Tech-Sphere Journal of Pure and Applied Sciences (TSJPAS)
A Subsidiary of Tech-Sphere Multidisciplinary International Journal (TSMIJ)
Fele et al. Vol 2, Issue 1, 2025 Publication Edition
ISSN: 3092-9598

Evaluation of Supervised Machine Learning Models with Recursive Feature Elimination for Network Traffic Attack Classification

Research Article https://stem.techspherejournal.com

Article Info
Revised Date: 1st September, 2025
Accepted Date: 5th September, 2025
Published Date: 9th September, 2025

Keywords
Machine Learning Algorithms
Network Traffic Attack CIC-IDS 2017
Feature Selection
Recursive Feature Elimination
Attacks Classification

Author Details
Fele Taiwo[1]*, Akinwamide Sunday Oluwafemi[2], Ojo Olufemi Ariyo[3]
1, 2, 3 Computer Science Department, Federal Polytechnic Ado-Ekiti, Ekiti State.

****Corresponding author's email***: fele_ta@fedpolyado.edu.ng
DOI: **https://doi.org/10.5281/zenodo.17072955**

This is an open-access article under the CC BY-SA license

ABSTRACT

The Internet is facing numerous attacks of different kinds that put its data at risk. The safety of information within the network is, therefore, a significant concern. One of the key challenges of machine learning Approaches for Network Traffic Attack Classification is the expensive computational complexity, which is largely due to redundant, incomplete, and irrelevant features contained in datasets for Network Traffic Attack Classification. In this work, we propose an approach for Network Traffic Attack Classification modeling approach with a Recursive Feature Elimination (RFE) algorithm for Feature Selection (FS) in Cognitive Radio Networks. The FS algorithm is a wrapper-based algorithm with a decision tree as the feature evaluator. The proposed FS method is used in combination with some selected supervised Machine Learning algorithms to build Network Traffic Attack Classification models using the CIC-IDS 2017 dataset. We evaluate the effectiveness of our proposed method by comparing the classification performance of different supervised learning algorithms using standard performance metrics. The implemented experiments compare the results of each algorithm and demonstrate that the Random Forest is the best algorithm used for the network traffic classification with accuracy, precision, recall and F1-score parameter of 0.9998, 0.9894, 0.9975 and 0.9934 respectively while Naïve Bayes achieves the lowest accuracy precision, recall and F1- score parameter of 0.9547, 0.7368, 0.9820 and 0.7844 respectively.

1 Introduction

Network security is now one of the most significant concerns with the network's explosive development because it directly influences the interests of the nation, companies, and individuals. The internet progression and the fast exchange of data bring the threat of increasing cyber-attacks targeting governments and commercial enterprises worldwide at a rapid rate. Network traffic classification (NTC) in network engineering represents an important process for categorizing network traffic in accordance with different parameters (i.e., port number, protocol, etc.). It presents a way for quantifying, monitoring, and understanding network traffic in order to troubleshoot network issues for an Internet Service Provider (ISP), in other words, it is used to estimate the network system's capacity based on Quality of Service (QoS) of traffic flow or lawful interception [1]. Machine learning-based traffic classification has become essential in network security research, particularly with the increasing prevalence of encrypted communication. As traditional

Tech-Sphere Journal of Pure and Applied Sciences (TSJPAS)
A Subsidiary of Tech-Sphere Multidisciplinary International Journal (TSMIJ)
Fele et al. Vol 2, Issue 1, 2025 Publication Edition
ISSN: 3092-9598

methods like deep packet inspection lose effectiveness, ML provides a viable and often the only alternative for identifying communication patterns.

Machine learning (ML) is a crucial tool for enabling Artificial Intelligence [2] as it can effectively predict and schedule network resources based on the available data inputs [3], [4]. It has applications in various areas, providing data acquisition and analysis by emulating human learning behavior of knowledge [5]. Machine learning aims to enable computers to determine and enhance their performance over time without being explicitly programmed to do so [6]. ML algorithms can be supervised, unsupervised, semi-supervised, or reinforcement, varying based on the type of data utilized for model training [7], [8]. Supervised learning is the process of training a model using labeled data when the right output for each input is known. Unsupervised learning includes finding patterns and relationships in unlabeled data. Semi-supervised learning is a set of both supervised learning and unsupervised learning. In reinforcement learning, an agent learns to act in a given environment to maximize a reward [9]. Figure 1 provides an overview of the ML algorithms.

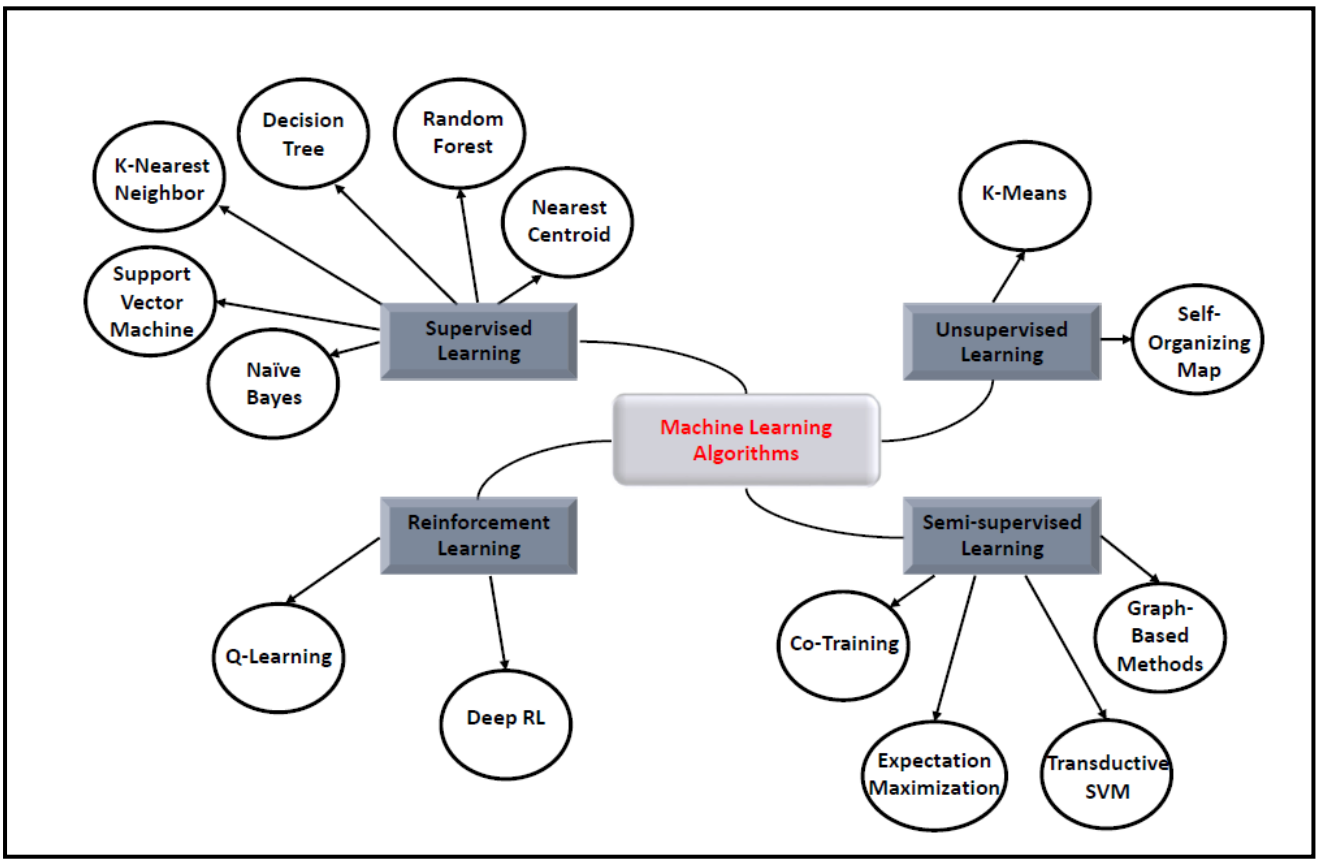

Figure 1: Machine Learning Algorithms.

Machine Learning models can be trained based on different learning approaches. Supervised Learning-based models are trained with labeled data, as shown in Figure 2. In Intrusion Detection Systems, most of the intrusion or anomaly detecting tasks are classified using different Supervised Learning-Based models [10]. On the other hand, unsupervised learning-based models gather information from unlabeled data, as shown in Figure 3. In contrast, reinforcement learning-based models rely on continuous feedback from critics based on some particular actions.

Tech-Sphere Journal of Pure and Applied Sciences (TSJPAS)
A Subsidiary of Tech-Sphere Multidisciplinary International Journal (TSMIJ)
Fele et al. Vol 2, Issue 1, 2025 Publication Edition
ISSN: 3092-9598

1.1 Supervised Learning-Based Models

In the existing IDS literature in SDN, most of the researchers performed classification of network traffic using some supervised Machine Learning models. In Supervised Learning, the machine is trained with well- labeled dataset and aids in the prediction of unseen data as shown in Figure 2. It implies that some input data has already been marked with the accurate output label. Decision Tree, Naïve Bayes, Random Forest, K-Nearest Neighbor and Support Vector Machine are used more frequently in intrusion detection compared to other supervised models.

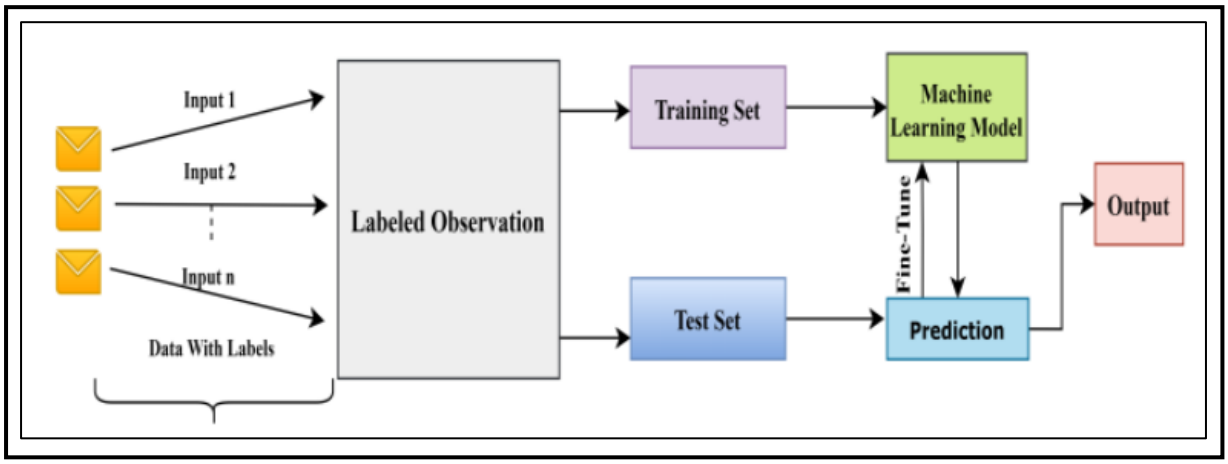

Figure 2: Supervised Learning method structure

A Decision Tree (DT) is a tree-based representation of data and has a structure of nodes. Each node represents a decision to be taken based on features, and the leaf nodes of the tree denote the class label [11]. The route from the root node to the leaf node of a particular class depicts the classification rule. Naïve Bayes (NB) classifier is a probabilistic Machine Learning model. Naïve Bayes classifier assigns some probability measure by calculating the frequency or density of the feature values provided by the input dataset. NB classifier assumes that every feature is conditionally independent given the label [12].

Random Forest is a very popular machine learning algorithm. It is an ensemble learning algorithm, meaning that it creates many decision trees (this is the reason it is called a forest) during the training phase of the algorithm and then produces the most popular output as its classification. It is also used for regression or prediction problems. Random forest is an ensemble classification method that combines a collection of classifiers (i.e., decision trees) to make a "forest". Each of the decision trees is generated by using a random selection of attributes at each node to determine the split.

Support Vector Machine (SVM) finds a hyperplane to separate the data into two different classes [13]. SVM finds some points nearest to the hyperplane separating the classes; these points are called support vectors. The goal is to maximize the margin or width of the hyperplane, separating the support vectors selected from both classes.

K-Nearest Neighbor (KNN) is an instance-based lazy classifier that performs relatively well without any assumptions about the underlying data [14]. This nonparametric feature is a compelling aspect, as most real-world data do not reflect any fundamental foundations and assumptions, e.g., linear independence, uniform or normal distribution, etc.

Tech-Sphere Journal of Pure and Applied Sciences (TSJPAS)
A Subsidiary of Tech-Sphere Multidisciplinary International Journal (TSMIJ)
Fele et al. Vol 2, Issue 1, 2025 Publication Edition
ISSN: 3092-9598

1.2 Unsupervised Learning-Based Models

Unsupervised Learning-based models are used when the class label is unknown. In most of the IDS, the Unsupervised Learning-based algorithm is used for cluster analysis. As shown in Figure 3, Unsupervised Learning-based algorithms do not require supervision and are trained using unlabeled data. Rather, it investigates the trends and patterns on its own. The unsupervised model's goal is to identify the group and categorized them according to their shared attributes. The most popular example of unsupervised models that are being used for intrusion detection are K-Means and SOM algorithm.

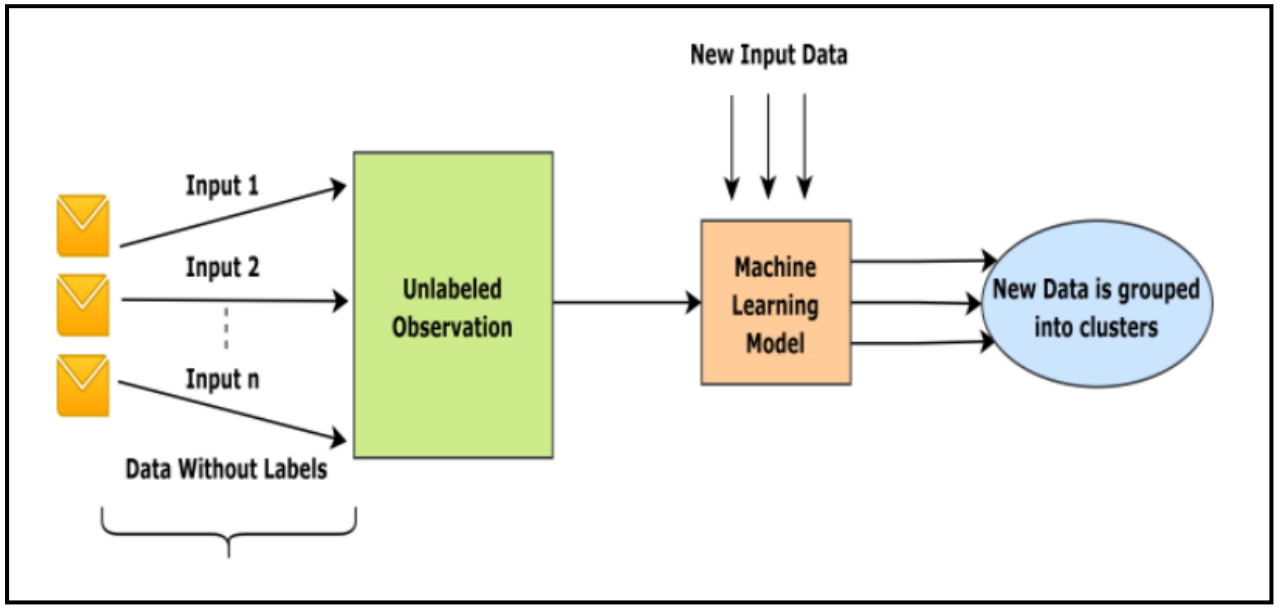

Figure 3: Unsupervised Learning method structure

K-Means Clustering K-means clustering method is used to find groups from a given dataset where several groups are represented by variable K. Generally, centroids are picked randomly, and K-clusters are formed. It works iteratively to assign a data point to a particular cluster. At the end of the iterations, each data point is clustered according to its feature similarity. Each time the cluster center (mean of a cluster) is updated, iterated until the criterion function converges [10]. Self-Organizing Map (SOM) SOM is an UL method that creates a nonlinear mapping of a high-dimensional data manifold on a regular, low-dimensional output space [15]. Using dimensionality reduction, they can cluster large amounts of data. Compared to the performance of other clustering algorithms, such as K-Means, the SOM output allows for simple visualization.

Hidden Markov Model (HMM) The Hidden Markov Model is a probabilistic model based on the Markov processes that have been used in a variety of research fields, including bioinformatics, speech recognition, and network intrusion detection [16], [17]. It enables us to forecast a series of hidden (unknown) states based on a set of observed states. HMMs can be applied to detect complicated internet attacks with a high noise ratio because of the variations in action sequence throughout execution of each identical attack [18].

Tech-Sphere Journal of Pure and Applied Sciences (TSJPAS)
A Subsidiary of Tech-Sphere Multidisciplinary International Journal (TSMIJ)
Fele et al. Vol 2, Issue 1, 2025 Publication Edition
ISSN: 3092-9598

1.3 Reinforcement Learning

Reinforcement Learning is one of three primary machine learning paradigms, next to supervised and unsupervised learning. Reinforcement Learning is concerned with how intelligent agents can achieve a goal in an unknown, potentially intricate environment in order to optimize the concept of total collective reward. Reinforcement Learning can be used to solve problems where notable domain information is either inaccessible or prohibitively expensive [19]. In most cases, a function approximation, such as a Neural Network, SVM, etc., is used to map state to value. For an Intrusion Detection System, designing a reward feature associated with the detection of intrusions is incredibly challenging because there is no automated approach to distinguish intrusions from the normal traffic flow. Algorithms such as Q-learning, Deep-Q Network (DQN), and Proximal Policy Optimization (PPO) are mostly used in RL-based IDS for SDNs. Q-learning is an off-policy RL algorithm that determines the optimum course of action given the present situation [20]. Because the Q-learning function learns from its actions and isn't reliant on the existing policy, it's termed off-policy.

2 Related Works

This section presents some of the current research related to the topic of network traffic classification in the scope of machine learning. There has been much research on network traffic classification for improving network security in recent years. This section looks at some of the most recent conference papers and journal publications that present machine learning-based algorithms for network traffic classification.

Azab et al. [21] reviewed various approaches for network traffic classification, focusing on methods used to identify different types of traffic, such as video conferencing, email, or malicious activity. The study examined both traditional and modern techniques, including port-based methods, deep packet inspection (DPI), statistical approaches with machine learning (ML), deep learning (DL), and hybrid semi-supervised strategies. In addition, the authors discussed datasets commonly employed in prior research, highlighting their advantages and limitations. Special emphasis was placed on the challenges of real-time monitoring and the complexities introduced by emerging traffic sources such as mobile applications, Internet of Things (IoT) devices, and encrypted communications. Their comparative analysis provided a comprehensive overview of the strengths and weaknesses of existing methods while identifying persistent research gaps in traffic classification.

Mahmood et al. [22] demonstrated the application of machine learning techniques for internet traffic classification, emphasizing its significance for Internet Service Providers (ISPs) in enhancing service quality and mitigating security threats. In their study, network traffic was captured using Wireshark, and relevant features such as packet length and duration were extracted with NetMate. Four machine learning algorithms—C4.5, Support Vector Machine (SVM), BayesNet, and Naïve Bayes—were employed to classify traffic such as WWW, DNS, FTP, P2P, and Telnet. The evaluation, conducted using 10-fold cross-validation in Weka, revealed that the C4.5 decision tree achieved the highest classification accuracy at 78.91%, outperforming the other models. This result highlights the effectiveness of decision-tree–based approaches for network traffic classification in comparison with alternative ML techniques.

Canever and Wang [23] explored the use of unsupervised machine learning techniques for clustering internet traffic data, aiming to identify the most suitable method for effective traffic classification. Such classification is essential for applications including bandwidth management and threat detection. The authors evaluated four clustering algorithms—K-means, DBSCAN, Bisecting K-means, and K-modes—on a large real-world dataset comprising more than 3.5 million network flows collected from a university network. The dataset underwent pre-processing, including cleaning and feature selection, where 15 key attributes were retained. Multiple hyperparameter settings were tested, and the algorithms were assessed using clustering validity indices to evaluate the quality of the groupings. Their findings provided insights into the comparative performance of clustering techniques for practical network traffic analysis.

Hossen et al. [24] investigated the application of machine learning techniques to enhance security in cloud computing environments, with a particular focus on detecting malicious internet traffic such as botnet activities. Their approach involved training models to distinguish between benign and harmful behaviors in network flows, thereby enabling the early detection and prevention of cyberattacks before they cause significant damage. The study highlights the effectiveness of machine learning in strengthening intrusion detection systems (IDS) and underscores its importance in safeguarding cloud infrastructures against evolving threats.

Shafiq et al. [25] examined various approaches for network traffic classification, comparing traditional techniques with machine learning–based methods to identify traffic types such as web browsing, file transfers, and potential malicious activity. In their study, real-time internet traffic was captured using Wireshark from applications including WWW, DNS, FTP, P2P, and Telnet. A total of 23 features—such as packet size and timing—were extracted using NetMate and classified using four machine learning algorithms: C4.5 decision tree, Support Vector Machine (SVM), BayesNet, and Naïve Bayes. The models were evaluated using the Weka tool with performance metrics such as accuracy, precision, and recall. Among the classifiers, the C4.5 decision tree achieved the highest accuracy, confirming its effectiveness for traffic classification in comparison with the other models.

Priya et al. [26] addressed the security challenges associated with Internet of Things (IoT) devices, which are highly vulnerable to cyberattacks due to their limited computational capabilities. The study explored the use of machine learning (ML) and deep learning techniques to classify IoT network traffic for improved intrusion detection and enhanced Quality of Service (QoS). IoT traffic datasets, including botnet-infected traffic, were collected using tools such as Wireshark. The data underwent pre-processing steps including cleaning, feature extraction (e.g., packet lengths, protocols, and flow attributes), and normalization. The authors evaluated multiple models—Random Forest, Support Vector Machine (SVM), and Convolutional Neural Networks (CNNs)—on both binary and multiclass classification tasks. Model performance was assessed using accuracy, precision, recall, F1-score, and ROC-AUC, demonstrating the effectiveness of ML-based methods in securing IoT environments against evolving threats.

Jerabek et al. [27] critically examined the reliance on complex machine learning models for network traffic classification, arguing that simpler approaches may achieve comparable or superior performance. The study demonstrated that a k-Nearest Neighbor (k-NN) baseline, using packet metadata such as sizes, timestamps, and directions, can match or outperform several state-of-the-art (SOTA) methods. The authors highlighted significant issues with existing research practices, including dataset redundancy and flawed evaluation protocols. Their baseline model applied k-NN with L1 distance on packet sequence features and was tested across 12 benchmark datasets, including ISCXVPN2016 and CIC-IDS-2017, using both random and time-based splits. To further support their claims, they analyzed redundancy by detecting duplicate samples and estimated the maximum achievable accuracy for each dataset. Their findings call into question the necessity of overly complex models when simpler baselines provide strong results under rigorous evaluation.

Serag et al. [28] explored the integration of machine learning (ML) techniques with Software-Defined Networking (SDN) to improve traffic classification (TC), Quality of Service (QoS), and network security. Leveraging SDN's centralized control and programmability, the study emphasized how ML models enhance traffic analysis by enabling anomaly detection, intrusion prevention, and resource optimization. The authors reviewed traditional versus ML-based TC approaches, highlighting methods such as Random Forest, Support Vector Machine (SVM), decision trees (DT), deep learning, and unsupervised clustering (K-means), as well as semi-supervised techniques like Laplacian SVM. Data collection and pre-processing involved SDN-generated traffic (e.g., flow statistics and packet headers) along with public datasets such as CIC-IDS-2017, with a focus on feature extraction (e.g., packet size, protocol type) and handling noisy or missing data. Model optimization was carried out using hyperparameter tuning strategies, including Optuna for k-NN. The models were evaluated with metrics such as accuracy, precision, recall, F1-score, and ROC-AUC across TC and security tasks, including DDoS detection. The findings demonstrated the potential of combining ML with SDN for

adaptive, secure, and efficient network management, while also identifying challenges related to dataset quality, scalability, and deployment feasibility.

Dawood [1] evaluated the performance of Naïve Bayes algorithms for network traffic classification. The study utilized two datasets: ISCXVPN2016, which distinguishes VPN from non-VPN traffic, and a custom Wi-Fi video-streaming dataset. The analysis focused on traffic classes involving payload protocols such as QUIC and TCP, as well as security protocols including TLS and SSL. The models were assessed using accuracy, F1-score, and processing time as key performance metrics. The comparative results highlighted trade-offs between accuracy and computational efficiency, demonstrating the strengths and limitations of each NB variant for real-time traffic classification tasks.

Wang, Fok, and Thing [29] investigated approaches to detecting encrypted malicious traffic using machine learning. The authors proposed a universal six-step framework for machine learning (ML)-based encrypted traffic detection, encompassing objective definition, dataset collection, feature extraction, algorithm selection, model training, and evaluation. To ensure diversity and balance, five publicly available datasets were integrated into a comprehensive benchmark covering both legitimate and malicious traffic from IoT and conventional devices. Within this framework, ten algorithms, including Random Forest (RF), XGBoost, and Convolutional Neural Networks (CNNs)—were evaluated across five different feature sets, such as flow-oriented statistics (FOS) and side-channel characteristics. The results provided a systematic comparison of ML approaches for encrypted traffic detection and highlighted the advantages and limitations of each method under varying feature representations.

3 Methodology

This section describes how the experiment is conducted by following the four basic machine learning steps (i.e., data acquisition, data pre-processing, model selection and training, and model evaluation). It also provides the tools used in experimentation.

3.1 Experimental Tools

In the literature, many tools are used for implementing, evaluating, and comparing various Network Traffic Classification works. WEKA, general-purpose programming languages (such as C, Java, Python, etc.), and Matlab are the most used tools [5]. In this work, Excel and Python are used for data analysis and exploration, pre-processing, implementing, and evaluating the supervised learning models. Jupyter Notebook is used as the execution environment for Python and its libraries.

Tech-Sphere Journal of Pure and Applied Sciences (TSJPAS)
A Subsidiary of Tech-Sphere Multidisciplinary International Journal (TSMIJ)
Fele et al. Vol 2, Issue 1, 2025 Publication Edition
ISSN: 3092-9598

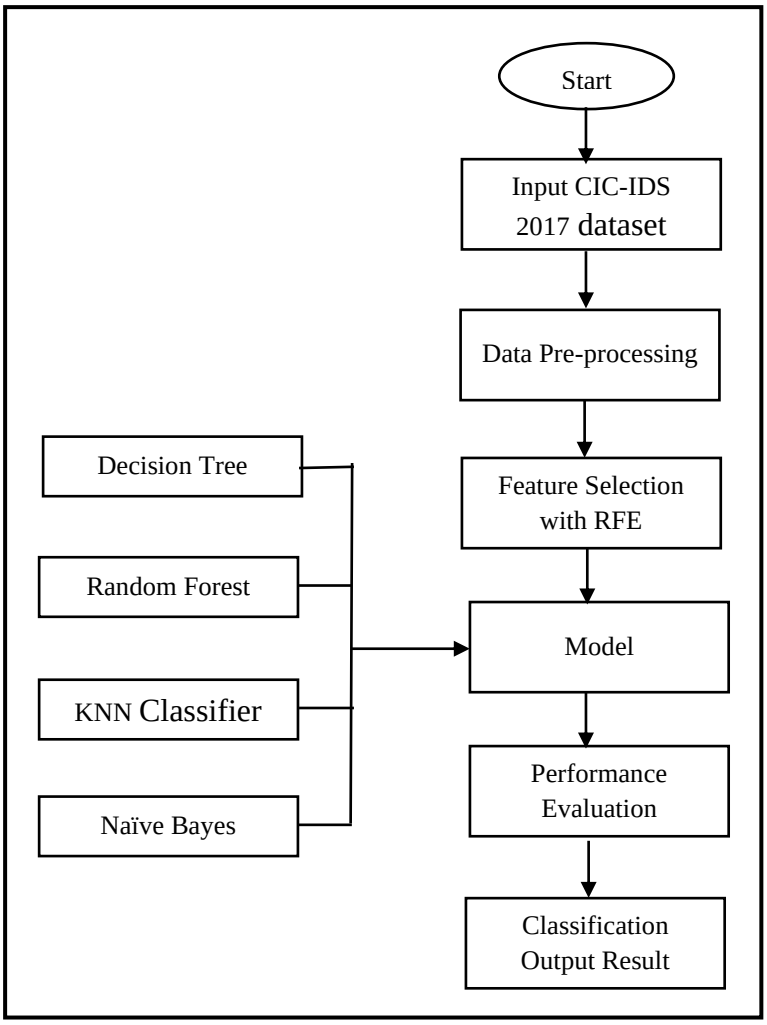

Figure 4: Proposed framework for Network Traffic Classification

3.2 Dataset Acquisition

This section shows the collected data and its description. We collected a dataset with 83 attributes, where 82 are prediction variables and 1 response variable for multiclass classification.

3.2.1 CIC-IDS 2017 Dataset

Sharafaldin et al. [30] proposed the CICIDS2017 dataset as a benchmark for evaluating intrusion detection systems. It contains two types of network traffic: normal and attack. One major drawback of the CICIDS2017 dataset is that it has class imbalance issues [31]. The dataset contains almost 1,048,575 records of intrusion behaviors with their parameters labeled in a particular class. Hence, this dataset was selected for training our model using four different supervised learning algorithms.

Tech-Sphere Journal of Pure and Applied Sciences (TSJPAS)
A Subsidiary of Tech-Sphere Multidisciplinary International Journal (TSMIJ)
Fele et al. Vol 2, Issue 1, 2025 Publication Edition
ISSN: 3092-9598

Table 1: Characteristics of CICIDS2017 Dataset

Dataset	**CICIDS2017**
Number of Features	82
Feature Extraction Tool	CICFlow-Meter [31]
Label	Yes
Metadata	Yes
Format	Packet + Flow
Instances	1,048,575
Number and Name of Available Attack Types	(4) Brute Force, DDoS, Port Scanning and Web Attack.
Traffic + Network Type	Realistic network traffic + Small-Scale Network
Balanced	No
Attack Diversity	Yes

The dataset comprises benign traffic and recent common attacks, mirroring real-world PCAP data. It incorporates network traffic analysis outcomes from CICFlow Meter, featuring labeled flows categorized by timestamps, source and destination IPs, ports, protocols, and attack types in CSV format. Created by the Canadian Institute for Cybersecurity (CIC).

3.3 Data Pre-processing

At this point, the collected data is arranged and transformed into a format that can be fed into the machine learning algorithm. In most cases, this is stored as a table or a NumPy array. Feature extraction also occurs at this point (not all information from the data collection stage is relevant to the experiment, as such, some things are ignored entirely).

3.4 Feature Selection

In any classification problem, dealing with large datasets may require selecting the most useful and relevant features, as many features may contain false correlations, redundant, and irrelevant features, which can increase computation time, impact the accuracy of the built model. Feature Selection is one of the important and frequently used data pre-processing techniques for selecting the optimal subset of relevant features from the original features for model construction. Feature selection reduces the number of features by removing irrelevant, redundant, or noisy data and has immediate effects on the subsequently built model.

3.4.1 Feature Selection Methods

The dataset consists of 82 features. Such a number of features imposes the issue of high dimensionality. The issue relates to the fact that with a high number of features comes the risk of encountering redundancy between features or features that do not provide information useful to the clustering, leading to an overall reduction in the performance of the algorithm. At this phase, we select features based on the Recursive Feature Elimination (RFE) feature selection method.

Tech-Sphere Journal of Pure and Applied Sciences (TSJPAS)
A Subsidiary of Tech-Sphere Multidisciplinary International Journal (TSMIJ)
Fele et al. Vol 2, Issue 1, 2025 Publication Edition
ISSN: 3092-9598

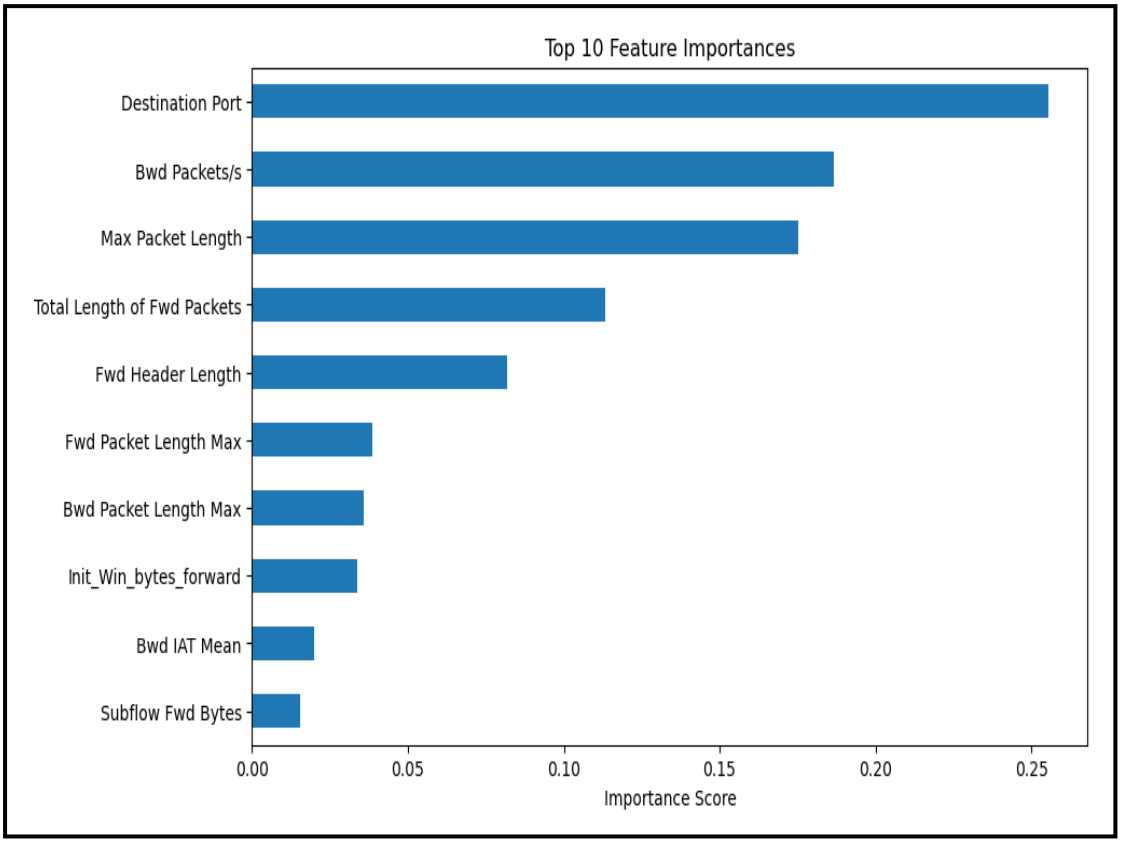

Figure 5: Importance of feature selection

3.5 Model Selection and Training

The pre-processed data is split into training datasets (838,860) and test datasets (209,715). The training dataset from the data pre-processing stage is fed into the chosen machine learning algorithm, and a model is built. Once the model has been built, the test dataset from the data pre-processing stage is fed into to model in the same format as the training dataset. The classification or prediction accuracy is taken. The closer to a hundred percent the accuracy is, the better the model. To test our work, the CICIDS2017 dataset, widely deployed in literature for network traffic classification in cloud computing, has been used with four categories of attacks: Brute Force, DDoS, Port Scanning, and Web Attack.

3.6 Model Evaluation Metrics

This section discusses standard performance metrics. Confusion Matrix, Accuracy, Precision, Recall, and F1-Score are all commonly used measures to evaluate the performance of a classification model. The confusion matrix compares predicted and true class labels to summarize the performance of the classification model considering True Positive (TP) as the number of cases correctly predicted by the model as positive, False Positive (FP) as the number of instances in which the model made an incorrectly positive prediction, False Negative (FN) as the number of instances that were incorrectly predicted by the model to be negative, and True Negative (TN) as the number of cases successfully obtained from the model as negative.

Tech-Sphere Journal of Pure and Applied Sciences (TSJPAS)
A Subsidiary of Tech-Sphere Multidisciplinary International Journal (TSMIJ)
Fele et al. Vol 2, Issue 1, 2025 Publication Edition
ISSN: 3092-9598

Accuracy measures the overall correctness of the model's predictions and is defined as (1):

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

Precision, defined as the ratio of actual positive results to those that a model anticipated would be positive, is measured by the following formula (2):

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

The percentage of correct answers is expressed as a recall percentage, and it is defined as (3):

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

F1-score is the harmonic mean of precision and recall and is defined as (4):

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

To summarize, the confusion matrix gives a more detailed assessment of the model's performance, whereas accuracy, precision, Recall, and F1-score are summary statistics that provide several viewpoints on the model's performance.

4 Result and Discussion

In this section, we discuss and show the comparative results from each algorithm. We can see the confusion matrix from each algorithm and compare the performance matrix in Figures 6a, 6b, 6c, and 6d. This section describes all the results that we got after applying four different supervised learning algorithms.

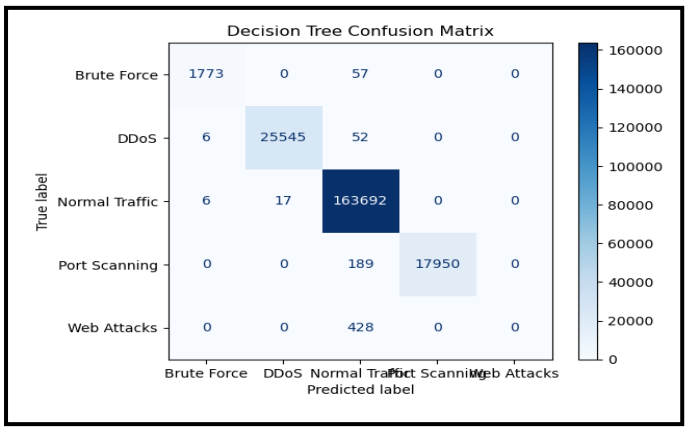

Figure 6a: Confusion matrix for decision tree

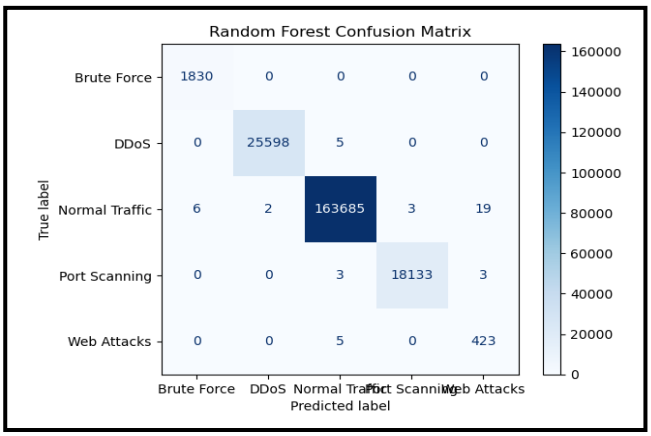

Figure 6b: Confusion matrix for random forest

Tech-Sphere Journal of Pure and Applied Sciences (TSJPAS)
A Subsidiary of Tech-Sphere Multidisciplinary International Journal (TSMIJ)
Fele et al. Vol 2, Issue 1, 2025 Publication Edition
ISSN: 3092-9598

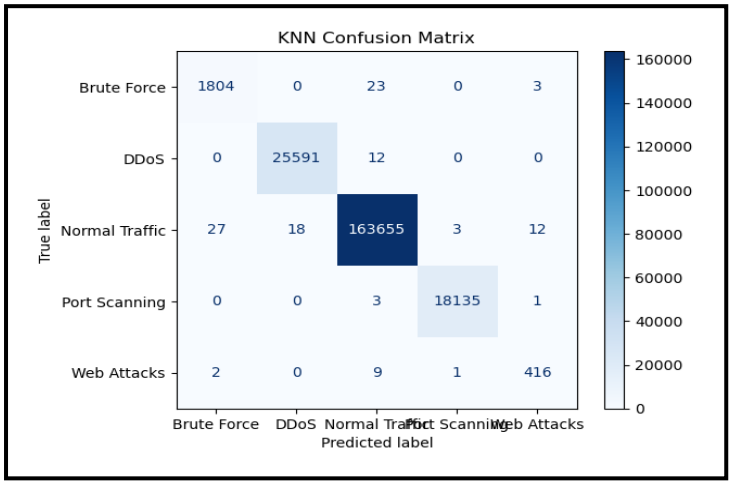

Figure 6c: Confusion matrix for KNN

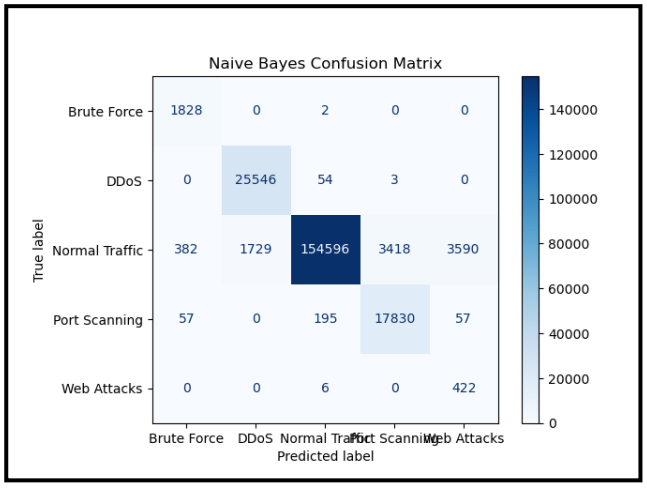

Figure 6d: Confusion matrix for Naïve Bayes

An exploration of the above results shows that while the accuracy is suitable for most of the algorithms applied individually, other parameters did not give good results. Performance Evaluation of some selected supervised machine learning algorithms is presented in Tables 2, 3, 4, 5, and 6. Web Attacks class shows 0.0000 precision/recall in the Decision Tree results in Table 2 because web attack samples are very few compared to major attack classes. Decision trees tend to be biased towards the majority classes.

Table 2: Decision Tree model performance with an Accuracy of 0.9964

	Precision	**Recall**	**F1-score**	**Support**
Brute Force	0.9933	0.9689	0.9809	1830
DDoS	0.9993	0.9977	0.9985	25603
Normal Traffic	0.9956	0.9999	0.9977	163715
Port Scanning	1.0000	0.9896	0.9948	18139
Web Attacks	0.0000	0.0000	0.0000	428
macro avg	0.7976	0.7912	0.7944	209715
weighted avg	0.9944	0.9964	0.9954	209715

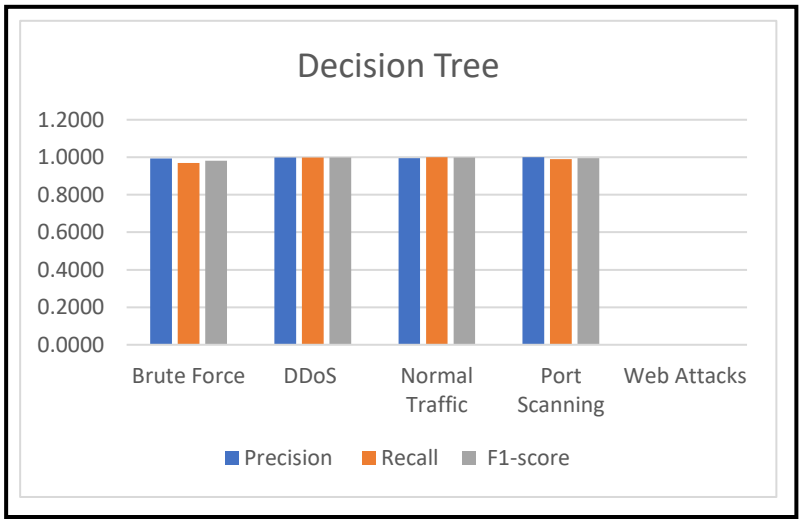

Figure 7: Decision Tree model performance with an Accuracy of 0.9964

Tech-Sphere Journal of Pure and Applied Sciences (TSJPAS)
A Subsidiary of Tech-Sphere Multidisciplinary International Journal (TSMIJ)
Fele et al. Vol 2, Issue 1, 2025 Publication Edition
ISSN: 3092-9598

Table 3: Random Forest model performance with an Accuracy of 0.9998

	Precision	Recall	F1-score	Support
Brute Force	0.9967	1.0000	0.9984	1830
DDoS	0.9999	0.9998	0.9999	25603
Normal Traffic	0.9999	0.9998	0.9999	163715
Port Scanning	0.9998	0.9997	0.9998	18139
Web Attacks	0.9506	0.9883	0.9691	428
macro avg	0.9894	0.9975	0.9934	209715
weighted avg	0.9998	0.9998	0.9998	209715

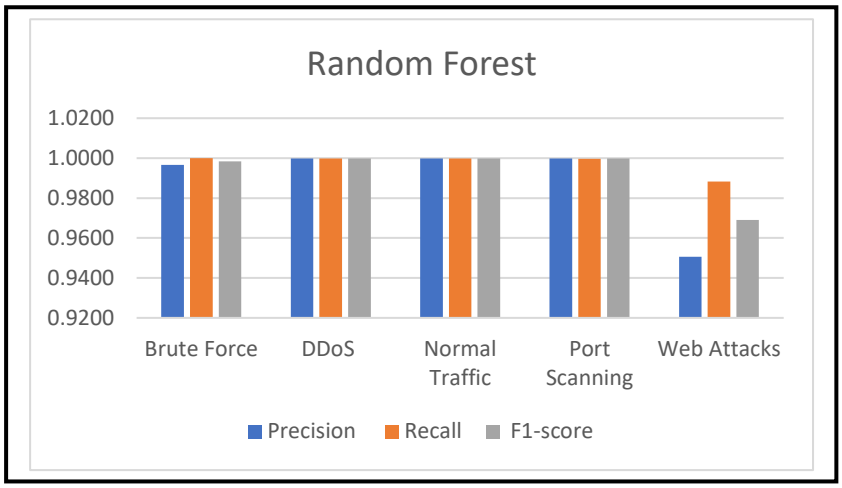

Figure 6: Random Forest model performance with an Accuracy of 0.9998

Table 4: K-Nearest Neighbor model performance with an Accuracy of 0.9995

	Precision	Recall	F1-score	Support
Brute Force	0.9842	0.9858	0.9850	1830
DDoS	0.9993	0.9995	0.9994	25603
Normal Traffic	0.9997	0.9996	0.9997	163715
Port Scanning	0.9998	0.9998	0.9998	18139
Web Attacks	0.9630	0.9720	0.9674	428
macro avg	0.9892	0.9913	0.9903	209715
weighted avg	0.9995	0.9995	0.9995	209715

Tech-Sphere Journal of Pure and Applied Sciences (TSJPAS)
A Subsidiary of Tech-Sphere Multidisciplinary International Journal (TSMIJ)
Fele et al. Vol 2, Issue 1, 2025 Publication Edition
ISSN: 3092-9598

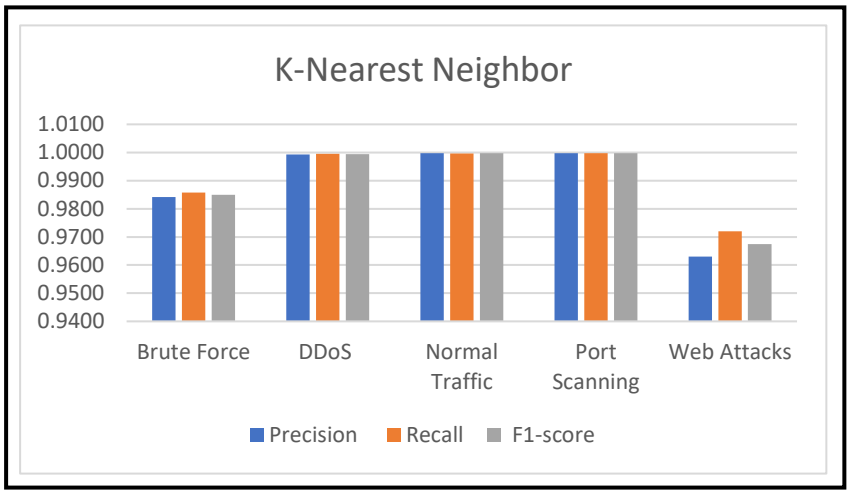

Figure 7: K-Nearest Neighbor model performance with an Accuracy of 0.9995

Table 5: Naïve Bayes model performance with an Accuracy of 0.9547

	Precision	**Recall**	**F1-score**	**Support**
Brute Force	0.8064	0.9989	0.8924	1830
DDoS	0.9366	0.9978	0.9662	25603
Normal Traffic	0.9983	0.9443	0.9706	163715
Port Scanning	0.8390	0.9830	0.9053	18139
Web Attacks	0.1037	0.9860	0.1877	428
macro avg	0.7368	0.9820	0.7844	209715
weighted avg	0.9735	0.9547	0.9621	209715

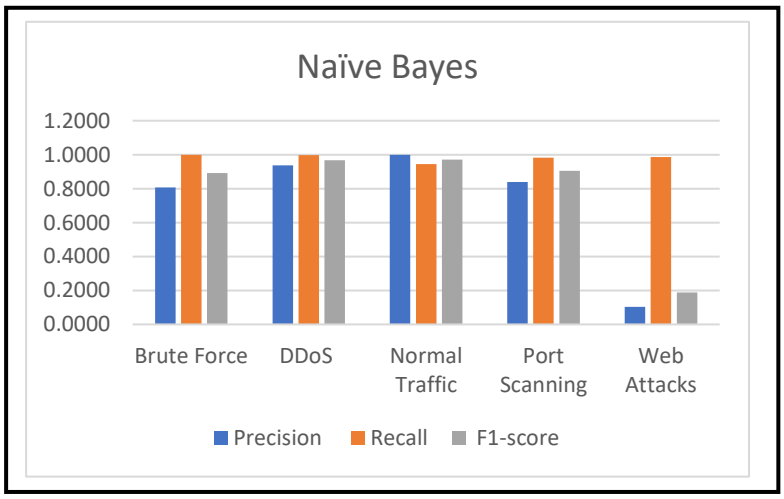

Figure 8: Naïve Bayes model performance with an Accuracy of 0.9547

Tech-Sphere Journal of Pure and Applied Sciences (TSJPAS)
A Subsidiary of Tech-Sphere Multidisciplinary International Journal (TSMIJ)
Fele et al. Vol 2, Issue 1, 2025 Publication Edition
ISSN: 3092-9598

Table 6: Comparison between classification techniques with RFE

Classifier	Accuracy	Precision	Recall	F1-score
Decision Tree	0.9964	0.7976	0.7912	0.7944
Random Forest	0.9998	0.9894	0.9975	0.9934
K-Nearest Neighbor	0.9995	0.9892	0.9913	0.9903
Naïve Bayes	0.9547	0.7368	0.9820	0.7844

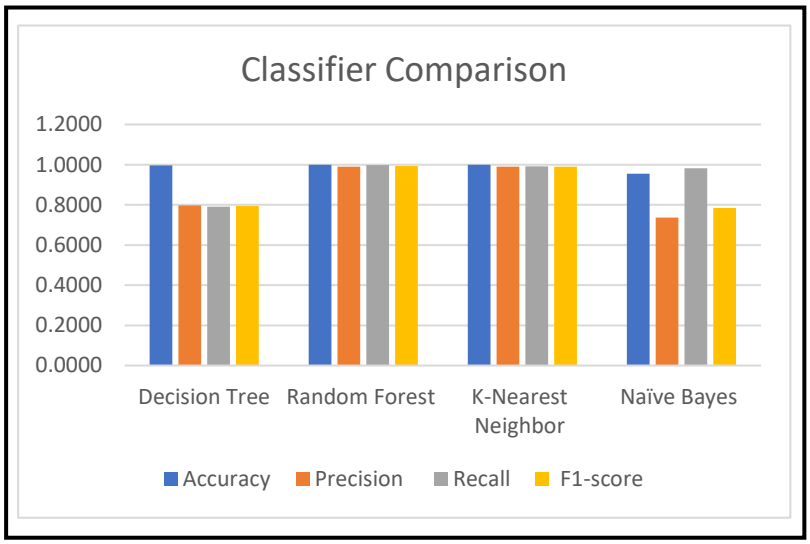

Figure 9: Comparison between classification techniques with RFE

Tables 2, 3, 4, 5, and 6 and the corresponding Figures present the comparison of Decision Tree, Random Forest, KNN, and Naïve Bayes algorithms. The results show that Random Forest performed best, achieving an accuracy of 0.9998, a precision of 0.9894, a recall of 0.9975, and an F1-score of 0.9934. In contrast, Naïve Bayes recorded the lowest performance, with an accuracy of 0.9547, precision of 0.7368, recall of 0.9820, and an F1-score of 0.7844.

5 Conclusion

In this study, supervised machine learning models were evaluated for network traffic attack classification using Recursive Feature Elimination (RFE) to enhance feature selection and model efficiency. The findings demonstrate that RFE significantly improves classification performance by reducing redundant attributes and focusing on the most relevant features. The results indicate that the Random Forest classifier achieved the highest performance, with an accuracy of 0.9998, a precision of 0.9894, a recall of 0.9975, and an F1-score of 0.9934. Conversely, the Naïve Bayes classifier recorded the lowest performance, obtaining an accuracy of 0.9547, a precision of 0.7368, a recall of 0.9820, and an F1-score of 0.7844. Future work may extend this study by incorporating deep learning architectures and hybrid models for more complex traffic scenarios.

References

[1] A. S. Dawood, "Performance evaluation of machine learning Naïve Bayes algorithms for network traffic classification," Technium Romanian Journal of Applied Sciences and Technology, vol. 13, no. 1, pp. 12–26, Jan. 2023, doi: 10.47577/technium.v13i.9473.

[2] J. Zhang, H. Guo, and J. Liu, "Adaptive task offloading in vehicular edge computing networks: A reinforcement learning-based scheme," Mobile Networks and Applications, vol. 25, pp. 1736–1745, 2020.

[3] S. S. Moustafa, G. E. A. Mohamed, M. S. Elhadidy, and M. S. Abdalzaher, "Machine learning regression implementation for high-frequency seismic wave attenuation estimation in the Aswan Reservoir area, Egypt," Environmental Earth Sciences, vol. 82, no. 307, 2023.

[4] O. Hamdy, H. Gaber, M. S. Abdalzaher, and M. Elhadidy, "Identifying exposure of urban area to certain seismic hazard using machine learning and GIS: A case study of greater Cairo," Sustainability, vol. 14, no. 10722, 2022.

[5] Y. Zhao, Y. Li, X. Zhang, G. Geng, W. Zhang, and Y. Sun, "A survey of networking applications applying the software defined networking concept based on machine learning," IEEE Access, vol. 7, pp. 95397–95417, 2019.

[6] S. Namasudra, P. Lorenz, and U. Ghosh, "The new era of computer network by using machine learning," Mobile Networks and Applications, vol. 28, pp. 764–766, 2023.

[7] M. S. Abdalzaher, M. S. Soliman, S. M. El-Hady, A. Benslimane, and M. A. Elwekeil, "Deep learning model for earthquake parameters observation in IoT system-based earthquake early warning," IEEE Internet of Things Journal, vol. 9, no. 11, pp. 8412–8424, 2022.

[8] E. Salazar, C. A. Azurdia-Meza, D. Zabala-Blanco, S. Bolufé, and I. Soto, "Semi-supervised extreme learning machine channel estimator and equalizer for vehicle-to-vehicle communications," Electronics, vol. 10, no. 968, 2021.

[9] I. H. Sarker, "Machine learning: Algorithms, real-world applications and research directions," SN Computer Science, vol. 2, no. 160, 2021.

[10] R. Ahmed, S. Islam, S. Shatabda, and A. K. M. Muzahidul, "Intrusion detection system in software-defined networks using machine learning and deep learning techniques – A comprehensive survey," TechRxiv Preprint, 2021, doi: 10.36227/techrxiv.17153213.v1.

[11] K. S. B., "Supervised machine learning: A review of classification techniques," Informatica, vol. 31, pp. 249–268, 2007.

[12] D. Lowd and P. Domingos, "Naïve Bayes models for probability estimation," in Proc. Int. Conf. Machine Learning (ICML), pp. 529–536, 2005.

[13] P.-Y. Zhou and K. C. C. Chan, "A model-based multivariate time series clustering algorithm," in PAKDD 2014: Trends and Applications in Knowledge Discovery and Data Mining, pp. 805–817, 2014.

[14] H. Zhang and J. Yan, "Performance of SDN routing in comparison with legacy routing protocols," in Proc. Int. Conf. Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), pp. 491–494, 2015.

[15] T. Kohonen, "Essentials of the self-organizing map," Neural Networks, vol. 37, pp. 52–65, 2013.

[16] D. H. Lee, D. Y. Kim, and J. I. Jung, "Multi-stage intrusion detection system using hidden Markov model algorithm," in Proc. Int. Conf. Information Science and Security (ICISS), pp. 72–77, 2008.

[17] T. Hurley, J. E. Perdomo, and A. Perez-Pons, "HMM-based intrusion detection system for software defined networking," in Proc. IEEE Int. Conf. Machine Learning and Applications (ICMLA), pp. 617–621, 2016.

[18] T. Guelzim and M. S. Obaidat, Formal Methods of Attack Modeling and Detection. Amsterdam, Netherlands: Elsevier, 2015.

[19] D. E. Moriarty, A. C. Schultz, and J. J. Grefenstette, "Evolutionary algorithms for reinforcement learning," Journal of Artificial Intelligence Research, vol. 11, pp. 241–276, 1999.

[20] C. J. C. H. Watkins and P. Dayan, "Q-learning," Machine Learning, vol. 8, pp. 279–292, 1992.

[21] A. Azab, M. Khasawneh, S. Alrabaee, and K. R. Choo, "Network traffic classification: Techniques, datasets, and challenges," Digital Communications and Networks, vol. 10, no. 3, pp. 676–692, 2021, doi: 10.1016/j.dcan.2022.09.009.

[22] M. Mahmood, S. A. Khalil, S. J. Shah, and M. Ahmad, "Network traffic classification techniques and comparative evaluation of machine learning models," Int. J. Innov. Sci. Res. Technol., vol. 8, no. 6, pp. 1135–1139, Jun. 2023, doi: 10.5281/zenodo.8098744.

[23] H. Canever, X. Wang, and U. Learning, "Network traffic classification using unsupervised learning: A comparative analysis of clustering algorithms," HAL Open Science Preprint, 2023.

[24] S. Hossen, T. Ahmad, M. Aidiel, and R. Putra, "Traffic classification with machine learning for enhancing cloud security," in Proc. Int. Conf. Intelligent Methods, Systems and Applications (IMSA), pp. 86–91, 2023, doi: 10.1109/IMSA58542.2023.10217598.

[25] M. Shafiq, X. Yu, and A. A. Laghari, "Network traffic classification techniques and comparative analysis using machine learning algorithms," in Proc. Int. Conf. Computer Communication and Systems (ICCCS), 2016, doi: 10.1109/CompComm.2016.7925139.

[26] C. S. Priya, B. Thulasi, V. Yashaswini, and P. Shivani, "IoT network traffic classification using machine learning algorithms," Int. J. Eng. Technol. Res. Manage., no. 05, pp. 535–539, 2025.

[27] K. Jerabek, J. Luxemburk, R. Plny, J. Koumar, J. Pesek, and K. Hynek, "When simple model just works: Is network traffic classification in crisis?" arXiv preprint, arXiv:2506.08655, Jun. 2025.

[28] R. H. Serag, M. S. Abdalzaher, H. A. E. Atty Elsayed, and M. Sobh, "Software defined network traffic classification for QoS optimization using machine learning," Journal of Network and Systems Management, vol. 33, article 41, Feb. 2025.

[29] Z. Wang, K. W. Fok, and V. L. L. Thing, "Machine learning for encrypted malicious traffic detection: Approaches, datasets and comparative study," Computers & Security, vol. 113, art. 102589, 2022.

[30] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in Proc. 4th Int. Conf. Inf. Syst. Security Privacy (ICISSP), Madeira, Portugal, pp. 108–116, 2018.

[30] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in Proc. Int. Conf. Information Systems Security and Privacy (ICISSP), pp. 108–116, 2018.

[31] R. Panigrahi and S. Borah, "A detailed analysis of CICIDS2017 dataset for designing intrusion detection systems," International Journal of Engineering and Technology, vol. 7, no. 3.24, pp. 479–482, 2018