular ML classifier used for both classification and regression tasks. It is a graphical representation of a decision-making process that breaks down a complex decision into a series of simpler decisions. Each decision in the tree is based on a specific feature of the data. The process of creating a DT involves selecting the best attribute at each node to divide the data into subsets that are as identical as possible with respect to the target variable. This splitting process is performed recursively until certain stopping criteria are met, such as reaching a maximum tree depth or if further splitting does not significantly increase the homogeneity of the subsets. Classification and Regression Tree (CART) is a popular program for building decision trees[10].

4.5 Naive Bayes

It is a classification method that is specified by the Bayes Theorem. It is mainly used in intrusion detection due to its simplicity and efficiency. This method implies that the probabilities of each characteristic belonging to a specific class value is independent of the probability of other features. It can effectively learn the probabilities of different feature values given each class from training data and use this information to classify new instances. Overall, NB is a simple yet powerful algorithm for intrusion detection, especially suitable for scenarios with a large number of features and limited computational resources [11-12].

5. Preprocessing Strategies

The selected benchmark datasets comprise diverse data types, including integers, symbols, and categorical variables. The presence of non-uniform features necessitates pre-processing before applying ML classifiers. Appropriate pre-processing strategies have been developed and applied to the NSL-KDD, UNSW-NB15, and CICIDS datasets, as illustrated in Figure 1.


Training Data (NSL-KDD)
Testing Data (NSL-KDD)
Preprocessing
Handling Null Values
Classify the attacks
Adding columns for binary and multi classification
Label Encoding and One Hot Encoding
Add missing features in the test dataset
Feature Scaling
Train and Test the Model using training and testing dataset
Output (Binary)
Output (Multi)
Input Data (UNSW-NB15) (Training Data)
Data Preprocessing
Handling null values
Handle categorial values by using Label Encoder and One Hot Encoder
Data-set
Training Set (80%)
Test Set (20%)
Feature Selection (Standard Scalar)
Input Data (8 csv files)
Preprocessing
Combined 8 files
Handling Nan and infinity values
Convert object features into numeric
Drop some columns
Label Encoding and One Hot Encoding
Splitting the dataset
Train set (60%)
Test set (40%)
Feature Scaling (Standard Scalar)
Train and Test the Model using training and testing dataset
Output (Binary)
Output (Multi)

(a) NSL-KDD (b) UNSW-NB15 (c) CIC-IDS

FIGURE 1. Preprocessing Strategies

6. Experimental Setup

In this work, we performed empirical tests to assessed the effectiveness of five different ML algorithms: LR, NB, SGD, kNN, and DT. The preprocessed benchmark datasets NSL-KDD, UNSWNB15, and CICIDS were used to evaluate these methods. In order to assess the algorithms' performance across various network traffic classifications, our experimental setup included two different scenarios: binary classification and multiclass classification.

6.1 Results of Binary Classification

Our experimental results for binary classification in intrusion detection utilizing the ML algorithms LR, NB, SGD, kNN, and DT are summarized in Tables 3 to 5. These tables facilitate comparisons based on accuracy, recall, and FPR by offering information on how well each algorithm performs across various benchmark datasets. The comparative performance of different ML classifiers in the NSL-KDD dataset using binary classification is displayed in Table 3. Each method's FPR, precision, and recall metrics are assessed. The KNN classifier is notable for having the lowest FPR of 0.320, which shows how well it reduces false alarms. Furthermore, KNN's comparatively high Precision of 0.661 and Recall of 0.948 indicate that it can effectively detect true positives while reducing FN. With a high Precision and Recall score of 0.675 and 0.967, respectively, and an FPR of 0.307, DT also exhibits strong performance. Despite achieving a comparatively high Recall of 0.937, the NB classifier has a higher FPR of 0.561, indicating a higher rate of FP. On the other hand, the SGD classifier has the greatest FPR of 0.569 and a Precision of 0.000, indicating great potential for performance enhancement.

Overall, the results suggest that KNN and DT classifiers show promise for intrusion detection in the NSL-KDD dataset, while NB and SGD classifiers may require further optimization to enhance their effectiveness.

Table 3. Comparative Performance of ML Classifiers in NSl-Kdd Dataset - Binary Classification

Method	FPR	Precision	Recall
NB	0.561	0.037	0.937
KNN	0.320	0.661	0.948
LR	0.398	0.551	0.878
DT	0.307	0.675	0.967
SGD	0.569	0.000	0.667

The comparative performance of different machine learning classifiers in the UNSW-NB15 dataset for binary classification is shown in Table4. Classifiers such as NB, kNN, LR, DT, and SGD are evaluated. It is clear from study that the KNN classifier obtained the highest precision of 0.964, exhibiting its capacity of accurately identifying true positives while lowering false positives. Notably, though, KNN also demonstrated a comparatively higher FPR of 0.079 in contrast to other classifiers.

Conversely, LR showed the lowest FPR 0.023, indicating a lower false alarm rate. Additionally, LR obtained a high precision score of 0.991, showing its ability to accurately detect intrusions. In contrast to other classifiers, LR's recall score of 0.916 suggests that it might overlook some actual positive cases. Significantly, the NB classifier showed the highest false alarm rate FPR of 0.617. While NB's precision score of 0.234 indicates a significant number of FP, its perfect recall score of 1.000 indicates that it accurately detected all occurrences of true positives.

Overall, even while each classifier shows strengths in particular measures, such FPR or precision, the choice of classifier should take into account the particular needs and trade-offs related to the intrusion detection task.

Table 5 summarizes the comparative performance of various ML classifiers on the CIC-IDS dataset for binary classification. Among them, the NB classifier achieved the lowest FPR of 0.001, reflecting its strong capability to reduce false positives by minimizing the misclassification of normal instances as attacks. However, despite its perfect precision score of 1.000, its recall was slightly lower at 0.806, suggesting some attack instances were misclassified as normal. In contrast, the KNN classifier exhibited excellent performance, with high precision 0.989 and recall 0.994, effectively distinguishing between normal and attack instances. The LR classifier performed moderately, with an FPR of 0.147 and balanced precision 0.965 and recall 0.960 scores. The DT classifier outperformed LR and SGD in terms of FPR, achieving a lower value of 0.092, while maintaining high precision 0.984 and relatively strong recall (0.918). This highlights the DT classifier's ability to deliver reliable and accurate classifications. Lastly, the SGD classifier exhibited a relatively higher FPR of 0.160 but achieved commendable precision 0.962 and recall 0.952, indicating its capability to handle classification tasks with reasonable accuracy.

Overall, the findings highlight the distinct strengths and limitations of each classifier. While KNN excelled with outstanding precision and recall, the NB classifier proved highly effective in minimizing false positives.

Table 4. Comparative Performance of ML Classifiers in Unsw-Nb15 Dataset - Binary Classification

Method	FPR	Precision	Recall
NB	0.617	0.234	1.000
KNN	0.079	0.964	0.944
LR	0.023	0.991	0.916
DT	0.039	0.983	0.934
SGD	0.008	0.997	0.911

Table 5. Comparative Performance of ML Classifiers In CIC-IDS Dataset - Binary Classification

Method	FPR	Precision	Recall
NB	0.001	1.000	0.806
KNN	0.045	0.989	0.994
LR	0.147	0.965	0.960
DT	0.092	0.984	0.918
SGD	0.160	0.962	0.952

6.2 Results of Multi Classification

The results of experiments on multiclass classification for intrusion detection, utilizing ML algorithms such as LR, NB, SGD, kNN, and DT, are summarized in Tables 6 to 8. These tables provide valuable insights into the effectiveness of each algorithm across various benchmark datasets, facilitating comparisons based on precision, recall, and FPR. Table 6 presents the comparative performance of ML classifiers for multiclass classification using the NSLKDD dataset. In terms of FPR, SGD recorded the lowest values across all classes, reflecting its strong ability to correctly classify instances as benign (Normal) with minimal false alarms. Conversely, NB exhibited higher FPR values for most classes, particularly for DoS and R2L, indicating a greater likelihood of misclassifying benign instances as attacks. Regarding precision, DT achieved the highest values across all classes except Normal, demonstrating its effectiveness in accurately identifying instances of specific attack categories. In contrast, SGD reported a Precision of 0 for all classes except Normal, highlighting its tendency to misclassify most instances as benign. For Recall, DT also performed well, attaining high values for most classes and showcasing its capacity to identify attack instances effectively. However, both DT and SGD struggled with the R2L class, displaying lower Recall values and indicating challenges in correctly identifying this attack type. Overall, the results indicate that DT performs consistently well across multiple performance metrics and is particularly effective in distinguishing between various attack classes in the NSL-KDD dataset. In comparison, while SGD demonstrates low FPR, it underperforms in terms of Precision and Recall, especially for attack categories.

Table 6. Comparative Performance of ML Classifiers in NSL-KDD Dataset - Multi Classification

	0 (Normal)	**1(DoS)**	**2(Probe)**	**3(U2R)**	**4(R2L)**
NB					
FPR	0.085	0.861	0.009	0.066	0.016
Precision	0.215	0.330	0.000	0.000	0.003
Recall	0.031	0.828	0.000	0.179	0.000
KNN					
FPR	0.356	0.090	0.023	0.000	0.000
Precision	0.671	0.814	0.687	0.000	0.000
Recall	0.957	0.764	0.428	0.000	0.000
LR					
FPR	0.971	0.003	0.010	0.000	0.000
Precision	0.437	0.787	0.000	0.000	0.000
Recall	0.997	0.021	0.000	0.000	0.000
DT					
FPR	0.337	0.018	0.034	0.000	0.001
Precision	0.684	0.958	0.694	0.565	0.827
Recall	0.966	0.808	0.645	0.194	0.039
SVM					
FPR	1.000	0.000	0.000	0.000	0.000
Precision	0.431	0.000	0.000	0.000	0.000
Recall	1.000	0.000	0.000	0.000	0.000

Table 7 presents the comparative performance of ML classifiers in the UNSWNB15 dataset for multiclass classification. For NB, the FPR ranges from 0.442 to 0.266, with the highest value observed for the "Worms" class. Precision shows significant variation across classes, peaking at 0.854 for the "Exploits" class and dropping to 0.002 for the "Worms" class. Similarly, recall varies, reaching its highest at 0.793 for the "Worms" class and its lowest at 0.003 for the "Generic" class. KNN demonstrates lower FPR values compared to NB, ranging from 0.058 to 0.000 across classes. Its highest precision, 0.881, is observed for the "Normal" class, while the lowest, 0.500, occurs for the "Worms" class. Recall for KNN spans from 0.899 for the "Normal" class to 0.034 for the "Worms" class. LR achieves the lowest FPR among the three algorithms, ranging from 0.020 to 0.000 across classes. Its precision remains consistently high, with a peak of 0.952 for the "Normal" class and a minimum of 0.558 for the "Backdoor" class. Recall values for LR vary, with the highest at 0.878 for the "Exploits" class and the lowest at 0.012 for the "Backdoor" class. DT demonstrates similar patterns in FPR, precision, and recall as LR, albeit with slightly lower FPR values and comparable precision and recall across different classes. SGD achieves FPR values in the same range as LR, between 0.008 and 0.000 across classes. Its precision varies significantly, with the highest precision observed for the "Normal" class (0.979) and the lowest for the "Analysis" class (0.000). Recall for SGD spans from 0.816 for the "Normal" class to 0.000 for multiple attack categories. These results underscore the variability in performance across different ML classifiers, emphasizing the need to select classifiers that align with the specific characteristics of the dataset and the desired evaluation metrics.

Table 7. Comparative Performance of ML Classifiers in UNSW-N15 Dataset - Multi Classification

					NB					
	Normal (0)	**Generic (1)**	**Exploits(2)**	**Fuzzers(3)**	**DoS(4)**	**Reconnaissance(5)**	**Analysis (6)**	**Backdoor(7)**	**Shellcode (8)**	**Worms (9)**
FPR	0.442	0.085	0.002	0.100	0.001	0.300	2.700	0.050	0.124	0.266
Precision	0.326	0.009	0.854	11.100	0.346	20.200	1.800	0.030	0.050	0.002
Recall	0.450	0.003	0.041	0.100	0.007	1.300	4.400	0.166	1.000	0.793
					KNN					
FPR	0.058	0.003	0.111	0.047	0.042	0.015	0.001	0.000	0.002	0.000
Precision	0.881	0.990	0.620	0.593	0.294	0.687	0.528	0.455	0.542	0.500
Recall	0.899	0.977	0.764	0.591	0.235	0.537	0.122	0.031	0.286	0.034
					LR					
FPR	0.020	0.004	0.139	0.073	0.007	0.027	0.001	0.000	0.000	0.000
Precision	0.952	0.985	0.599	0.532	0.351	0.577	0.558	1.000	1.000	0.000
Recall	0.836	0.972	0.878	0.708	0.053	0.594	0.112	0.012	0.004	0.000
					DT					
FPR	0.005	0.001	0.145	0.068	0.004	0.004	0.002	0.000	0.005	0.000
Precision	0.987	0.998	0.606	0.593	0.500	0.926	0.449	0.723	0.475	0.800
Recall	0.811	0.980	0.941	0.846	0.048	0.723	0.158	0.104	0.634	0.138
					SVM					
FPR	0.008	0.001	0.150	0.088	0.000	0.020	0.000	0.000	0.000	0.000
Precision	0.979	0.997	0.591	0.512	0.941	0.620	0.756	0.000	0.000	0.000
Recall	0.816	0.971	0.917	0.796	0.007	0.524	0.081	0.000	0.000	0.000

Table 8 summarizes the comparative performance of various ML classifiers on the CIC-IDS dataset for multiclass classification tasks. For each classifier, metrics such as FPR, precision, and recall are reported for each attack category, highlighting variations in performance across different classifiers. For example, the NB classifier achieves high precision for benign instances and specific attack categories like Web Attack XSS and Heartbleed but shows comparatively lower recall for benign instances. Conversely, the KNN classifier demonstrates high precision and recall across most attack categories, underscoring its effectiveness in accurately detecting various attack types while reducing false positives. Similarly, classifiers like LR, DT, and SGD exhibit unique performance patterns, with varying levels of precision and recall for different attack categories. These results offer valuable insights into the strengths and weaknesses of various classifiers for intrusion detection using the CIC-IDS dataset, guiding the selection of suitable classifiers for specific attack scenarios and the optimization of IDSs.

Table 8. Comparative Performance of ML Classifiers In CIC-IDS Dataset - Multi Classification

Metric	**(1) Benign (normal)**	**(2) DoS Hulk**	**(3) Port Scan**	**(4) DDoS**	**(5) DoS GoldenEye**	**(6) FTP-Patator**	**(7) SSH-Patator**	**(8) DoS slow loris**	**(9) DoS Slow httptest**	**(10) Bot**	**(11) Web Attack Brute Force**	**(12) Web Attack XSS**	**(13) Infiltration**	**(14) Web SQL Attack Injection**	**(15) Heart-bleed**
								NB							
FPR	0.001	0.011	0.001	0.010	0.018	0.000	0.004	0.021	0.007	0.208	0.014	0.001	0.010	0.001	0.000
Precision	1.000	0.860	0.985	0.823	0.161	0.950	0.362	0.060	0.153	0.003	0.033	0.011	0.001	0.007	1.000
Recall	0.653	0.762	0.988	0.956	0.942	0.995	0.991	0.640	0.665	0.996	0.839	0.031	0.800	1.000	1.000
								KNN							
FPR	0.008	0.000	0.003	0.000	0.000	0.000	0.000	0.000	0.000	0.00	0.000	0.000	0.000	0.000	0.000
Precision	0.998	0.990	0.952	0.999	0.983	0.997	0.966	0.990	0.978	0.691	0.726	0.380	1.000	1.000	0.000
Recall	0.996	0.990	0.984	0.999	0.991	0.996	0.984	0.995	0.987	0.604	0.762	0.272	0.133	0.167	0.000
								LR							
FPR	0.058	0.007	0.014	0.000	0.000	0.000	0.000	0.000	0.000	0.00	0.000	0.000	0.000	0.000	0.000
Precision	0.986	0.926	0.809	0.993	0.983	0.911	0.102	0.856	0.817	0.583	0.000	0.000	0.000	0.000	0.000
Recall	0.975	0.960	0.992	0.960	0.991	0.551	0.004	0.513	0.874	0.019	0.000	0.000	0.000	0.000	0.000
								DT							
FPR	0.015	0.000	0.000	0.004	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Precision	0.996	0.991	0.994	0.929	0.983	0.999	0.999	0.983	0.985	0.000	1.000	0.000	0.000	0.000	0.000
Recall	0.999	0.991	0.998	0.997	0.991	0.995	0.987	0.897	0.754	0.000	0.047	0.000	0.000	0.000	0.000
								SGD							
FPR	0.130	0.004	0.015	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Precision	0.968	0.953	0.799	0.989	0.983	0.318	0.000	0.943	0.731	0.000	0.000	0.000	0.000	0.000	0.000
Recall	0.977	0.863	0.990	0.898	0.991	0.011	0.000	0.464	0.669	0.000	0.000	0.000	0.000	0.000	0.000

7. Conclusion

The empirical analysis of various conventional ML classifiers, including LR, NB, SGD, kNN, and DT is carried out. Experiments were conducted on three benchmark datasets—NSL-KDD, UNSW-NB15, and CICIDS—using standard evaluation metrics like precision, recall, and FPR to enable a comprehensive performance comparison. For binary classification, methods such as LR and DT demonstrated strong performance in detecting prevalent attack classes like Probe and DoS but struggled with accurately identifying minority attack classes, including U2R and R2L. On the other hand, techniques like NB and SGD achieved more balanced detection across various attack classes, though they still failed to attain optimal detection rates (DR) for all categories. Similarly, in multiclass classification, certain techniques excelled at identifying common attack classes but fell short in detecting less frequent ones. This disparity highlights the persistent challenge of achieving comprehensive intrusion detection across diverse attack scenarios, emphasizing the need for further research to develop more robust and adaptive detection strategies.

Acknowledgments

The authors would like to express their sincere gratitude to MRSPTU, Bathinda and SBSSTC, Ferozepur for providing the necessary resources and support for this research.

References

1. Kocher, G., & Kumar, G. (2021). Machine learning and deep learning methods for intrusion detection systems: recent developments and challenges. *Soft Computing*, *25*(15), 9731-9763.
2. Hamid, Y., Balasaraswathi, V. R., Journaux, L., & Sugumaran, M. (2018). Benchmark Datasets for Network Intrusion Detection: A Review. *Int. J. Netw. Secur.*, *20*(4), 645-654.
3. Kim, G., Lee, S., & Kim, S. (2014). A novel hybrid intrusion detection method integrating anomaly detection with misuse detection. *Expert Systems with Applications*, *41*(4), 1690-1700.
4. Belavagi, M. C., & Muniyal, B. (2016). Performance evaluation of supervised machine learning algorithms for intrusion detection. *Procedia Computer Science*, *89*, 117-123.
5. Guo, C., Ping, Y., Liu, N., & Luo, S. S. (2016). A two-level hybrid approach for intrusion detection. *Neurocomputing*, *214*, 391-400.
6. Ashfaq, R. A. R., Wang, X. Z., Huang, J. Z., Abbas, H., & He, Y. L. (2017). Fuzziness based semi-supervised learning approach for intrusion detection system. *Information sciences*, *378*, 484-497.
7. Çavuşoğlu, Ü. (2019). A new hybrid approach for intrusion detection using machine learning methods. *Applied Intelligence*, *49*, 2735-2761.
8. Nivaashini, M., Suganya, E., Sountharrajan, S., Prabu, M., & Bavirisetti, D. P. (2024). FEDDBN-IDS: federated deep belief network-based wireless network intrusion detection system. *EURASIP Journal on Information Security*, *2024*(1), 8.
9. Dhanabal, S., & Chandramathi, S. J. I. J. C. A. (2011). A review of various k-nearest neighbor query processing techniques. *International Journal of Computer Applications*, *31*(7), 14-22.
10. Kocher, G., & Kumar, G. (2020). Performance analysis of machine learning classifiers for intrusion detection using unsw-nb15 dataset. *Comput. Sci. Inf. Technol.(CS IT)*, *10*(20), 31-40.
11. Saleh, A. I., Talaat, F. M., & Labib, L. M. (2019). A hybrid intrusion detection system (HIDS) based on prioritized k-nearest neighbors and optimized SVM classifiers. *Artificial Intelligence Review*, *51*, 403-443.
12. Gu, J., & Lu, S. (2021). An effective intrusion detection approach using SVM with naïve Bayes feature embedding. *Computers & Security*, *103*, 102158.