Available online at www.qu.edu.iq/journalcm
JOURNAL OF AL-QADISIYAH FOR COMPUTER SCIENCE AND MATHEMATICS
ISSN:2521-3504(online) ISSN:2074-0204(print)

An Empirical Study of Machine Learning Algorithms for Network Flow Anomaly Detection

Hasanain Mohammed Manji Al-Rzoky[1]

General Directorate of Education in Babylon, Ministry of Education, Iraq. hasanain.comp86@gmail.com

ARTICLEINFO

Article history:
Received: 23 /12/2025
Rrevised form: 20 /02/2026
Accepted : 26 /02/2026
Available online: 30 /00/2026

Keywords:

Anomaly Detection

Network Log Data

Cybersecurity

Machine Learning

Intrusion Detection Systems

ABSTRACT

Cyberattacks have increased significantly, while networks are becoming more complex and difficult to defend; therefore, there is a pressing need to find intelligent ways to identify and respond to abnormal activity threatening information security. Current systems can identify abnormalities based on static and repetitive network behavior that do not apply to the constantly changing nature of the volatile Information Technology (IT) environment we operate in today. This study uses Artificial Intelligence (AI) and Machine Learning (ML) to develop an anomaly-detection model from Network Log Data to help bridge this gap. This study utilized a complete dataset with several important network-flow characteristics (ports, protocols, duration of flows, number of packets in each flow, length of packets, labels of flows (normal or malicious)) that are commonly found in network log data. Preprocessing of data included filling-in missing values, converting categorical variables into numeric format, and eliminating all illogical records. In order to test the ability to evaluate the performance of the developed model, the total dataset was split equally between training (70%) and testing (30%). Four Machine Learning algorithms (Random Forest, Support Vector Machine, K-Nearest Neighbors and Decision Trees) were applied to distinguish between normal and abnormal network behaviors. The results of the experiments indicated that the aforementioned models provide good accuracy and efficiency in dealing with large and diverse datasets. Overall, this research aims to provide a systematic comparison of machine language models within the context of a network dataset for analyzing network logs and identifying anomalies, thereby contributing to the improvement of various cybersecurity systems and supporting proactive defense strategies for networks. The results will offer a real-world example of the performance tradeoffs associated with different models to assist organizations in making informed decisions when selecting an appropriate model for their specific network environment.

https://doi.org/ 10.29304/jqcsm.2026.18.22517

1. Introduction

As we now have an abundance of threats through various types of cyberattacks on networks as well as developments of Internet technology; Network Intrusion Detection has been an increasingly significant area of study. Thus, the need to develop and implement more effective anomaly detection systems, so as to protect network integrity, has also increased. Anomaly detection is a method of detecting and preventing anomalous behaviors (or patterns) within network traffic, and could represent possible security threats. Due to the inherent characteristics of anomalies, it is not easy to find a solution to this anomaly identification problem. Defining precisely what constitutes "normal" versus "abnormal" or "anomalous" behavior within the context of a computer network is equally as complex. Additionally, since many anomaly detection methods require the classification of "normal" versus "deviant" behaviors; obtaining this classification is difficult [1], [2].

Also, selecting appropriate tools to identify anomalies is also complicated and that the suggested apparatus will not fit all possible situations, but only for one particular type of anomaly. Therefore, it is fairly reasonable to assume that choosing an anomaly detection method is not easy when the types of anomalies are not known in advance. The

size of the network is another problem. When identifying a fault, consider fault tolerance (the ability of a system to continue working even if any of its component components fail) and load balancing (the process of distributing execution tasks among multiple network servers to improve overall performance), especially as the size of the current network increases [3], [4]. The detection of breaches in networks is still a major problem in the realm of cyber security, due to the changing nature of attacks on computer networks and the complexity of traffic in networks. Although machine learning presents attractive solutions to these issues, it is still the case with problems such as unbalanced data, the complexity of features, and the efficiency of models. The rise of AI technologies has reinforced AI's role as an effective tool in the pursuit of improving accuracy and effectiveness of anomaly detection systems [5], [6].

Four techniques in AI will be used to detect anomalies in networks and try to remedy that anomaly. Firstly, one of the techniques known as the random forest is the regression tree technique, which uses automatic aggregation and random distribution of predictions to achieve a high degree of predictive accuracy [7], [8]. In this research, we rely on the creation of a large number of decision trees and the integration of their results using the voting method. This method reduces the risk of overfitting and provides high accuracy in classifying complex data. Then, will turn to vector machine (SVM) technology, which is considered one of the most important and effective data separation techniques, and is the best choice to use, especially when the given data is ambiguous. This technique was invented in 1951 for discriminatory testing, when determining probability densities using parametric estimation was relatively difficult [9], [10]. In this research, the multi-category classification function was trained using the fitcecoc function. It relies on finding the optimal boundary split between categories using an RBF kernel (kernel function), which allows for the differentiation of nonlinear data.

Besides, K-Nearest Neighbors (KNN) is based on the idea that the patterns closest to the x′ target pattern we're looking to name provide useful nomenclature information. KNN assigns class naming to the majority of the styles closest to K in the data space. For this purpose, we should be able to determine a scale of similarity in the data space [11], [12]. In this research, the number of neighbors (K = 5) was selected, and based on this, each flow was classified based on the five closest samples of training data. This technique is suitable for classifying data with convergence. Moreover, the Decision Tree technology will be used, which is one of the most successful learning techniques, due to its various attractive features such as simplicity, ease of understanding, no transactions, and the ability to handle mixed types of data. In this technique, the decision tree is induced from a set of named training cases, represented by a set of attribute values and class naming [13], [14].

In this research, we will build a decision tree based on dividing the data according to the most discriminatory values of the categories. This method is easy to understand and interpret and is used as the basis for many other models. There has been a lot of literature on trying to address anomaly for networks and attacks, but using different techniques or perhaps using one of the techniques used in this research.

A study [15] addresses the challenge of detecting anomalies in networks of intelligent manufacturing systems that have become more complex and interconnected with the Fourth Industrial Revolution. It proposes an AI-based approach and reverse engineering of protocols to enable direct network data analysis without the need for specialized site knowledge. The method targets non-exploratory operating environments and employs external signatures, temporal information, and time interval patterns to identify abnormal behavior. It can find irregularities regardless of what information has been encoded into its manufacture protocols.

The paper analyzes security issues in IoT networks, edge computing and cloud computing, specifically as it relates to attacks that result in broken links and route diversion. The paper presents SDN-ADS (Software Defined Network Anomaly Detection System), an SDN based anomaly detection system that detects malicious behavior on the part of IoT devices and edge computing, and TA-Edge (Trusted Authority Model for Edge Computing), a trusted authority model for edge computing that enables efficient certificate verification. The simulations demonstrated that both systems performed well regarding security, routing efficiency and network stability [16]. The study [17], Provides a framework for anomaly detection in video big data for smart cities of the future based on artificial intelligence of things (AIoT), technologies using dual stream neural network, to perform real time detection on resource constrained IoT devices and deep analysis via cloud computing; also uses spatiotemporal and visual features across the BD-LSTM network to classify types of anomalies such as assault or abuse. The experimental results on the UCF-Crime and RWF-2000 datasets showed that proposed model was superior than the proposed model with the increase in accuracy of 9.88% and 4.01% compared to the latest methods.

This article [18] a review of current techniques for finding anomalies in network information based on artificial intelligence (AI) and machine learning (ML). This study compared supervised and unsupervised ML and Deep

Learning approaches to determine which were most useful and what the benefits and drawbacks of each were in terms of anomaly detection. The results from the analysis showed that neural networks such as Recurrent Neural Networks (RNN) and Convolutional Neural Networks (CNN), were able to identify many different types of anomalies but also found that there are a number of significant challenges to using neural networks for anomaly detection. These include the issues of quality of input data, the need for high powered computers to run them, and that they are difficult to understand how they make their decisions. Overall, this study provides an up-to-date view of the application of AI and ML in Network Security and identifies trends and future directions in these fields.

Another study addressed the challenges of detecting network traffic anomaly as attackers create new complex forms of attacks which will reduce the effectiveness of traditional rule-based or signature-based techniques. This study developed the Trans-M model to detect anomalies in transformers and hostile generative networks through correction hashing and random data blocks to improve interference resistance and class balance. According to test results the Trans-M model when balanced had an accuracy of 98.12% a recall of 97.86% and an F1 score of 98.46%, and exhibited notable improvements over other models tested. These results show that this system has advantages over traditional systems, and can meet the real-world network security protection requirements [19]. A review of other studies describes a process for detecting and categorizing anomalies within a network through the use of an artificial neural network (ANN), which is applied to NetFlow data. In addition, this research includes training a multi-layer connected (MLP) artificial neural network with a back-propagation (reverse propagation) algorithm. The experiment included both actual ISP data and modified dataset (simulating cyber-attacks). The results demonstrated that the proposed artificial neural network was able to identify multiple anomaly types efficiently [20].

A study [21] examined the problems associated with the anomaly detection problem in encrypted Internet traffic. The study was a systematic examination of artificial intelligence (AI) techniques for detecting anomalies on encrypted traffic. This study analyzed 30 studies based on the researchers' specific evaluation criteria, including but not limited to; datasets, feature extraction, algorithm selection and performance indicators. The research findings suggested that some of the methodologies applied to detect anomalies in encrypted traffic are comparable to those applied to unencrypted traffic. However, the research also demonstrated that some methodologies have been developed especially to address challenges created by encryption.

The study [22] examines the use of machine learning classification to identify anomalies in NetFlow data flow. In the study, four machine learning classifiers (RF, SVM, K-NN, and AdaBoost) were tested on the UNSW-NB15 dataset. The researchers investigated the effects of varying ratios between training and testing data sets, method of feature coding and reducing the amount of features. The study's results indicated that the RF classifier achieved the highest F2 score (97.68%) and Area Under Curve (AUC) (98.47%). Additionally, the study indicated that the RF classifier performed faster when utilizing Label Encoding to encode the labels compared to One-Hot encoding. Furthermore, the study indicates that its contribution lies in the improvement of the machine learning process by applying appropriate feature selection and coding techniques to optimize the performance of machine learning in unbalanced data environments.

The study [23] assesses the most recent and effective methods for identifying anomalies in network traffic due to an increase in security threats. The study assessed both traditional approaches such as statistical methods, machine learning, deep learning, and behavior-based methods, while assessing the benefits and drawbacks of each approach and the applicability of each approach to real-world network security scenarios. The study identified the efficacy and value of hybrid approaches to anomaly detection and quantitatively compared the effectiveness of different approaches to anomaly detection in various network security contexts. Finally, the study predicted future trends in anomaly detection, which include intelligent systems, automated processes and interpretability, and emphasized the need for inter-disciplinary collaboration and the development of large-scale data collection systems to improve network security and the development of active immunity systems.

This study [6] is examining the application of Artificial Intelligence and Deep Learning for identifying Anomalies on Modern Communication Networks, in order to overcome the limitations associated with Traditional Rule-Based Methods for Anomaly Detection. This study provides a survey of how Anomaly Detection Techniques have progressed from traditional anomaly detection methodologies to AI-driven methodologies with respect to challenges, applications of algorithms, and real-world case studies. Also, this study identifies new technologies; specifically, Generative Adversarial Networks (GAN) and Reinforcement Learning; which can improve Anomaly Detection Capabilities through 5G/6G, Edge Computing, and the Internet of Things. Finally, this study recommends Hybrid Models, Advanced Pre-Processing and Self-Adaptive Systems in order to increase the robustness and reliability of systems and enable proactive Anomaly Management.

Study [24] discusses the role of anomaly detection in network traffic as a major component of Network Security against increasing complexity of Cyber Threats. The study also evaluated the effectiveness of several types of Machine Learning Models (e.g., Isolation Forest, Naive Bayes, XGBoost, LightGBM, and Support Vector Machine) to address two of the main problems of using anomaly detection in network traffic: class imbalance and high dimensional space (i.e., a large number of feature values). The study found that models like XGBoost and LightGBM were very effective at detecting anomalies in network traffic while other models (e.g., Isolation Forest) were less effective. While the study provided empirical evidence to support which models are most effective in detecting anomalies in network traffic, the study also identified the relative advantages and disadvantages of each model so users can select models which will improve their ability to detect cyber threats.

Study [25] examines the efficiency of Intrusion Detection Systems (IDS) in Cyberspace and identifies significant drawbacks to using traditional Signature-Based IDSs for addressing evolving threats in the future. This study uses Network Behavior Analysis (NBA), as an alternative method to signature-based systems, by focusing on the analysis of the behavior of hosts and/or applications to identify patterns that indicate the presence of anomalous or potentially malicious activity. Also, conducted a comprehensive review of existing literature from January 1, 2014 to April 30, 2024 to evaluate the methodologies used, data sets, types of attacks evaluated, performance metrics used to assess the ability of these IDSs to detect various attacks, and the limitations and challenges currently associated with NBA-based IDSs. Overall, the research provides a foundation for the development of more flexible and capable IDSs using more advanced artificial intelligence technologies, and is of interest to both researchers and practitioners who seek to create more effective and robust IDS systems.

Unlike studies that focus on algorithmic development, this study emphasizes empirical comparison using machine learning techniques to identify anomalous patterns in data received through the network from an anomaly detection model. The study examines five machine learning approaches with the primary purpose of evaluating the complexity of each approach and determining how effective they are in network security. Ultimately, this study will link the use of time representations to deep learning models within the area of anomaly detection, and create a practical and feasible method for enhancing the reaction of security teams. Likewise, this study will also serve as a reference for future research in related areas of study, contribute to the advancement of technology, and its applications. Additionally, this research will provide valuable information, and methodologies that can assist in solving network anomaly problems.

2. Research Methodology

This study utilized machine learning techniques for identifying network flows that contain cyberattacks using the MATLAB software programming environment. The performance of four different types of algorithms were compared: Random Forest, Support Vector Machine (SVM) and Decision Trees as well as K-Nearest Neighbor (KNN). Additionally, a description is given of how the experimental process was executed and analyzed for all steps including data preparation and analysis of model results with respect to the performance of each algorithm.

2.1. Dataset Preparation

This research uses a dataset with around eighty properties taken from network flows. Properties are comprised of time scale, statistics of the packet length, indices of TCP tags, properties of the flow rate, and advanced properties of TCP windows. The Label property is used as the source of information for labeling normal traffic and traffic attacks. Pre-processing was done in the following steps:

1. Removed all missing values using the rmmissing function.
2. Converted all text variables into numerical variables by using categorical and double.
3. Identified the most relevant variable or group of variables affecting the classification outcome.
4. Splits the data into 70 % training data and 30 % test data, using cvpartition.

After deleting missing values, and converting the text values to numerical values, the data was evaluated. There were five lines (out of 1,048,575) which contained some logical errors in some of the properties, e.g., FlowDuration and FlowIATMean. As they were so few lines, these lines have been ignored, it has been noted that due to their low number, their impact on the performance of the different classification models will be minimal and inconsequential. The process of creating the prepared data is shown in **Fig. 1**.

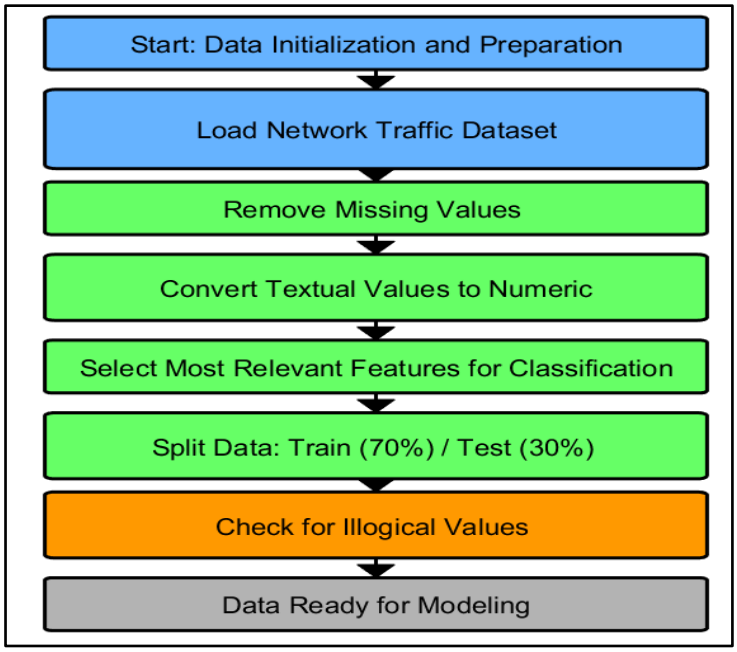

Fig. 1- Diagram Stages of Data Preparation and Setup Before the Training Process

2.2. Presentation and Analysis of the Characteristics of the Dataset Used

The Network Traffic Logs dataset that is being utilized for analysis in order to identify and categorize data flow as normal or malicious. Each row represents a network flow (single), while the multiple column attributes are the descriptive behaviors associated with the flow from different perspectives. These characteristics can be classified into main groups as follows:

1. Basic flow information (DstPort: Destination port number, Protocol: Protocol type (e.g., TCP, UDP, ICMP), Timestamp: Flow start time, FlowDuration: Flow duration in microseconds).
2. Forward and backward packet statistics (Number of packets: TotFwdPkts and TotBwdPkts, Packet lengths: FwdPktLenMax, FwdPktLenMin, FwdPktLenMean, FwdPktLenStd, and their reverse counterparts, Total packet lengths: TotLenFwdPkts and TotLenBwdPkts).
3. Flow rate and inter-arrival times (FlowByts_s, FlowPkts_s: Byte and packet rate per second, FlowIATMean, FlowIATStd, FlowIATMax, FlowIATMin: Packet interval statistics for each flow, Separate statistics for forward and backward directions: FwdIATTot, FwdIATMean, FwdIATStd, FwdIATMax, FwdIATMin, and BwdIATTot, etc).
4. TCP Flags and Special Flags (FINFlagCnt, SYNFlagCnt, RSTFlagCnt, PSHFlagCnt, ACKFlagCnt, URGFlagCnt, ECEFlagCnt, CWEFlagCount, Flags specific to forward and backward directions such as FwdPSHFlags, BwdPSHFlags, FwdURGFlags, and BwdURGFlags).
5. Other Advanced Properties (Packet Header Length (FwdHeaderLen, BwdHeaderLen), Statistics of time intervals for activity and inactivity (ActiveMean, ActiveStd, ActiveMax, ActiveMin, IdleMean, IdleStd, IdleMax, IdleMin), Average packet and data size per flow (PktLenMean, PktLenStd, PktLenVar, PktSizeAvg, FwdSegSizeAvg, BwdSegSizeAvg)).
6. Final flow classification (The Label column indicates whether the flow is normal (Benign) or represents some type of attack. The majority of records in this sample appear to be normal flows). The **Fig. 2** shows hierarchical structure of network traffic dataset features.

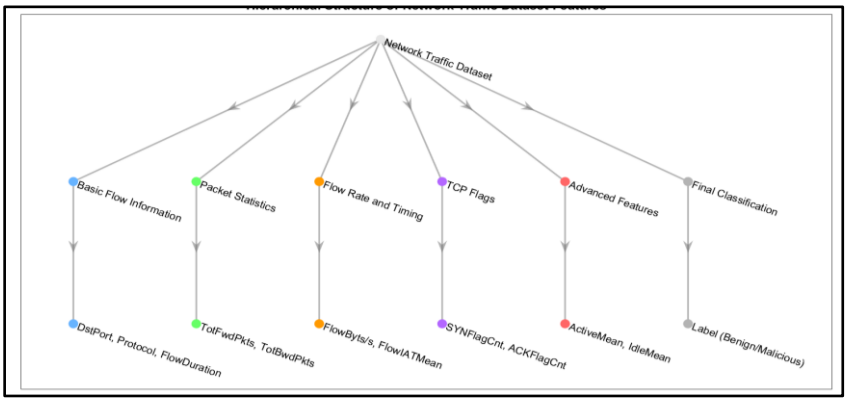

Fig. 2- Hierarchical Structure of Network Traffic Dataset Features

2.3. Exploratory Procedures

The preliminary exploratory data analysis (EDA) was conducted with Matlab, using the following EDA steps:

1. head(data); The head () function is used to show the first eight rows of the data to check the number of variables (columns), and the values of those variables.
2. summary(data); The summary () function is applied to get an overall statistical description of each variable (column), including:

   - Minimum, maximum, and mean values,
   - std.dev. (standard deviation),
   - number of "missing" entries per variable,
   - class frequency in text columns (e.g., Label).

Thus, the EDA will provide an insight into the data's distribution, outliers, and possible holes prior to commencing the preprocessing and/or model building phases.

2.4. System Design

The Attack Detection System was developed by combining four key components; Work Environment, Inputs, Analytical Algorithms, Outputs as follows:

First: Work Environment
- Software Used (MATLAB R2024a, Microsoft Excel)
- Hardware Used (Operating System: Windows 11, Random Access Memory (RAM): 8 GB, Processor (CPU): Intel Core i5)

Second: Inputs
Input Data Files - Data files in .CSV format containing Network Flow Logs, that include specific attributes of the Data Packets and the Protocols being utilized.

Third: Algorithms Used
Four key algorithms were applied to analyze input data to identify malicious activity:
- Random Forest utilizing the fitcensemble function.
- Support Vector Machine (SVM) utilizing the fitcecoc function.
- K-Nearest Neighbors (KNN) utilizing the fitcknn function.
- Decision Trees utilizing the fitctree function.

Fourth: Outputs
Each Network Flow is classified into either one of the two categories:
- Normal (Benign)
- Attack (Attack)

The **Fig. 3** illustrates the system design process stages.

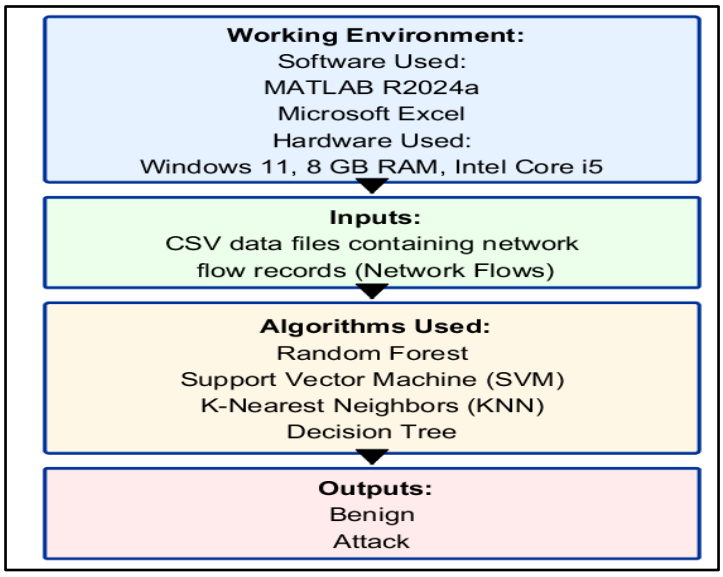

Fig. 3- System Design Stages

2.5. Feature Analysis and Categorization

To better understand the structure of the data, and to make more methodical decisions about how to analyze it, all the features derived from the network flow were grouped into categories based on what they represent in terms of the way people use networks. There is a large amount of temporal data in this dataset. This includes the duration of individual flows (flow_duration), and metrics related to inter arrival times (inter-arrival time, IAT). Inter-arrival time metrics include the average number of milliseconds between each packet in an individual flow, and the average number of milliseconds between each packet in both the forward and backward directions within a single flow. Temporal metrics are important because they allow for the analysis of how traffic patterns evolve over time; and because many types of anomalies will have an unusual temporal pattern.

There is also a large amount of statistical data in this dataset. This includes metrics related to the size of individual packets (packet_length), and the total number of packets sent in the forward direction and in the backward direction. The statistical data allows for the analysis of the distribution of packet sizes, and can be used to identify when there is a significant departure from expected values.

Finally, there are structural and directional features in this dataset. Structural features relate to the structure of the communication process. Directional features relate to how much data is sent in one direction versus another. Examples of structural features include the total number of packets sent in the forward direction and in the backward direction, and the sum of the headers of the packets sent in the forward and backward directions. An example of a directional feature is the Down/Up ratio, which indicates the proportion of bytes received at the receiver versus the proportion of bytes transmitted by the sender. Behavioral and flag-based features (e.g., SYN, ACK, RST, PSH, URG, ECE, and FIN flag counts) relate to the specific state and actions of connections and sessions. This information can be useful in determining whether an activity is malicious or deviant.

These categories show that this dataset represents many dimensions and is of high dimensionality, and therefore, there is a need to test multiple approaches to classify in order to see how best to take advantage of the variety of features.

2.6 Classification Models and Selection Rationale

The study applies four supervised classifications (Decision Tree DT, Random Forest RF, Support Vector Machines SVM, and K-Nearest Neighbor KNN) to classify normal versus abnormal network traffic. The choice of the models was based on both theoretical diversity and the structure of the data used in the study.

Theoretical diversity: The selected classifiers represent different learning paradigms. Decision Trees are an example of a rule-based, interpretable model that can generate non-linear decision boundaries. Random Forest is an extension of this paradigm using ensemble learning, which enhances generalizability and reduces overfitting when dealing with large amounts of data. Support Vector Machines represent a margin-based learning paradigm that effectively constructs optimal separating hyperplanes in high dimensional feature space. K-Nearest Neighbor, being a distance-based and instance-based classifier, detects local similarities in the data.

Data driven diversity: With 1,048,575 records in the database, there are only 5 records that contain nonsensical values in the features, thus indicating that the data has high quality and low levels of noise. There are also several dimensions of features associated with each record including temporal, statistical, structural, and behavioral aspects of network flows, suggesting that there could be many complexes, and possibly non-linear, relationships between the features of the data. Therefore, tree-based models were chosen because they can capture non-linear interactions between features and SVM was included because it is known to be effective in high dimensional feature space. KNN was included to assess whether the classifier would be able to detect local anomalies (i.e., localized anomaly patterns).

By combining models from multiple different methodological families under the same experimental design, the study provides a comprehensive and balanced assessment of how well the various classification models perform at detecting anomalies in network flows.

3. Result and Discussion

3.1. Training and Testing Strategy

Once we finished the data processing and the preparation of the feature set most influential for the attack detection, we moved to the phase of modeling training and testing, which aims at teaching the algorithms how to identify attacks from normal flows by analyzing the statistical patterns in the network traffic characteristics. To accomplish this, the dataset has been split into two major datasets:

- Training Dataset (70%): It contains all the data of the original dataset that are utilized to train the models on all the types of attacks as well as on normal behavior.
- Test Dataset (30%): It contains all the data of the original dataset that are utilized to test the models on new data that have never been seen by the models in order to guarantee an objective assessment of their performance.

3.2 Class Distribution Analysis

Before model training, the class distribution was examined. The dataset contained three classes: Benign traffic, FTP-BruteForce, SSH-BruteForce. Although the dataset is not perfectly balanced, the imbalance level is moderate rather than extreme. To minimize potential bias toward the majority class, macro-averaged evaluation metrics were computed in addition to overall performance measures.

3.3 Model Evaluation

To create the training and testing datasets, MATLAB's cvpartition function was employed using a stratified hold-out approach to ensure a random yet proportionally balanced distribution of classes between subsets. Four machine learning classification algorithms were trained on the networks' flow data to classify it:

- Random Forest
- Support Vector Machine (SVM)
- K-Nearest Neighbors (KNN)
- Decision Tree

Each model was then independently tested against the test set. Using the predict () function in MATLAB, each model was able to produce predicted classifications for each test case; after which, confusion matrices were generated to assess how well each model classified data across all classes. The performance of each model was also assessed using the same four primary evaluation metrics:

- Accuracy
- Precision
- Recall
- F1-score

3.4 Mathematical Formalism

To represent rigorously using mathematical language the methodologies for machine learning that were employed in this research. The table (1) uses the below mathematical representations to define both the classification algorithms and the metrics by which they are evaluated. The incorporation of formalized mathematical definitions of these components provides a high degree of methodological clarity and allows for specific interpretations of how each model behaves with regard to the intrusion detection paradigm.

Table 1- Mathematical Representation of Machine Learning Models and Evaluation Metrics

Model / Metric	Mathematical Representation
Random Forest	$h_i(x)\ \sum_{i=1}^{N} \frac{1}{N} = H(x)$
Decision Tree	$(_mR \ni c_m I(x\ \sum_{m=1}^{M} = f(x)$
Support Vector Machine (SVM)	$0 = b + x \cdot w$ Optimization: (\min \frac{1}{2}
K-Nearest Neighbors (KNN)	$(c = {}_i\arg\max_c \sum_{,i\in N} I(y = \hat{}y$
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$
Precision	$\frac{TP}{TP+FP}$
Recall	$\frac{TP}{TP+FN}$
F1-Score	$\frac{Precision \times Recall}{Precision+Recall} \times 2$

3.5 Performance Analysis

An assessment of variation in Accuracy, Precision, Recall and F1-score of the classification models was carried out through a graphical comparative study. Additionally, an examination of confusion matrices was used to analyze class level performance and misclassifications trends.

The macro-averaged metrics for each of the evaluation criteria were calculated to allow for a fair comparison among the classification models and to determine which classification model was the best at intrusion detection. The results indicated that the models performed differently as far as discriminability is concerned when detecting network traffic patterns, and there were substantial differences among the models in their ability to distinguish between attack types.

Results of performance are shown in Table (2). The results show that the Decision Tree and Random Forest performed significantly better than SVM and KNN. Specifically, the Decision Tree model had the highest Accuracy (0.98184) and F1-Score (0.97383) with only a slight difference from Random Forest. In addition, Random Forest

demonstrated high and consistent performance on all evaluation criteria. This indicates that tree-based models are very successful in finding non-linear relationships within structured features of network flows.

Table 2- Comparative performance analysis between models

Model	Accuracy	Precision	Recall	F1 Score
Random Forest	0.98105	0.97231	0.97343	0.97279
SVM	0.84547	0.79049	0.79871	0.76527
KNN	0.88517	0.86971	0.81989	0.80275
Decision Tree	0.98184	0.97574	0.97196	0.97383

On the other hand, SVM and KNN performed moderately. Specifically, SVM demonstrated good Precision and Recall for the Benign Class; however, SVM had difficulty distinguishing between FTP-BRUTEFORCE and SSH-BRUTEFORCE traffic. Furthermore, KNN demonstrated good discrimination between Normal and Malicious Traffic; however, KNN had considerable confusion between the two BRUTEFORCE attack categories.

Following training of the Random Forest model on the dataset utilized and analysis of the classification results through an ambiguity matrix, the results are presented in **Fig. 4**.

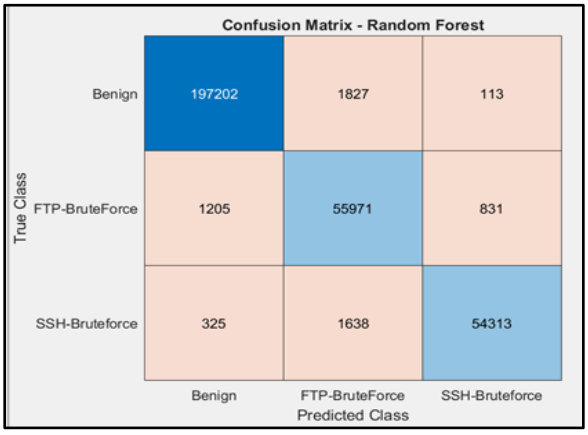

Fig. 4- Confusion Matrix of the RF Model

The confusion matrix of the Random Forest model presents a high level of accurate classification throughout the three types of traffic: Benign, FTP-BruteForce, and SSH-BruteForce. The model produced large amounts of true positives for all three classes; Benign (197,202), FTP-BruteForce (55,971), and SSH-BruteForce (54,313) which is indicative of the ability of the Random Forest model to accurately detect both normal and malicious traffic. Misclassifications were minimal when compared to the total amount of data analyzed and are indicative of the robust nature of the Random Forest model due to its use of an ensemble-based approach. There was however some confusion found between the two BruteForce classes (FTP-BruteForce and SSH-BruteForce), this is likely due to the similarity in the traffic patterns of these two classes. Overall, the results provided further support for the reliability and stability of the Random Forest models for use in intrusion detection systems. To evaluate the ability of the SVM model to classify network flow traffic using the ambiguity matrix, the ambiguity matrix shown in **Fig. 5** was created.

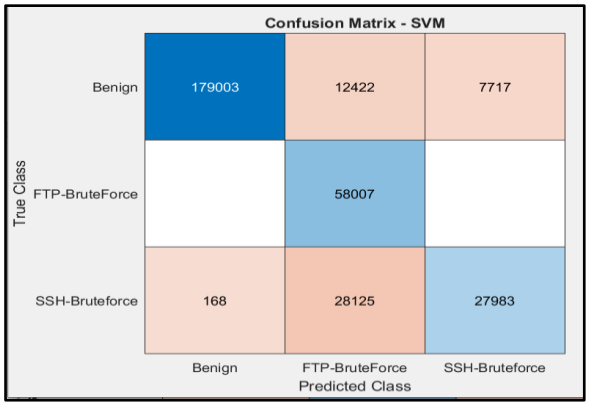

Fig. 5- Confusion Matrix of the SVM Model

The SVM model was able to produce a high rate of correct classifications of Benign traffic (179,003 samples) while also producing significant amounts of incorrect classifications of benign flows into either FTP-BruteForce (12,422) or SSH-BruteForce (7,717) indicating sensitivity in distinguishing between normal and attack traffic. In addition to the incorrect classifications of benign flows into either of the two BruteForce categories, the SVM model also incorrectly classified a significant amount of FTP-BruteForce flows into SSH-BruteForce category. Specifically, 28,125 SSH samples were misclassified as FTP. This misclassification of similar brute-force attack traffic patterns indicates that the decision boundaries established by the SVM model regardless if they are linear or kernel-based do not have sufficient detail to distinguish between structurally similar traffic patterns. To show how well the K-Nearest Neighbors (KNN) model performs at classifying network records. The confusion matrix illustrated in **Fig. 6** represents the confusion matrix that resulted from testing the model.

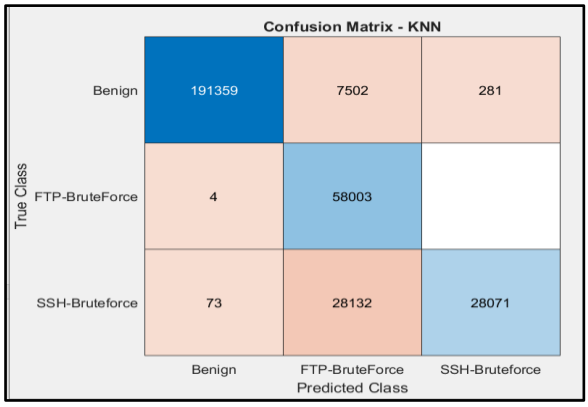

Fig. 6- Confusion Matrix of the KNN Model

The ambiguity matrix illustrates that The KNN model performed well in distinguishing Benign and FTP-BruteForce samples, resulting in a high correct classification rate in each category. However, there is significant confusion between SSH-BruteForce and FTP-BruteForce traffic, primarily because the KNN model has incorrectly classified SSH samples as FTP. These results are consistent with the fact that distance-based classifier performance is highly sensitive to the degree of overlap in the feature space. Because KNN is based solely upon the proximity of data points within a feature space, a large degree of overlap in statistical distribution among similar types of attacks will greatly decrease the ability of a classifier to discriminate between the classes. Therefore, parameter optimization (i.e., finding the optimal value of k), or dimensionality reduction algorithms could potentially increase the separation between the classes and improve the overall performance of the model. The **Fig. 7** illustration displays the performance of the decision tree algorithm in distinguishing between normal behavior and the two types of attacks (FTP-BruteForce and SSH-BruteForce) using an ambiguity matrix.

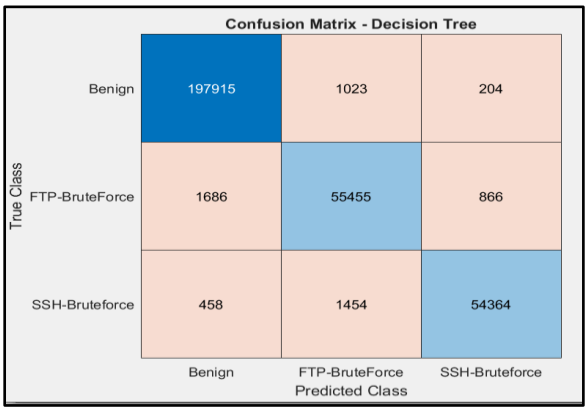

Fig. 7- Confusion Matrix of the Decision Tree Model

Decision Trees performed better than any other model examined in this study. The confusion matrix also supports the idea that the Decision Tree model exhibits strong generalization capabilities, and effectively separates normal from malicious traffic. The hierarchical nature of decision trees allows for the creation of clearly defined decision boundaries, which is particularly beneficial when examining structured network flow data.

3.6 Threat Detection Implications in Real-World Environments

The data collected from the experiments indicate that tree-based algorithms (Decision Trees & Random Forest) are capable of identifying brute force attacks within network traffic with significant success. Due to their high classification accuracy and consistent macro-average performance they have the potential to be utilized as part of future IDS systems. Practical cybersecurity systems rely on models which can operate effectively within high levels of traffic and varied attack signatures. The ability to identify both FTP-BruteForce and SSH-BruteForce attacks demonstrates the capability of tree-based algorithms to classify small differences in the statistical characteristics of features from network flows. Additionally, the lower complexity associated with tree-based models when compared to more complex deep learning architectures allows for the possibility of utilizing them for real time monitoring systems.

In addition to the advantages previously mentioned the moderate level of class imbalance present in the data set did not impact the reliability of the model's performance due to the utilization of stratified sampling and macro-average performance metrics. Therefore, the results provided do accurately represent the model's performance regardless of the proportion of each type of traffic in the training data. Although the model is able to differentiate between FTP-BruteForce and SSH-BruteForce traffic there is an important limitation to consider: attacks with similar behavior may require additional feature development or the implementation of a hybrid detection system to better distinguish between the two.

Therefore, it appears that classical machine learning methods are still highly competitive for structured network flow analysis and offer an attractive alternative to traditional deep learning-based methods for developing efficient, interpretable and deployable operational cybersecurity systems.

4. Conclusion

The four machine learning models used in this study produced a variety of levels of performance in regard to classification between legitimate and malicious network traffic. All models were able to identify network attacks, however each had differences in terms of accuracy, consistency, and ability to generalize. While both the Decision Tree and K-Nearest Neighbor (KNN) models performed acceptably well, each had shortcomings in dealing with feature interaction complexity and class overlap. In addition, while the Support Vector Machine (SVM) was able to produce good detection across classes, it struggled with separating very similar classes of attacks from one another.

Of the methodologies studied, the Random Forest model showed the best results overall in terms of both accuracy and consistency in detection, and thus is most suitable for intrusion detection. The ensemble-based approach of

Random Forest allows the combination of many decision trees to improve generalization and reduce the impact of variability, making the method more reliable for classification. These findings show the potential of ensemble learning for use in network security, and that future enhancements using hybrid or other advanced learning methods may provide even better detection.

In conclusion, these studies confirmed that machine learning-based intrusion detection systems are capable of differentiating between normal and malicious traffic. However, to be effective against emerging cyber threats and evolving attacks, continued improvements will be needed in both the development of models and the selection of features.

5. Limitations and Future Research Directions

This research used the provided dataset to assess the ability of intrusion detection based on machine learning models. Although this dataset is extensive and includes many different aspects of network traffic and attack scenarios there are some limitations. The dataset was generated in an experimental lab environment which does not completely replicate the nature of real-world network environments. In addition, to being highly variable, real-world networks have dynamic traffic patterns and new and evolving methods of attacks that do not exist in all lab created datasets. Therefore, the performance of intrusion detection models in live environments could potentially be significantly different than their performance in the lab.

Additionally, while the classes were relatively well distributed in the dataset, it should be noted that there is still some degree of imbalance remaining in the distribution of the classes in the dataset. Moreover, the overlapping of the features associated with different classes of attacks (for example FTP-BruteForce and SSH-BruteForce) could also impact the performance of the classification process. Therefore, it is clear that there is a need for additional research into advanced feature extraction and imbalance mitigation techniques.

Finally, this study only examined a limited subset of possible attack types and network features. The use of larger, more varied, datasets in future studies will increase the generality of the models developed and increase the practicality of the models developed for intrusion detection. However, despite these limitations, the conclusions drawn from this research are beneficial for understanding intrusion detection based on machine learning and show promise for the development of enhanced security defenses for the protection against cyber threats through the use of ensemble and hybrid models.

The research presented here will be expanded upon by developing hybrid and ensemble-based methods for anomaly detection, to improve the robustness and generality of the classification processes. Future research will investigate the implementation of various advanced feature extraction techniques to generate more discriminatory network attributes, reduce class overlap and improve the interpretability of the models developed. In addition, future research will implement hyperparameter optimization using systematic approaches such as grid search and cross validation to improve the stability and performance of the models developed. Further, to assess the performance of the models developed in dynamic network environments and against a variety of attack scenarios, future studies will utilize operational datasets obtained from the real world.

In addition, the research presented here will be extended to explore the capabilities of deep learning architectures and automated feature learning mechanisms for the detection of complex and novel attack patterns. It is expected that these approaches will provide increased capabilities for detecting complex and novel attack patterns. As has been the case throughout this research, transparency and interpretability of the models developed will continue to be major concerns to ensure that the models developed can be trusted and deployed in practice for use in cybersecurity systems.

6. References

[1] W. Abbass, N. Abbas, U. Majeed, W. Nawaz, Q. Abbas, and A. H. Farooqi, “A Cyber Resilient Framework for V2X Enabled Roundabouts in Intelligent Transportation Systems,” *IEEE Access*, 2025, doi: 10.1109/ACCESS.2025.3604095.

[2] A. Hussain, M. Yasir, N. Iqbal, S. Tripura, and A. Aslam, “Heart Disease Classification Using Machine Learning Techniques: An Impact Analysis of GridSearchCV-based Optimization,” in *2025 5th International Conference on Emerging Smart Technologies and Applications (eSmarTA)*, IEEE, 2025, pp. 1–8. doi: 10.1109/eSmarTA66764.2025.11132291.

[3] K. R. Ahmed, R. S. Shammah, S. Kowser, O. Faruq, M. A. Sufian, and M. R. Ahmmed, "Strengthening Digital Security in MIS: A Business Analytics Approach to Deepfake Detection," in *2025 International Conference on Quantum Photonics, Artificial Intelligence, and Networking (QPAIN)*, IEEE, 2025, pp. 1–6. doi: 10.1109/QPAIN66474.2025.11171733.

[4] N. A. A. Taleb *et al.*, "New Approach for Network Threat Detection and Prevention Using Real-time Data Analysis and Deep Learning," in *2025 5th International Conference on Emerging Smart Technologies and Applications (eSmarTA)*, IEEE, 2025, pp. 1–7. doi: 10.1109/eSmarTA66764.2025.11132116.

[5] Y. Zhang, R. C. Muniyandi, and F. Qamar, "A Review of Deep Learning Applications in Intrusion Detection Systems: Overcoming Challenges in Spatiotemporal Feature Extraction and Data Imbalance," *Applied Sciences*, vol. 15, no. 3, p. 1552, 2025, doi: 10.3390/app15031552.

[6] E. Edozie, A. N. Shuaibu, B. O. Sadiq, and U. K. John, "Artificial intelligence advances in anomaly detection for telecom networks," *Artif Intell Rev*, vol. 58, no. 4, p. 100, 2025, doi: 10.1007/s10462-025-11108-x.

[7] H. A. Salman, A. Kalakech, and A. Steiti, "Random forest algorithm overview," *Babylonian Journal of Machine Learning*, vol. 2024, pp. 69–79, 2024.

[8] S. J. Rigatti, "Random forest," *J Insur Med*, vol. 47, no. 1, pp. 31–39, 2017.

[9] M. Bansal, A. Goyal, and A. Choudhary, "A comparative analysis of K-nearest neighbor, genetic, support vector machine, decision tree, and long short term memory algorithms in machine learning," *Decision analytics journal*, vol. 3, p. 100071, 2022, doi: 10.1016/j.dajour.2022.100071.

[10] E. Fix, *Discriminatory analysis: nonparametric discrimination, consistency properties*, vol. 1. USAF school of Aviation Medicine, 1985.

[11] O. Kramer, "K-nearest neighbors," in *Dimensionality reduction with unsupervised nearest neighbors*, Springer, 2013, pp. 13–23. doi: 10.1007/978-3-642-38652-7_2.

[12] S. B. Imandoust and M. Bolandraftar, "Application of k-nearest neighbor (knn) approach for predicting economic events: Theoretical background," *Int J Eng Res Appl*, vol. 3, no. 5, pp. 605–610, 2013.

[13] J. Su and H. Zhang, "A fast decision tree learning algorithm," in *Aaai*, 2006, pp. 500–505.

[14] A. Navada, A. N. Ansari, S. Patil, and B. A. Sonkamble, "Overview of use of decision tree algorithms in machine learning," in *2011 IEEE control and system graduate research colloquium*, IEEE, 2011, pp. 37–42. doi: 10.1109/ICSGRC.2011.5991826.

[15] H. Kim and T. Shon, "Industrial network-based behavioral anomaly detection in AI-enabled smart manufacturing," *J Supercomput*, vol. 78, no. 11, pp. 13554–13563, 2022, doi: 10.1007/s11227-022-04408-4.

[16] K. N. Qureshi, G. Jeon, and F. Piccialli, "Anomaly detection and trust authority in artificial intelligence and cloud computing," *Computer Networks*, vol. 184, p. 107647, 2021, doi: 10.1016/j.comnet.2020.107647.

[17] W. Ullah *et al.*, "Artificial Intelligence of Things-assisted two-stream neural network for anomaly detection in surveillance Big Video Data," *Future Generation Computer Systems*, vol. 129, pp. 286–297, 2022, doi: 10.1016/j.future.2021.10.033.

[18] V. P. PM and S. Soumya, "Advancements in anomaly detection techniques in network traffic: The role of artificial intelligence and machine learning," *Journal of Scientific Research and Technology*, pp. 38–48, 2024, doi: 10.61808/jsrt114.

[19] H. Cao, "The Detection of Abnormal Behavior by Artificial Intelligence Algorithms under Network Security," *IEEE Access*, 2024, doi: 10.1109/ACCESS.2024.3436541.

[20] S. Andropov, A. Guirik, M. Budko, and M. Budko, "Network anomaly detection using artificial neural networks," in *2017 20th conference of open innovations association (FRUCT)*, IEEE, 2017, pp. 26–31. doi: 10.23919/FRUCT.2017.8071288.

[21] I. H. Ji, J. H. Lee, M. J. Kang, W. J. Park, S. H. Jeon, and J. T. Seo, "Artificial intelligence-based anomaly detection technology over encrypted traffic: A systematic literature review," *Sensors*, vol. 24, no. 3, p. 898, 2024, doi: 10.3390/s24030898.

[22] I. Fosić, D. Žagar, K. Grgić, and V. Križanović, "Anomaly detection in NetFlow network traffic using supervised machine learning algorithms," *J Ind Inf Integr*, vol. 33, p. 100466, 2023, doi: 10.1016/j.jii.2023.100466.

[23] W. Zhang and J. P. Lazaro, "A survey on network security traffic analysis and anomaly detection techniques," *International Journal of Emerging Technologies and Advanced Applications*, vol. 1, no. 4, pp. 8–16, 2024, doi: 10.62677/IJETAA.2404117.

[24] S. Ness, V. Eswarakrishnan, H. Sridharan, V. Shinde, N. V. P. Janapareddy, and V. Dhanawat, "Anomaly Detection in Network Traffic using Advanced Machine Learning Techniques," *IEEE Access*, 2025, doi: 10.1109/ACCESS.2025.3526988.

[25] M. Janati and F. Messaoudi, "Intrusion Detection System-Based Network Behavior Analysis: A Systemic Literature Review," *International Journal of Advanced Computer Science and Applications*, vol. 16, no. 3, 2025, doi: 10.14569/IJACSA.2025.0160378.