Journal of Computer Science

Research Article

Intelligent Intrusion Detection System Using RF, SVM, and DT: A Comparison-Based KDD Data Set

[1]Walaa Hassan Elashmawi, [2]Alaa Sheta and [3]Ahmad Al-Qerem

[1]*Department of Computer Science, Faculty of Computer Science, Misr International University, Cairo, Egypt*
[2]*Computer Science Department, Southern Connecticut State University, New Haven, CT, United States*
[3]*Computer Science Department, Faculty of Information Technology, Zarqa University, Zarqa, Jordan*

Article history
Received: 12-12-2024
Revised: 22-02-2025
Accepted: 07-04-2025

Corresponding Author:
Walaa Hassan Elashmawi
Department of Computer Science,
Faculty of Computer Science, Misr
International University, Cairo,
Egypt
Email:
walaa.hassan@miuegypt.edu.eg

Abstract: The rapid growth of technology has brought about many advantages, but has also made networks more susceptible to security threats. Intrusion Detection Systems (IDS) play a vital role in protecting computer networks against malicious activities. Given the dynamic and constantly evolving nature of cyber threats, these systems must continuously adapt to maintain their effectiveness. Machine Learning (ML) methods have gained prominence as effective tools for constructing IDS that offer both high accuracy and efficiency. This study conducts a performance assessment of several machine learning classifiers, including Random Forests (RF), Decision Trees (DT), and Support Vector Machines (SVM), in addressing multiclass intrusion detection as a means to counter cybersecurity threats. The NSL-KDD dataset, which includes various network attacks, served as the basis for our experimental evaluation. The research explores two classification scenarios: a five-class and a three-class model, analyzing their impact on detection performance. The results demonstrate that RF consistently achieves the highest accuracy (85.42%) on the three-class scenario testing set, highlighting its effectiveness in handling patterns and non-linear relationships within the intrusion data. Furthermore, reducing the classification complexity (three classes vs. five classes) significantly improves model generalization, as evidenced by the reduced performance gap between training and testing data. Friedman's rank test and Holm's post-hoc analysis were applied to ensure statistical rigor, confirming that RF outperforms DT and SVM in all evaluation metrics. These findings establish RF as the most robust classifier for intrusion detection and underscore the importance of simplifying classification tasks for improved IDS performance.

Keywords: Intrusion Detection System, Random Forest Classifier, Multi-Class Classifications

Introduction

The proliferation of interconnected digital systems has significantly enhanced the efficiency and accessibility of information. However, this interconnectedness has also made networks vulnerable to various malicious activities. Intrusion Detection Systems (IDS) have emerged as a critical component of network security, designed to identify and mitigate unauthorized access attempts (Hossain and Islam, 2023). Intrusion detection systems can be categorized into Misuse Detection Systems (MID) and Anomaly Detection Systems (AID) based on their analytical approaches (Axelsson, 2000; Alkasassbeh and Al-Haj Baddar, 2023). MID, or signature-based detection, identifies threats by matching predefined patterns of potentially harmful activities or operations stored in a database. However, AID monitors the behavior of the network and triggers alerts when deviations from predefined norms occur. (Gong *et al.*, 2009). Furthermore, they can be classified as network-based (NIDS) or host-based (HIDS) depending on their data source (Mishra *et al.*, 2019). HIDS mostly examines system calls and process IDs of the operating system data.

In contrast, NIDS investigates network-related events, such as traffic volume, IP addresses, service ports, and the protocols used (Jyothsna and Prasad, 2019). The most effective intrusion detection systems can detect new attacks quickly and take the necessary actions. Although perfect accuracy is difficult to achieve,

SCIENCE Publications

researchers continue to work to improve the accuracy of IDS.

In addition, IDS is a significant factor in the market's dramatic rise. Effective IDs are in high demand due to the growing importance of data protection and compliance among enterprises. The global distribution of the Intrusion Detection And Prevention System (IDPS) industry (Business Research Insights, 2024) is shown in Figure 1. The market size is projected to grow significantly and is predicted to be USD 6.96 billion in 2024 to USD 11.99 billion by 2029, exhibiting a cumulative annual growth rate (CAGR) of 11.5%.

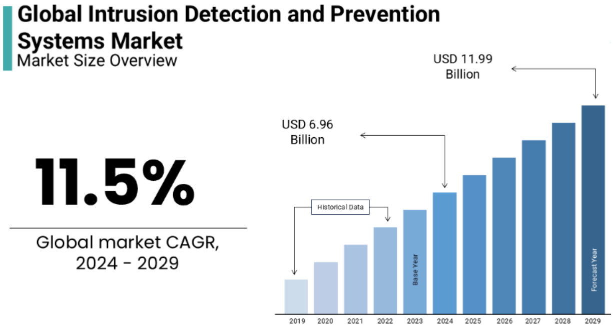


Fig. 1: Market Size Overview of Global IDPS (2024-2029) (Business Research Insights, 2024)

The growing demand for IDS is driven by factors such as the increased cyber threats associated with remote work and cloud adoption, and the advancements in Artificial Intelligence (AI) and Machine Learning (ML), as a result of which better IDS solutions have been developed.

Conventional intrusion detection systems frequently employ predefined patterns or anomaly detection methods. Although effective against recognized threats, these approaches may face challenges in identifying innovative and complex attacks (Dini *et al.*, 2023).

In addition, conventional methods have problems with computational inefficiencies in large-scale networks, high false positive rates, and missing new threats. Prior studies also have focused on binary classification (attack vs. normal traffic) or individual classifiers, neglecting a thorough comparative analysis of multi-class intrusion detection models.

To address these challenges, ML techniques have gained prominence in recent years in various domains (Sheta *et al.*, 2024; Elashmawi *et al.*, 2024a). ML algorithms can analyze vast network data to learn patterns and anomalies indicative of malicious behavior, enabling IDS to adapt to evolving threats. In the study by Tait *et al.* (2021), the effectiveness of current machine learning algorithms was analyzed, as their potential use in enhancing the present IDS against emerging intrusion attacks. Furthermore, ML models, especially classification algorithms, can be trained to classify attacks into different types. This classification helps security analysts quickly understand the nature of the attack and respond accordingly (Elashmawi *et al.*, 2024b), especially for novel or zero-day attacks that do not have known signatures. Also, there is limited research on how simplifying classification tasks (e.g., reducing the number of classes) affects model performance and generalization capabilities.

This study addresses existing gaps by evaluating the impact of class reduction on detection performance, applying statistical validation techniques (Friedman's and Holm's tests), and identifying the most effective ML classifier for intrusion detection using the NSL-KDD dataset. To achieve this, we utilize machine learning-based intrusion detection systems to identify attacks, focusing on the AID. Among the most effective ML techniques, Random Forest (RF), Decision Tree (DT), and Support Vector Machine (SVM) are applied to the NSL-KDD dataset, a commonly used standard for assessing IDS. We determine the most effective approach for enhancing intrusion detection accuracy and generalization through rigorous testing and comparative analysis. While the primary focus of IDS lies in monitoring network traffic and identifying malicious activities, related applications in cybersecurity, such as phishing detection (Ishtaiwi *et al.*, 2024), have demonstrated the efficacy of machine learning techniques in identifying threats.

Continuing with the paper's framework, a concise review of relevant literature on intrusion detection systems is presented. The subsequent section examines the ML classifiers employed in the investigation and offers a thorough examination of the NSL-KDD dataset that was employed. Then, it presents the developed model and the evaluation metrics employed to assess its performance. Following this, the empirical findings from evaluating various ML-based intrusion detection approaches for multiclass classification are discussed. Finally, the paper discusses key considerations and insights from the study.

Related Works

Intrusion detection is a critical pillar in the architecture of network security. Nevertheless, the heterogeneous landscape of intrusion detection and prevention systems frequently grapples with limitations in operational performance and overall efficiency. An IDS's success relies on accurately detecting network anomalies without generating excessive false alarms. Researchers have suggested employing ML classification algorithms to overcome the challenges faced by intrusion detection systems in terms of accuracy and performance.

Data mining has emerged as a standard to evaluate network intrusion detection technologies alongside popular machine learning techniques such as Extreme Learning Machines (ELM) (Ahmad *et al.*, 2018). Almseidin *et al.* (2017) conducted and analyzed many experiments to evaluate different machine learning classifiers using the KDD intrusion dataset through the WEKA toolbox. Gao *et al.* (2019) proposed an adaptive ensemble learning model utilizing the NSL-KDD dataset and incorporating RF, DT, and Deep Neural Network (DNN) algorithms with an accuracy of 85.2%. In the study by Alaketu *et al.* (2024), identifying intrusions on the complete and reduced CIC-IDS2017 datasets was accomplished with the help of SVM, DNN, and RF. The RF has achieved superior results in both cases, reaching up to 90.6% (whole dataset) and 90.9% (reduced dataset).

Sheta and Alamleh (2015) employed three machine learning algorithms — DT, SVM, and Multi-Layer Perceptron (MLP) to tackle intrusion detection tasks. The models were evaluated using the DARPA training dataset, with feature selection carried out via Best First Search and Genetic Search methods to enhance classification performance.

However, in the study by Agarwal *et al.* (2021), the authors examine the accuracy levels and optimal model fit of three trustworthy machine learning classification algorithms, such as SVM, K-Nearest Neighbor (KNN), and Na¨ıve Bayes (NB). The models were trained using the UNSW-NB15 dataset. The same dataset assesses various ML models in the study by Dini *et al.* (2023). The study conducted by Bitra *et al.* (2024) utilizes the KDDCup99 dataset to evaluate various ML models, including KNN, SVM, RF, and LightGBM. Among these, the RF achieved the highest accuracy, reaching 98.74%. Moreover, in Al-Daja *et al.* (2023), various ML classifiers are analyzed for intrusion detection systems and highlight that RF outperforms the other classifiers.

In training models, regularization is integrated with artificial neural networks (ANN) to categorize and identify abnormalities in the study by Albahar *et al.* (2020). They tested their model on the 20% of NSL-KDD training set for some reasons, such as the large NSL-KDD dataset being training on the entire dataset being computationally expensive. Eshak Magdy *et al.* (2022) investigates the NSL-KDD dataset through the lens of several intrusion detection systems. According to Halimaa and Sundarakantham (2019), an ML model utilizing the SVM technique is proposed. The model's performance was assessed using the NSL-KDD dataset, resulting in an accuracy of 93.95%.

A hybrid model was developed in Aljawarneh *et al.* (2018). The authors employed J48, Random Tree, and Na¨ıve Bayes to predict the degree of intrusion scope threshold using data from network transactions. Using the NSL-KDD data set, they were able to obtain 99.81% accuracy with the binary dataset and 98.56% accuracy with the multiclass dataset, both of which included 20% testing data. However, Bhattacharya *et al.* (2020) introduced a hybrid ML model (XGBoost) that combines principal component analysis (PCA) and the firefly algorithm to classify the IDS dataset. The hybrid model is implemented to perform dimensionality reduction, followed by applying the XGBoost algorithm to classify the transformed dataset.

In another study by Hossain and Islam (2023), various ensemble techniques and publicly available datasets are used for intrusion detection. Compared to other approaches, the Random Forest methodology consistently achieves FPR and accuracy levels above 99%. Despite notable progress in developing ML-based IDS for enhancing network cybersecurity, the field continues to face unresolved research challenges, particularly in efficiently managing and processing large-scale datasets.

Materials and Methods

This section outlines the methodology for the proposed ML-based intrusion detection system, covering the ML classifiers used, the NSL-KDD dataset, the system workflow from preprocessing to prediction, and the evaluation metrics applied to measure model performance.

Machine Learning Classifiers

A variety of ML algorithms have been designed to detect anomalies within network traffic, aiming to distinguish malicious behavior from regular activity. These methods typically compare network traffic to a baseline of normal behavior. Anomaly detection in machine learning is categorized into supervised, unsupervised, and semi-supervised approaches. This study focuses on supervised methods using labeled data and highlights standard ML algorithms for multi-class intrusion detection.

Random Forest (RF)

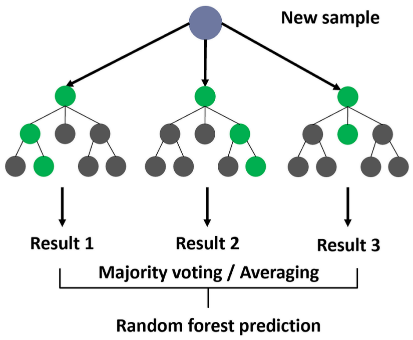


Fig. 2: RF classifier

RF is a robust ML algorithm that addresses classification and regression tasks. It generates an ensemble of decision trees, each trained on distinct data subsets to enhance predictive accuracy and robustness (Breiman, 2001). By combining predictions from several decision trees, RFs enhance generalization performance and decrease the likelihood of overfitting, as shown in Fig. 2. This ensemble approach helps to reduce bias and improve prediction accuracy.

Decision Trees (DT)

The interpretability and ability to provide light on the underlying data distribution make DTs popular (Quinlan, 1986) and widely used in ML. Decision Trees are easy to grasp and intuitive because of their tree-like structure, as seen in Figure 3. They divide the feature space using a sequence of binary decisions. They comprise decision nodes, which split the data based on feature values, and leaf nodes, which contain the final predictions or labels.

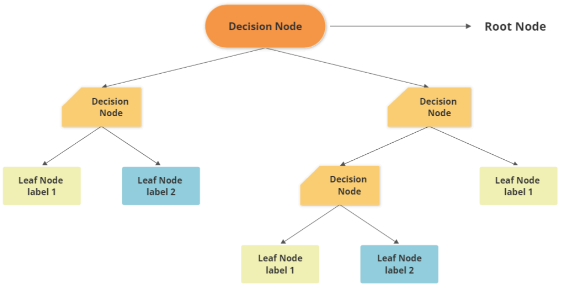


Fig. 3: DT classifier

Support Vector Machines (SVM)

At its core, the algorithm seeks to construct the most discriminative hyperplane in a high-dimensional space, ensuring a precise separation of classes — a concept visually represented in Figure 4. SVM can be used for binary and muti-class classifications. The authors in (Xu *et al.*, 2017) suggested a multiclass SVM model that maximizes the separation between training examples and the hyperplane. SVMs can handle non-linear relationships using kernel functions to map data into a higher-dimensional space (Ghosh *et al.*, 2019).

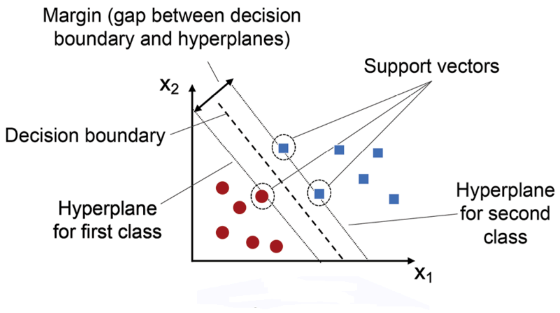

Margin (gap between decision boundary and hyperplanes)
Support vectors
x_2
Decision boundary
Hyperplane for first class
Hyperplane for second class
x_1

Fig. 4: SVM classifier

NSL-KDD Dataset Description

The Network Security Laboratory Knowledge Discovery and Data Mining (NSL-KDD) Dataset (Tavallaee *et al.*, 2009) is utilized in this study. It serves as a commonly adopted benchmark for assessing the effectiveness of IDS. The dataset comprises a comprehensive network traffic record set encompassing normal and attack activities. It is partitioned into separate training and testing subsets, enabling researchers to develop and assess their models on distinct data portions.

The NSL-KDD dataset encompasses a broad spectrum of attack types, including denial-of-service (DoS), probing, user-to-root (U2R), remote-to-local (R2L), as well as regular traffic. This diversity renders it a valuable benchmark for evaluating the robustness and generalizability of IDS models. It had 43 variables, such as packet length, protocol type, and source/destination IP addresses, that contained numerical and categorical data and a labeled field indicating the type of attack. The distribution of instances across all categories is detailed in Table 1. A thorough examination and understandable description of the NSL-KDD dataset are provided by (Dhanabal and Shantharajah, 2015).

Table 1: The distribution of the NSL-KDD's primary attack types

Attack Category	Total instances
dos	45927
normal	67343
probe	11656
r2l	995
u2r	52

The dataset exhibits a significant imbalance, with some classes significantly under-represented compared to others. Including the least two classes, which account for only a small fraction of the total instances, could heavily bias the classifier toward these minority classes, leading to poor generalization of the more prevalent classes. This poses a considerable challenge in training a classifier.

Therefore, this study focuses on detecting well-represented attack types (e.g., dos, probes, and normal ones). These classes represent the most critical attack types in real-world IDS and are more likely to be encountered in practice. By concentrating on these classes, the model's performance on the more prevalent attack types is optimized, which is crucial for real-world applications of intrusion detection systems. For the most focused class, the authors (Baleev *et al.*, 2024) concentrated on the DoS attack, in which unauthorized users ”sniff” the network to find the weak spots of a particular resource in a probing attack and dos attacks, the goal is to overwhelm network services with excessive data packets to render them inaccessible (Bhattacharya *et al.*, 2020).

Using all five classes, including the minority ones, might lead to a trade-off in model performance due to the difficulty in adequately learning from the highly imbalanced data. By excluding the least represented classes, we can enhance the classifier's focus on distinguishing between the more common attack types, enhancing the model's accuracy regarding the majority classes.

ML-Based Intrusion Detection System

ML-based IDSs have emerged as powerful tools for safeguarding networks against a constantly evolving landscape of cyber threats. These systems examine network data using advanced algorithms to spot unusual patterns that might be signs of malicious activity. In contrast to signature-based IDS that depend on predetermined criteria, ML-IDS is able to learn from past data and identify both known and new attacks.

Figure 5 below illustrates a deep understanding of the workflow of the proposed ID classification model and the evaluation criteria for assessing it.

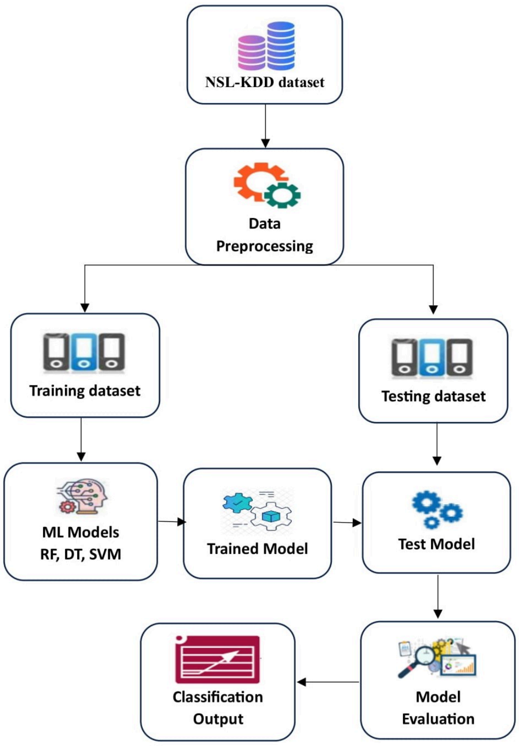


Fig. 5: The workflow of the ML-based multiclass ID classification

Model Development

Intrusion detection, a critical task in network security, presents unique challenges for machine learning algorithms. Accurate classification of malicious activities often requires a multi-step data preparation process. This process may involve data collection, cleaning, scaling, and partitioning (into training and testing sets) before applying the ML model. For example, Figure 5 visually represents the complete workflow of the ML-based multiclass ID classification on the NSL-KDD dataset.

- Data input: This phase involves loading the NSLKDD dataset (i.e., labeled data) into the ML system and preparing it for analysis.
- Data preprocessing: Effective data preprocessing is a critical component of data analysis that directly impacts the accuracy of predictions. Several techniques can be applied, such as removing duplicates, one-hot encoding (Matos *et al.*, 2022), and data standardization. One-hot encoding is a method that helps transform categorical input into a format that ML algorithms can understand—data standardization to ensure that different features contribute equally to the model's learning process.
- Training and Testing datasets: The dataset should represent real-world network traffic, encompassing a diverse range of normal and abnormal activities. The training dataset is used to train the ML model, while the testing dataset evaluates its performance on unseen data.
- ML classifier models: constructing prediction models for datasets where RF, DT, and SVM were the three machine learning classifiers utilized in this step. The model is trained by exposing it to the training data, allowing it to learn the underlying patterns and relationships. During training, the model undergoes iterative adjustments to minimize the discrepancy between its predicted outputs and the actual labels in the training dataset.
- Model evaluation: Evaluate the model's ability to generalize patterns to new data. Several metrics, including the confusion matrix and accuracy, may be applied to evaluate the machine learning models that are being utilized.
- Classification outputs: Once the model is trained and tested by interpreting unfamiliar network traffic from the test set, it generates informed predictions about the most likely attack types associated with each instance. These predictions can be in the form of probabilities or class labels. The output of the classification process is crucial for identifying potential intrusions and triggering appropriate response actions.

Evaluation Metrics

Several evaluation metrics may be used to evaluate the performance of the utilized ML-based ID models according to the actual and predicted results (Japkowicz, 2006). These metrics are predominantly composed of four distinct factors: True Positive (T+), False Negative (F-), False Positive (F+), and True Negative (T-), as illustrated in Table 2.

Table 2: The four distinct factors

Actual	Predicted	Terminology	Definition
+	+	True Positive (T+)	Correctly predicted + instance
-	-	True Negative (T-)	Correctly predicted instance
+	-	False Negative (F-)	Incorrectly predicted - instance
-	+	False Positive (F+)	Incorrectly predicted + instance

Accuracy (*Acc*): The ratio of accurately classified instances to the overall number of predictions (i.e., *Total*).

$$Acc = \frac{(\mathrm{T}+)+(\mathrm{T}-)}{\mathrm{Total}} \quad (1)$$

Precision (*P*): The count of instances predicted as belonging to the positive class.

$$P = \frac{(\mathrm{T}+)}{(\mathrm{T}+)+(\mathrm{F}+)} \quad (2)$$

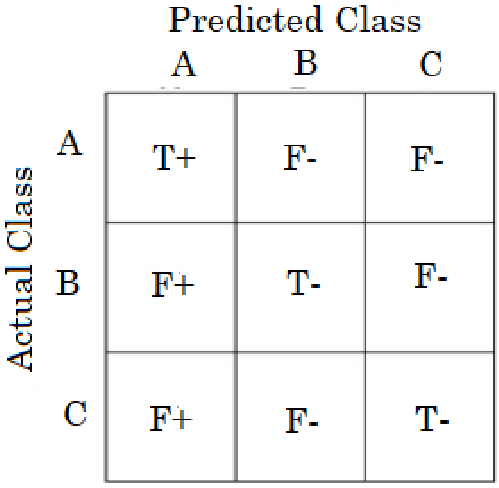
Recall (*R*): The proportion of accurately predicted instances in comparison to the total number of actual positive cases.

$$R = \frac{(\mathrm{T}+)}{(\mathrm{T}+)+(\mathrm{F}-)} \quad (3)$$

F1-score (*F −measure*): The metric quantifies the optimal balance between recall and precision.

$$F - measure = \frac{2 \times P \times R}{P + R} \quad (4)$$

Confusion Matrix (*CM*): It is a visual representation of all factors for each class as shown in Fig. 6 in the case of multiclass classification.



		Predicted Class		
		A	B	C
Actual Class	A	T+	F-	F-
	B	F+	T-	F-
	C	F+	F-	T-

Fig. 6: Visual representation of CM for multiclass classification

Results and Discussion

This section explores the experimental findings derived from implementing diverse ML models on the NSL-KDD dataset, aimed at addressing the complexities of multiclass intrusion detection. Since its introduction in 2009, the NSLKDD dataset has gained broad recognition as a benchmark standard in modern cybersecurity research. The NSL-KDD dataset comprises KDDTrain+ with 125,973 records and KDDTest+ with 22,544 records, by considering the top three classes with a total of 113,270 records in training (89.92%) and 19,590 records in testing (86.89%) with 9 and 3 duplication records in training and testing, respectively.

The experimental phase comprised training and evaluating several widely recognized ML models, such as RF, DT, and SVM. These models were selected based on their proven effectiveness in handling complex classification tasks and suitability for intrusion detection problems.

To evaluate the effectiveness of each model, a comprehensive set of performance metrics was utilized —both for the complete five-class intrusion detection scenario, as shown in Table 3, and for the top three-class setting, as detailed in Table 4.

According to Table 3, the ML models have achieved perfect results in the training while showing a substantial performance drop from the training set to the testing set, especially regarding accuracy and F-measure. RF and DT have a noticeable drop of approximately 24-25% in accuracy from training to testing, while SVM experiences a slightly smaller drop (around 24%). Therefore, using all five classes, including the minority ones, might lead to a trade-off in model performance due to the difficulty in adequately learning from the highly imbalanced data. By excluding the least represented classes, we can enhance the classifier's focus on distinguishing between the more common attack types, making the model more accurate and robust regarding to the majority classes as listed in Table 4.

Table 3: Train/Test results of ML-based IDS classification (5 classes)

ML Model	Train				Test			
	Acc	*P*	*R*	*F–measure*	*Acc*	*P*	*R*	*F–measure*
RF	99.9944	99.9944	99.9944	99.9944	75.9982	81.8537	75.9982	71.7215
DT	99.9944	99.9944	99.9944	99.9944	76.4241	81.6141	76.4241	72.6774
SVM	98.5456	98.5184	98.5456	98.5252	74.1127	73.8859	74.1127	70.2674

Table 4: Train/Test results of ML-based IDS classification (3 classes)

ML Model	Train				Test			
	Acc	*P*	*R*	*F–measure*	*Acc*	*P*	*R*	*F–measure*
RF	99.9970	99.9970	99.9970	99.9970	85.4216	86.7788	85.4216	85.1202
DT	99.9952	99.9952	99.9952	99.9952	85.3053	86.3102	85.3053	72.6774
SVM	99.0226	99.0202	99.0226	99.0207	84.5162	84.9988	84.5162	84.3678

As a result of the comparison results between five and three ID classes in training, both RF and DT show almost identical performance in both tables during the training phase, with slight improvements in accuracy, precision, recall, and F-measure when moving from the 5-class to the 3-class. SVM performs slightly better with three classes than with five classes, which indicates that the more straightforward classification task with fewer classes is more manageable for SVM to handle. The training metrics for SVM in the 5-class table were around 98.5%, while for the 3-class table, SVM achieved around 99.02%. However, in testing, RF and DT showed better performance in the 3-class compared to the 5-class in the test data. The accuracy for RF and DT increases from around 76% in the 5-class to 85% in the 3-class. SVM also performs better in the 3-class than the 5-class, with an accuracy improvement from 74.11% to 84.52% . The performance gap between training and testing data is smaller for all models in the 3-class compared to the 5-class, indicating that simplifying the classification task (fewer classes) leads to better generalization.

Furthermore, the confusion matrix for each of the utilized classifiers is shown in Figure 7, 8, and 9 for three classes. In Figure 7, the RF correctly classifies almost all ”dos,” ”normal” instances, and all ”probe” instances in training. while in testing, the RF model correctly classified 16,887 out of 19,769.

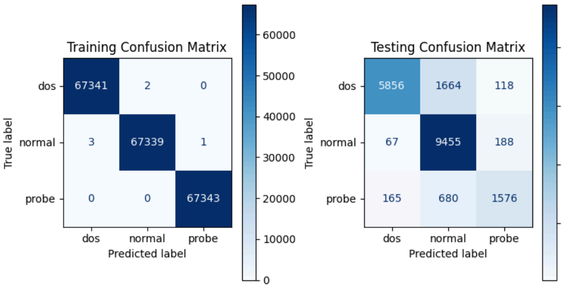


Fig. 7: RF Confusion Matrix

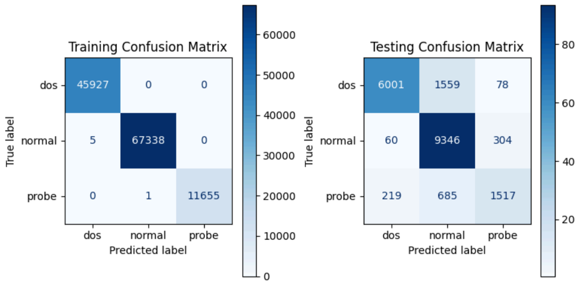


Fig. 8: DT Confusion Matrix

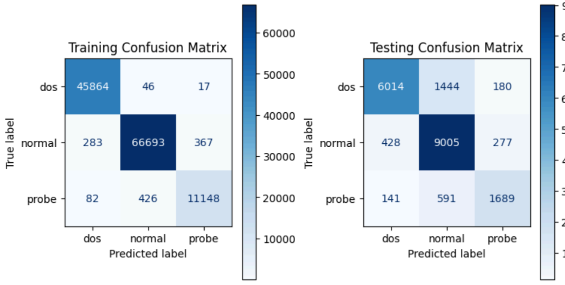


Fig. 9: SVM Confusion Matrix

The DT confusion matrix reveals a lower number of incorrect instances (6 out of 124926) in training data and (2,905 out of 19769) in testing data (as shown in Figure 8).

As shown in Figure 9, both the training and testing confusion matrices show high accuracy, indicating that the SVM model is generalizing well to unseen data. It correctly classified 123,705 instances in training and 16,708 in testing.

Furthermore, Fig. 10 shows the Receiver Operating Characteristic curves (ROC curve), for the RF classifier in 3-class ID classification. The micro-average ROC curve achieves an AUC (Area Under the Curve) of 0.95, indicating a strong overall classification performance across all classes.

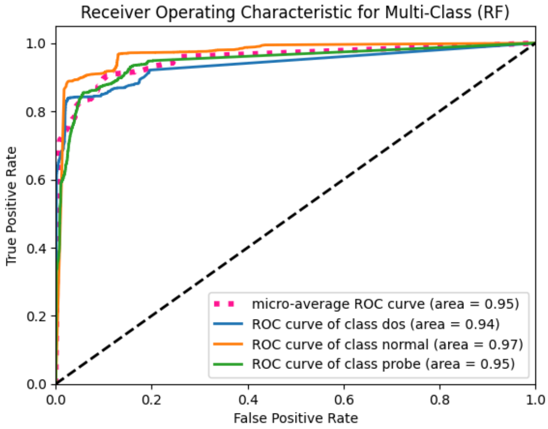


Fig. 10: RF-ROC curves for multiclass classification

These findings indicate that the RF model demonstrates strong class discrimination capabilities while sustaining a minimal F+ rate. The normal traffic class achieved the highest classification performance, with an AUC of 0.97, indicating the model’s strong capability in accurately distinguishing normal traffic from attacks activity. Also, the probe class (AUC = 0.95) and DoS class (AUC = 0.94) exhibit strong classification capability, with consistent separation from other attack types. Overall, the curves exhibit a steep rise towards the top-left corner, indicating that the model achieves high T+ rates with relatively low F+ rates.

Furthermore, Figure 11 and 12 show the ROC curves for DT and SVM models for multiclass classification of IDs. As shown in Figure 11, the micro-averaged ROC curve yields an AUC of 0.90, reflecting the model’s robust overall classification performance. Among individual classes, the ”normal” class has the highest AUC (0.90), suggesting excellent separability, while the ”dos” class follows closely with an AUC of 0.89. The ”probe” class has the lowest AUC (0.79), implying that the model struggles more in distinguishing this class than the others.

From Figure 12, the ”probe” class has the highest AUC (0.93), followed by the ”normal” class with an

AUC of 0.92, demonstrating excellent classification performance. The ”dos” class has a slightly lower AUC of 0.89 but performs well. The SVM exhibits superior performance compared to the DT model, particularly in distinguishing the ”probe” class.

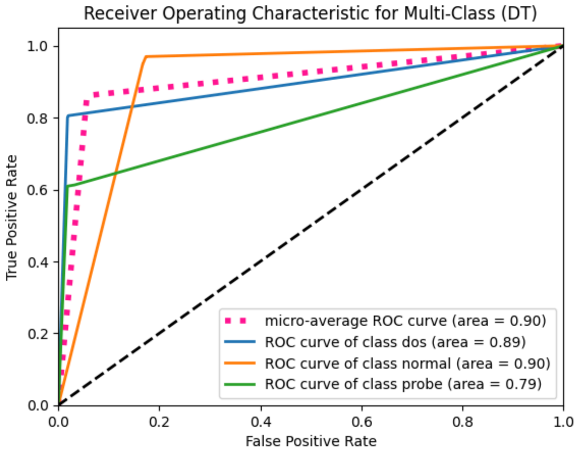


Fig. 11: DT-ROC curves for multiclass classification

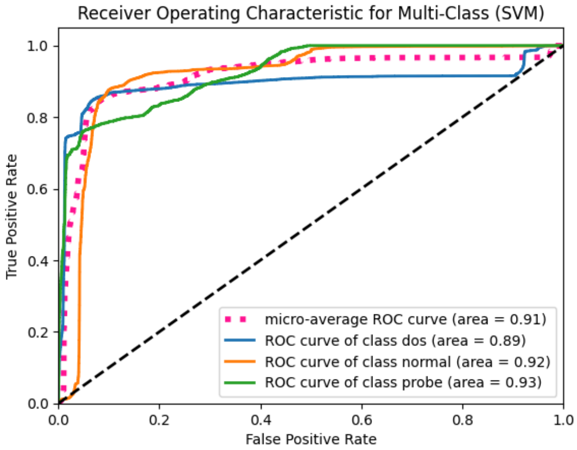


Fig. 12: SVM-ROC curves for multiclass classification

Statistical Test

Friedman’s non-parametric statistical test was utilized as the evaluation method to compare the classification algorithms robustly. This method was deliberately selected for its robustness in identifying statistically significant differences in performance among the evaluated models. In Tables 5 and 6, the mean rankings from Friedman’s test are showcased, offering a clear comparison of the competing methods across key performance metrics such as accuracy, precision, recall, and F-score in both cases (5 classes and 3 classes).

According to Table 5, DT has the lowest ranking values across various metrics (1.25 for Accuracy, Recall, and F-measure), indicating that it outperforms the other classifiers in this evaluation. Although the RF ranks second in various metrics (1.75 for Accuracy, Recall, and F-measure; 1.25 for Precision), it outperforms DT in Precision. SVM consistently ranks 3rd in all metrics (Accuracy, Precision, Recall, and F-measure = 3.0), making it the weakest classifier in the case of 5 classes.

Table 5: Mean ranking results of ML-based IDS classifications using Friedman’s statistical test (5 classes)

Classifier	*Acc*	*P*	*R*	*F−measure*
RF	1.75	1.25	1.75	1.75
DT	1.25	1.75	1.25	1.25
SVM	3.0	3.0	3.0	3.0
p-value	0.15612	0.15612	0.15612	0.15612

Table 6: Mean ranking results of ML-based IDS classifications using Friedman’s statistical test (3 classes)

Classifier	*Acc*	*P*	*R*	*F −measure*
RF	1.0	1.0	1.0	1.0
DT	2.0	2.0	2.0	2.0
SVM	3.0	3.0	3.0	3.0
p-value	0.13534	0.13534	0.13534	0.13534

In the case of the 3 classes (as listed in Table 6), RF ranks first (1.0) across all metrics when compared with other classifiers. These results show that RF outperforms all other classifiers in this 3-class IDS classification task. The DT consistently ranks second across all metrics (2.0), outperforming SVM (i.e., the weakest classifier) but trailing behind RF. The *p*-value (0.15612) in the case of five classes and *p*-value (0.13534) in the case of three classes for all metrics indicate that the observed ranking differences are not statistically significant at a conventional threshold (e.g., $\alpha = 0.05$). To further examine the distinctions between the control classifier and alternative models, the Holm method was employed as a post hoc statistical analysis. In line with Friedman’s test findings, the control classifier outperformed the others on all performance criteria.

In Tables 7 for five classes and 8 for three classes, the statistical results from Holm’s study are shown. Based on the classifiers’ Friedman ranks, $z - value$ is used to determine the statistical significance of differences between classifiers (i.e., $z_i = {}^{R}\underline{0}\sigma\underline{-}R^{Ri}$). The control classifier attained an average rank of R_0, the *i*th classifier’s rank is R^i, and σ_R is the Standard error of ranks.

Holm’s test was applied to evaluate the competing classifiers, leading to the rejection of hypotheses with *p*-values ≤ 0.025 across all evaluation metrics for both the five-class and three-class scenarios, as presented in Tables 7 and 8. Given that all *p*-values are above their adjusted significance levels, there are no statistically significant differences among the classifiers across any measures, according to Holm’s test findings in Table 7. For Accuracy, Recall, and F1-measure, DT was the control classifier, with SVM ($z = 1.75$, *p*-value= 0.080118) and RF ($z = 0.50$, *p*-value=0.617075) showing no significant deviation. Similarly, for Precision, RF was the control, and DT and SVM also resulted in non-rejection of the null hypothesis.

Table 7: Holm's test results among the classification models (5 classes)

i	Classifier	$z = \frac{(R_0 - R^i)}{\sigma_R}$	p-value	α÷i	Hypothesis
Accuracy (DT-control classifier)					
2	SVM	1.75	0.080118	0.025	Non Rejected
1	RF	0.50	0.617075	0.050	Non Rejected
Precision (RF-control classifier)					
2	SVM	1.75	0.080118	0.025	Non Rejected
1	RF	0.50	0.617075	0.050	Non Rejected
Recall (DT-control classifier)					
2	SVM	1.75	0.080118	0.025	Non Rejected
1	RF	0.50	0.617075	0.050	Non Rejected
F1-score (DT-control classifier)					
2	SVM	1.75	0.080118	0.025	Non Rejected
1	RF	0.50	0.617075	0.050	Non Rejected

Table 8: Holm's test results among the classification models (3 classes)

i	Classifier	$z = \frac{(R_0 - R^i)}{\sigma_R}$	p-value	α÷i	Hypothesis
Accuracy (DT-control classifier)					
2	SVM	2.0	0.045500	0.025	Non Rejected
1	DT	1.0	0.317311	0.050	Non Rejected
Precision (RF-control classifier)					
2	SVM	2.0	0.045500	0.025	Non Rejected
1	DT	1.0	0.317311	0.050	Non Rejected
Recall (RF-control classifier)					
2	SVM	2.0	0.045500	0.025	Non Rejected
1	DT	1.0	0.317311	0.050	Non Rejected
F1-score (RF-control classifier)					
2	SVM	2.0	0.045500	0.025	Non Rejected
1	DT	1.0	0.317311	0.050	Non Rejected

According to Table 8, the results of Holm's test confirm that RF is the best-performing classifier across all evaluation metrics (Accuracy, Precision, Recall, and F1-measure) in the 3-class classification task. RF consistently achieved the highest ranking as the control classifier, while DT and SVM ranked lower. While SVM shows a noticeable difference from RF (z = 2.0, p-value= 0.0455), the null hypothesis is not rejected due to the adjusted significance threshold ($\alpha \div\ i$ = 0.025). Similarly, DT (z = 1.0, p-value= 0.317311) does not show a significant difference from RF, reinforcing RF's superior standing. These findings suggest that RF provides the most robust classification performance and should be the preferred model for this task.

Conclusion and Future work

In this study, we compared three different intrusion detection systems that rely on machine learning: RF, DT, and SVM. We tested them on the NSL-KDD dataset to see how well they performed. The findings demonstrated that RF consistently outperformed DT and SVM across all evaluation metrics, achieving the highest accuracy (85.42%) in the three-class classification scenario. Additionally, the study highlighted the impact of reducing classification complexity, where simplifying the task from five classes to three classes improved detection accuracy and model stability, reducing the performance gap between training and testing. Friedman's test was applied to rank classifier performance to ensure statistical rigor, followed by Holm's post-hoc analysis to assess statistical significance. The results confirmed that RF consistently ranked highest in the reduced classes across all metrics. While the study provides strong empirical evidence supporting RF as the most robust classifier for intrusion detection, practical implementation in real-world environments introduces several challenges. Deploying machine learning-based Intrusion Detection Systems (IDS) requires careful consideration of computational efficiency, scalability, and adaptability to evolving cyber threats. Although the RF model has a strong performance, it might require further optimization to fulfill real-time detection needs, especially in high-traffic network environments. Additionally, periodic retraining is necessary to ensure IDS can adapt to emerging attack patterns. Future research should focus on ensemble learning approaches or hybrid IDS frameworks to enhance robustness and provide better threat detection. Furthermore, exploring new datasets and addressing class imbalance is crucial by utilizing resampling techniques. Addressing these implementation challenges will be vital for moving from experimental validation to practical deployment, strengthening protection against cyber threats.

Acknowledgment

The authors sincerely thank Zarqa University, Jordan, for fully funding this research.

Funding Information

This research is funded fully by Zarqa University, Jordan.

Author's Contributions

Walaa Hassan Elashmawi: Contributed to the methodology design and implementation of algorithms.

Alaa Sheta: Contributed to the conceptualization and data analysis.

Ahmad Al-Qerem: Contributed to the literature review, interpreting results, and manuscript drafting.

Ethics

The authors declare that no ethical issues are associated with the publication of this manuscript.

References

Agarwal, A., Sharma, P., Alshehri, M., Mohamed, A. A., & Alfarraj, O. (2021). Classification model for accuracy and intrusion detection using machine learning approach. *PeerJ Computer Science*, *7*, e437. https://doi.org/10.7717/peerj-cs.437

Ahmad, I., Basheri, M., Iqbal, M. J., & Rahim, A. (2018). Performance Comparison of Support Vector Machine, Random Forest, and Extreme Learning Machine for Intrusion Detection. *IEEE Access*, *6*, 33789-33795. https://doi.org/10.1109/access.2018.2841987

Alaketu, M. A., Oguntimilehin, A., Olatunji, K. A., Abiola, O. B., Badeji-Ajisafe, B., Akinduyite, C. O., Obamiyi, S. E., Babalola, G. O., & Okebule, T. (2024). Comparative Analysis of Intrusion Detection Models using Big Data Analytics and Machine Learning Techniques. *The International Arab Journal of Information Technology*, *21*(2), 326-337. https://doi.org/10.34028/iajit/21/2/14

Albahar, M. A., Binsawad, M., Almalki, J., El-etriby, S., & Karali, S. (2020). Improving Intrusion Detection System using Artificial Neural Network. *International Journal of Advanced Computer Science and Applications*, *11*(6). https://doi.org/10.14569/ijacsa.2020.0110670

Al-Daja, S., Ala'Alyabrodi, Al-Mousa, M. R., Al-Qammaz, A., Olimat, K. N., Olemat, H. M., & Daoud, M. S. (2023). Analyzing and contrasting machine learning algorithms for Intrusion Detection System. *2023 24th International Arab Conference on Information Technology (ACIT)*, 1-6. https://doi.org/10.1109/acit58888.2023.10453827

Aljawarneh, S., Aldwairi, M., & Yassein, M. B. (2018). Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model. *Journal of Computational Science*, *25*, 152-160. https://doi.org/10.1016/j.jocs.2017.03.006

Alkasassbeh, M., & Al-Haj Baddar, S. (2023). Intrusion Detection Systems: A State-of-the-Art Taxonomy and Survey. *Arabian Journal for Science and Engineering*, *48*(8), 10021-10064. https://doi.org/10.1007/s13369-022-07412-1

Almseidin, M., Alzubi, M., Kovacs, S., & Alkasassbeh, M. (2017). Evaluation of machine learning algorithms for intrusion detection system. *2017 IEEE 15th International Symposium on Intelligent Systems and Informatics (SISY)*, 000277-000282. https://doi.org/10.1109/sisy.2017.8080566

Axelsson, S. (2000). *Intrusion detection systems: a survey and taxonomy*.

Baleev, M., Shevchenko, A., Basan, E., Lapina, M., & Elashmawi, W. H. (2024). Detection of Anomalous Activity in Wireless Communication Channels. *Molecular System Biology*, *1207*, 16-23. https://doi.org/10.1007/978-3-031-77229-0_3

Bhattacharya, S., S, S. R. K., Maddikunta, P. K. R., Kaluri, R., Singh, S., Gadekallu, T. R., Alazab, M., & Tariq, U. (2020). A Novel PCA-Firefly Based XGBoost Classification Model for Intrusion Detection in Networks Using GPU. *Electronics*, *9*(2), 219. https://doi.org/10.3390/electronics9020219

Bitra, V. F., Kumar, A., Rao, S., Prakash, & Shakeel Ahmed, M. (2024). Comparative analysis on intrusion detection system using machine learning approach. *World Journal of Advanced Research and Reviews*, *21*(2), 2555-2562. https://doi.org/10.30574/wjarr.2024.21.3.0983

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5-32. https://doi.org/10.1023/A:1010933404324

Business Research Insights. (2024). *Intrusion Detection And Prevention Systems Market Report Overview*.

Dhanabal, L., & Shantharajah, S. P. (2015). A study on nsl-kdd dataset for intrusion detection system based on classification algorithms. *International Journal of Advanced Research in Computer and Communication Engineering*, *4*(6), 446-452. https://doi.org/10.17148/IJARCCE.2015.4696

Dini, P., Elhanashi, A., Begni, A., Saponara, S., Zheng, Q., & Gasmi, K. (2023). Overview on Intrusion Detection Systems Design Exploiting Machine Learning for Networking Cybersecurity. *Applied Sciences*, *13*(13), 7507. https://doi.org/10.3390/app13137507

Elashmawi, W. H., Djellal, A., Sheta, A., Surani, S., & Aljahdali, S. (2024a). Machine Learning for Enhanced COPD Diagnosis: A Comparative Analysis of Classification Algorithms. *Diagnostics*, *14*(24), 2822. https://doi.org/10.3390/diagnostics14242822

Elashmawi, W. H., Osman, H., Osama, M., & Nader, N. (2024b). Development Generative AI for Cybersecurity: Evaluating Script Generation and Attack Classification in Penetration Testing. *Springer*, *1207*, 48-62. https://doi.org/10.1007/978-3-031-77229-0_6

Eshak Magdy, M., M. Matter, A., Hussin, S., Hassan, D., & Elsaid, S. (2022). A Comparative study of intrusion detection systems applied to NSL-KDD Dataset. *The Egyptian International Journal of Engineering Sciences and Technology*, *43*(2), 88-98. https://doi.org/10.21608/eijest.2022.137441.1156

Gao, X., Shan, C., Hu, C., Niu, Z., & Liu, Z. (2019). An Adaptive Ensemble Machine Learning Model for Intrusion Detection. *IEEE Access*, *7*, 82512-82521. https://doi.org/10.1109/access.2019.2923640

Ghosh, S., Dasgupta, A., & Swetapadma, A. (2019). A Study on Support Vector Machine based Linear and Non-Linear Pattern Classification. *2019 International Conference on Intelligent Sustainable Systems (ICISS)*, 24-28. https://doi.org/10.1109/iss1.2019.8908018

Gong, Y., Mabu, S., Chen, C., Wang, Y., & Hirasawa, K. (2009). *Intrusion Detection System Combining Misuse Detection and Anomaly Detection Using Genetic Network Programming*. 2009 ICCAS-SICE, Fukuoka, Japan.

Halimaa A., A., & Sundarakantham, K. (2019). Machine Learning Based Intrusion Detection System. *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, 916-920. https://doi.org/10.1109/icoei.2019.8862784

Hossain, Md. A., & Islam, Md. S. (2023). Ensuring network security with a robust intrusion detection system using ensemble-based machine learning. *Array*, *19*, 100306. https://doi.org/10.1016/j.array.2023.100306

Ishtaiwi, A., Ali, A. M., Al-Qerem, A., Sabahean, M., Alzubi, B., Almomani, A., Alauthman, M., Aldweesh, A., & Al Khaldy, M. A. (2024). Next-Gen Phishing Defense Enhancing Detection With Machine Learning and Expert Whitelisting/Blacklisting. *International Journal of Cloud Applications and Computing*, *14*(1), 1-17. https://doi.org/10.4018/ijcac.353301

Japkowicz, N. (2006). Why question machine learning evaluation methods. *AAAI Workshop on Evaluation Methods for Machine Learning*, 6-11.

Jyothsna, V., & Prasad, K. M. (2019). Anomaly-Based Intrusion Detection System. *Computer and Network Security*, 35. https://doi.org/10.5772/intechopen.82287

Matos, L. M., Azevedo, J., Matta, A., Pilastri, A., Cortez, P., & Mendes, R. (2022). Categorical Attribute traNsformation Environment (CANE): A python module for categorical to numeric data preprocessing. *Software Impacts*, *13*, 100359. https://doi.org/10.1016/j.simpa.2022.100359

Mishra, P., Varadharajan, V., Tupakula, U., & Pilli, E. S. (2019). A Detailed Investigation and Analysis of Using Machine Learning Techniques for Intrusion Detection. *IEEE Communications Surveys & Tutorials*, *21*(1), 686-728. https://doi.org/10.1109/comst.2018.2847722

Quinlan, J. R. (1986). Induction of Decision Trees. *Machine Learning*, *1*(1), 81-106. https://doi.org/10.1007/BF00116251

Sheta, A., Elashmawi, W. H., Djellal, A., Braik, M., Surani, S., Aljahdali, S., Subramanian, S., & Patel, P. S. (2024). Comprehensive Evaluation of Machine Learning Techniques for Obstructive Sleep Apnea Detection. *International Journal of Advanced Computer Science and Applications*, *15*(12). https://doi.org/10.14569/ijacsa.2024.0151211

Sheta, A. F., & Alamleh, A. (2015). A professional comparison of c4.5, mlp, svm for network intrusion detection based feature analysis. *The International Congress for Global Science and Technology*, 15-66.

Tait, K.-A., Khan, J. S., Alqahtani, F., Shah, A. A., Ali Khan, F., Rehman, M. U., Boulila, W., & Ahmad, J. (2021). Intrusion Detection using Machine Learning Techniques: An Experimental Comparison. *2021 International Congress of Advanced Technology and Engineering (ICOTEN)*, 1-10. https://doi.org/10.1109/icoten52080.2021.9493543

Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009). A Detailed Analysis of the KDD CUP 99 Data set. *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, 1-6. https://doi.org/10.1109/cisda.2009.5356528

Xu, J., Liu, X., Huo, Z., Deng, C., Nie, F., & Huang, H. (2017). Multi-Class Support Vector Machine via Maximizing Multi-Class Margins. *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, 3154-3160. https://doi.org/10.24963/ijcai.2017/440