mathematics

MDPI

Article

A Machine Learning-Based Framework with Enhanced Feature Selection and Resampling for Improved Intrusion Detection

Fazila Malik [1,†], Qazi Waqas Khan [2,†], Atif Rizwan [2], Rana Alnashwan [3,*] and Ghada Atteia [3]

[1] Department of Computer Science, Iqra University Islamabad, Islamabad 44000, Pakistan ; fazila.malik@iqraisb.edu.pk
[2] Department of Computer Engineering, Jeju National University, Jejusi 63243, Republic of Korea; waqasqazi19@stu.jejunu.ac.kr (Q.W.K.); atifrizwan@jejunu.ac.kr (A.R.)
[3] Department of Information Technology, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia; geatteiaallah@pnu.edu.sa
* Correspondence: roalnashwan@pnu.edu.sa
† These authors contributed equally to this work.

Abstract: Intrusion Detection Systems (IDSs) play a crucial role in safeguarding network infrastructures from cyber threats and ensuring the integrity of highly sensitive data. Conventional IDS technologies, although successful in achieving high levels of accuracy, frequently encounter substantial model bias. This bias is primarily caused by imbalances in the data and the lack of relevance of certain features. This study aims to tackle these challenges by proposing an advanced machine learning (ML) based IDS that minimizes misclassification errors and corrects model bias. As a result, the predictive accuracy and generalizability of the IDS are significantly improved. The proposed system employs advanced feature selection techniques, such as Recursive Feature Elimination (RFE), sequential feature selection (SFS), and statistical feature selection, to refine the input feature set and minimize the impact of non-predictive attributes. In addition, this work incorporates data resampling methods such as Synthetic Minority Oversampling Technique and Edited Nearest Neighbor (SMOTE_ENN), Adaptive Synthetic Sampling (ADASYN), and Synthetic Minority Oversampling Technique–Tomek Links (SMOTE_Tomek) to address class imbalance and improve the accuracy of the model. The experimental results indicate that our proposed model, especially when utilizing the random forest (RF) algorithm, surpasses existing models regarding accuracy, precision, recall, and F Score across different data resampling methods. Using the ADASYN resampling method, the RF model achieves an accuracy of 99.9985% for botnet attacks and 99.9777% for Man-in-the-Middle (MITM) attacks, demonstrating the effectiveness of our approach in dealing with imbalanced data distributions. This research not only improves the abilities of IDS to identify botnet and MITM attacks but also provides a scalable and efficient solution that can be used in other areas where data imbalance is a recurring problem. This work has implications beyond IDS, offering valuable insights into using ML techniques in complex real-world scenarios.

Keywords: feature selection; data resampling; intrusion detection; applied machine learning; deep learning

MSC: 68T05; 68T07

check for updates

Citation: Malik, F.; Waqas Khan, Q.; Rizwan, A.; Alnashwan, R.; Atteia, G. A Machine Learning-Based Framework with Enhanced Feature Selection and Resampling for Improved Intrusion Detection. *Mathematics* **2024**, *12*, 1799. https://doi.org/10.3390/math12121799

Academic Editors: Shuo Yu and Feng Xia

Received: 29 April 2024
Revised: 3 June 2024
Accepted: 6 June 2024
Published: 9 June 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Due to the advancement of internet technology, internet-based services have become trendy in the last two decades, especially after the COVID-19 pandemic [1]. People typically use smartphones, laptops, tablets, and other electronic gadgets to access such services anytime and anywhere. As a result, the data start traveling through these networks between the machines and data storage centers, which could contain sensitive or private information [2]. Hence, it also creates a new opportunity for the attackers to break the

security walls and launch widespread attacks that the organization and the individuals may threaten [3]. Attackers use a variety of cutting-edge tactics to attack system security flaws. It might result in the misuse of private or sensitive information, unauthorized access to the system, or a breach of client accounts [4]. Defending against these assaults, safeguarding highly sensitive data, and protecting the networks from external threats have become the primary concerns for researchers and scientists [5]. Therefore, in current trends, one of the most prominent and popular mechanisms is IDS, which investigates incoming traffic and is classified as legitimate or malicious to detect potential threats in specific systems or networks [6].

An IDS is one of the critical security measures presently used as security techniques in the modern world to protect a network or system from potential attacks [7]. Even though many IDSs have been established over the past 20 years to identify and guard against prospective attacks, they still need more flexibility and scalability, which makes them continually exposed to hidden attacks [8]. Consequently, the domain of IDSs is a highly significant area of research due to the increasing number of attacks. An extensive IDS is necessary to analyze the vast amount of data, identify the crucial characteristics, and classify the traffic as either malicious or normal to prevent all potential attacks [9]. However, it might be challenging for an IDS to analyze incoming traffic to extract helpful or pertinent information from the massive amounts of data generated by evolving technologies and transmitted over networks [10]. To address these challenges, IDSs must employ a large dataset and feature selection methods capable of eliminating irrelevant data and identifying the features that impact attack detection [11]. Moreover, a large dataset sometimes includes noise and redundant or duplicate elements. In addition, a possible side effect of considering a massive dataset is that the feature count rises in direct correlation with the total number of observations [12]. It may lead to a significant number of false positive findings. Although numerous features in IDS datasets shed light on traffic flow anomalies, not all may be necessary for detection. Therefore, picking more useful features can boost IDS efficiency and effectiveness. The real-world data of internet traffic have very few samples of an attack and many samples of a normal class, which leads to a class imbalance problem. Many studies have used feature selection methods in this context to address data dimension issues and data resampling methods to solve the class imbalance problem. However, they have higher false positive and lower true positive rates. This study proposed a method to enhance the prediction performance of an ML model for detecting attack classes with lower false positive rates.

This study utilizes feature selection and data resampling methods with ML models to handle the abovementioned problems. SFS, RFE, and statistical feature selection are utilized to select the relevant features. The Principal Component Analysis (PCA) and Deep Autoencoder (DAE) extract the features from the MITM and botnet attack datasets. The fusion of PCA and DAE is performed using the early fusion strategy. Further, SMOTE_ENN, ADASYN, and SMOTE_Tomek methods are used to perform data resampling of minority and majority classes of datasets. The performance results of the RF, Gradient Boosting (GB), TabNet, and neural oblivious decision ensembles (NODE) models are evaluated using the abovementioned feature selection and data resampling methods. The experiments of ML models are performed on WUSTL and UNSW 2018 datasets, respectively, for MITM and botnet attack classification. The key contribution of this study is listed below:

- Utilized several feature selection methods to select the relevant features;
- Employed several data resampling methods to solve the class-imbalanced problem;
- Conducted the experiments using the deep tabular and tree-based methods;
- Utilized the Bayesian optimization method to search the optimal parameters of learning models.

The rest of the paper has been organized as follows: Section 2 presents the existing literature on intrusion detection systems. Section 3 presents the proposed methodology. Sections 4 and 5 present the results and conclusions of the proposed intrusion detection system.

2. Related Work

In the paper [13], the authors analyzed the possibility of implementing ML-based IDS for resource-constrained Internet of Things (IoT) systems. The proposed ML system is used to identify abnormal activity on vulnerable IoT networks. The ideal solution for a Deep Learning (DL) based IDS is assessing the technique's performance against five diverse attack situations, including opportunistic service attacks, black holes, Distributed Denial of Service (DDoS), sinkholes, and wormhole attacks. Through the review of precision–recall curves, an average rate of precision of 95% and a rate of recall of 97% for various attack conditions were achieved. In another significant contribution to the field, [14] presents a new model for IDS that relies on a unique two-tier classification model and a two-layer dimension reduction approach. This model identifies malignant actions such as remote to local and root attacks. Linear Discriminant Analysis (LDA) achieves the model's dimensionality reduction. In contrast, a two-tier classification model is employed to detect malicious activity based on k-nearest neighbor (KNN) factor rendition and Naive Bayes (NB). The model's effectiveness is demonstrated through the NSL-KDD dataset, where it outperforms previous models in identifying U2R and R2L attacks with detection accuracies of 70.15% and 42%.

In the paper [15], the authors proposed an anomaly detection technique utilizing DL methods for gateway intrusion detection and Support Vector Machine (SVM) for Wireless Sensor Network (WSN) intrusion identification. The proposed detection protocol actively hierarchically performs an on-demand SVM classifier, when an intrusion is supposed to occur. The ML classification with a statistical methodology for malignant node localization was combined. The methodology combined statistical-based and two ML methods and identified this attack with accuracy (over 95%) when malignant node packet dropping rates were high. In the study [16], the authors presented the enhanced Genetic Algorithm (GA) integrated with a Deep Belief Network (DBN) in a paper. DBN improved the classification results and was capable of processing high-dimensional data effectively. After several rounds, GA generated the optimal network structure. DBN then categorized the attacks using the network structure gained as an intrusion detection system. Finally, the algorithm model was simulated and assessed using the NSL-KDD dataset. The challenge of selecting an appropriate network architecture while employing deep learning techniques for intrusion detection was addressed while protecting against diverse threats. It improved the model's classification accuracy and prediction while reducing network complexity.

The paper [17] suggests an effective ensemble feature selection technique for IDS to identify the best-performing subset for attack detection. The KDDCup-99 network dataset compares the performance results with the standard feature selection techniques. The reported outcomes confirm our system's effectiveness in F Score, AUC score, accuracy, recall, precision, and execution time. The paper [18] presented a novel Golden Jackal Optimization Algorithm-based method for network security that combines a DL-assisted intrusion detection system with the Golden Jackal Optimization Algorithm. The primary goal of this system is to identify and classify intrusions to ensure network security accurately. This method uses the Attention-based Bidirectional Long Short-term Memory (A-BiLSTM) model. Based on the comparative results, this method beats the other models. The paper [19] developed a novel, effective defensive solution against adversarial ML attacks for IDS. Using Thompson sampling, Apollon's multi-armed bandits model selects the optimal classifier for every real-time input. It adds uncertainty to the IDS behavior, making it harder for attackers to duplicate and generate unfriendly traffic.

The paper [20] presented a model combining different algorithms that significantly improve detection features. Their method obtains time-based relevant features using a Bidirectional Long Short-term Memory Network (BI-LSTM) and a Temporal Convolutional Network (TCN), then reduces the dimensional of the features using a Stacked Sparse Autoencoder (SSAE). By fine-tuning the time steps, they highlight the importance of temporal data in promoting detection accuracy. This study [21] aims to create and utilize a Deep Neural Network (DNN) model to determine computer network intrusions. The CICIDS

2017 dataset's data imbalance problem is addressed using SMOTE and random sampling methods. With an accuracy score of 99.68% and a loss of 0.0102, the results show that the DL model performed well at predicting attacks using the CICIDS 2017 dataset. The paper [22] proposed a method to combat security challenges and ensure the safety of IoT networks. They combine different DL and optimization techniques to support IoT devices against possible threats and unauthorized access. Various samples were collected from the NSL-KDD and BoT-IoT datasets to authenticate the efficiency of the proposed method.

The paper [23] used a Convolutional Neural Network (CNN) for botnet attack detection. A DL-based categorization model is applied to detect botnet activity in network traffic. The CTU-13 dataset trains and evaluates a real-time model to identify zero-day botnet attacks. Using the neural networks, the proposed model demonstrates good results in correctly detecting botnet attacks. The results show that the Artificial Neural Network (ANN) model can correctly and effectively detect botnets. The study [24] utilized the B-CAT model, which uses deep attack behavior analysis on network traffic flows for botnet detection. A DL architecture's automatic feature extraction capabilities are highlighted as a significant advantage in botnet detection. The proposed approach consists of two primary parts. The first phase is to train and build a knowledge base, then proceed to test for botnet activity and attack characteristics. It employs dynamic thresholds to improve the model's sensitivity in identifying attack elements via similarity analysis. Experiments have been carried out for the evaluation using three unique datasets, with the results revealing that some performed better than others.

The study [25] presented a novel approach for enhancing botnet attack detection in IoT devices. The study utilized the UNSW-NB15 dataset and evaluated the proposed system using various classification models, including decision trees, random forests, k-nearest neighbors, adaptive boosting, and bagging. This study utilized three feature selection methods: generalized normal distribution optimization, correlation analysis, and the lasso method. The experimental results show that the Adaboost model has 99.38% accuracy. Another study [26] utilized the PCA to extract the features of DDoS attack classification. RF, KNN, and Naïve Bayes (NB) were used to evaluate the effectiveness of the feature extraction. The results of the experiments show that combining PCA and Robust Scaler preprocessing approaches significantly improves the accuracy of DDoS attack detection in connected devices.

The study [27] proposed an intelligent detection system for identifying cyber-attacks in Industrial IoT networks. The proposed model uses the Singular Value Decomposition (SVD) technique to reduce data features and improve detection results. They used the SMOTE technique to avoid over-fitting and under-fitting issues that result in biased classification. Several ML and DL algorithms have been implemented to classify data for binary and multi-class classification. They evaluated the efficacy of the proposed intelligent model on the ToN dataset. The proposed approach achieved an accuracy rate of 99.99%, a reduced error rate of 0.001% for binary classification, an accuracy rate of 99.98%, and a reduced error rate of 0.016% for multi-class classification. The study [28] presented a BEFNNet (BERT-based Feed-Forward Neural Network) framework suitable for malware detection. This study used an innovative architecture with several modules to analyze eight datasets, each representing a different kind of malware. The Spotted Hyena Optimizer (SHO) was employed to optimize BEFNNet and demonstrate its flexibility in handling various types of malware data. BEFSONet has been shown to have outstanding performance metrics in numerous exploratory research and comparative evaluations. The paper [29] thoroughly investigated an IDS in an IoT system. This work will assist researchers by offering advice on dataset selection and proving the utility of the Fisher score technique. Careful comparative research uses selection approaches, such as Mutual Information (MI), Chi-Square (CHI), PCA, and RFE. This study utilized the logistic regression model. The findings highlight the Fisher Score algorithm's significance and accuracy in selecting essential criteria for intrusion detection in IoT systems.

3. Proposed Methodology for Intrusion Detection

Figure 1 presents the architecture of the proposed work for intrusion detection. This study performed the experiments using the WUSTL and UNSW datasets. In the pre-processing stage, this study selected the optimal features using sequential feature selection, Recursive Feature Elimination, and statistical feature importance methods. This study also used data resampling methods such as SMOTE_ENN, ADASYN, and SMOTE_Tomek to resample the data. After this, RF, GB NODE, and TabNet models for intrusion detection were used. The performance of these methods was evaluated using the accuracy, precision, recall, and f-score metrics. Algorithm 1 shows the steps of the proposed methodology.

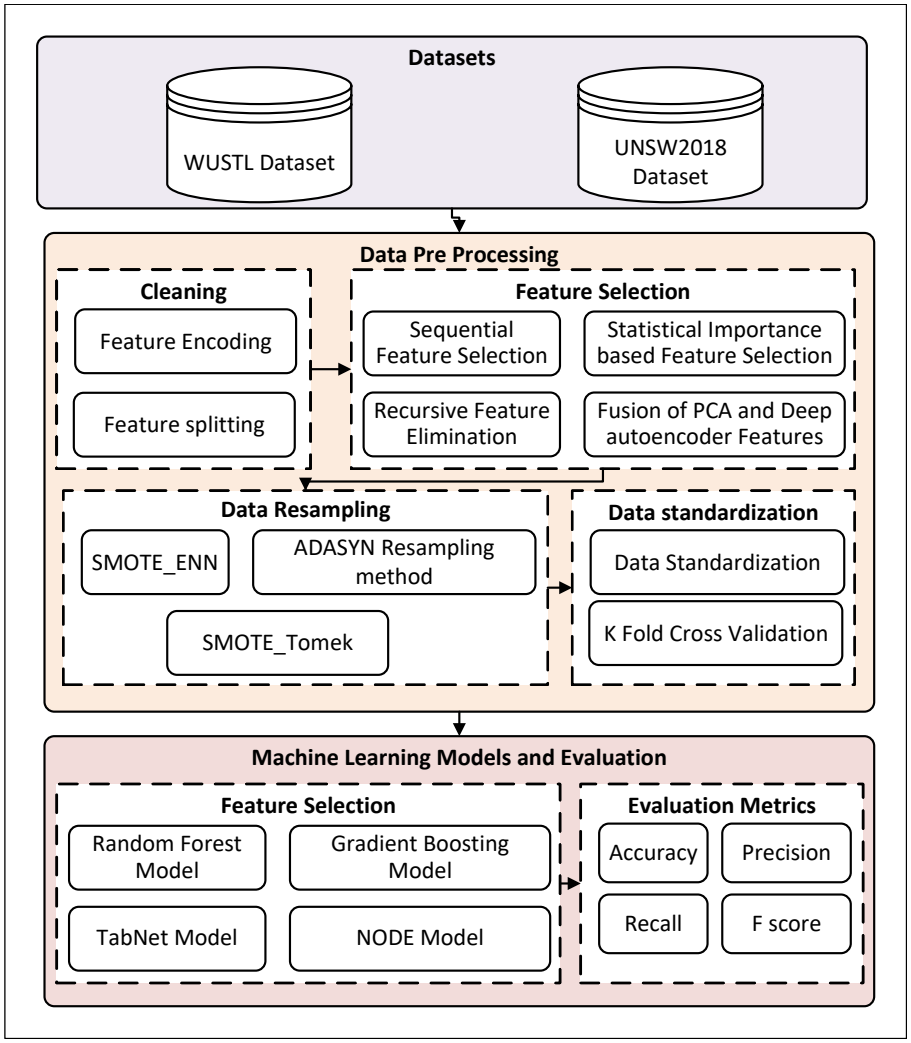


Figure 1. Architecture diagram of proposed ML-based intrusion detection system.

Algorithm 1: ML methods for intrusion detection.

```
Data: Two datasets Dataset1 = Wustl dataset and Dataset2 = Unsw2018 dataset
Result: Attack or Normal Class
initialization;
Dataset ← READCSV();
if AlphaNumericFeatures then
    Dataset ← FeatureEncoding(Dataset);
Dataset ←FeatureSplitting(Dataset);
Dataset ← SFS_FeatureSelection(Dataset);
Dataset ← RFE_FeatureSelection(Dataset);
Dataset ← Statistical_FeatureSelection(Dataset);
for each sample in Dataset do
    Dataset ← SMOTE_ENN(Dataset);
    Dataset ← ADASYN(Dataset);
    Dataset ← SMOTE_Tomek(Dataset);
Dataset ←StandaradScalar(Dataset);
Models ← Model_Parameter_Initialization()
for each Models do
    Trainx,Trainy,Testx,Testy ← DataSplit(Dataset);
    Model ← TrainModel(Trainx, Trainy);
    Predictedy ← Model.Predict(Testx);
    Accuracy ← AccuracyMetrics(Predictedy, Testy);
    Precision ←PrecisionMetrics(Predictedy, Testy);
    Recall ← RecallMetrics(Predictedy, Testy);
    Fscore ← FscoreMetrics(Predictedy, Testy);
```

3.1. MITM and Botnet Attack Dataset Detail

This study utilized the two benchmark datasets to validate the model performance. The details of these datasets are given below.

3.1.1. WUSTL Man-in-the-Middle Attack Dataset

The WUSTL dataset belongs to the Internet of Medical Things (IoMT) environment, and it contains information on the biometric flow of the patient and the network flow metric. The Enhanced Healthcare Monitoring System (EHMS) testbed created this real-time WUSTL dataset. This dataset has 43 independent features and one target label. Thirty-five features in this dataset are related to the network flow metrics, and eight features are associated with the biometric flow of the patients. It has 16,318 instances, 14,272 related to the normal class, and 2046 to the attack class. The class distribution of the WUSTL dataset is presented in Figure 2.

Figure 2 shows the attack and normal class distributions. In this figure, the x-axis represents the classes, and the y-axis represents the number of samples or data points. It depicts that the attack class has fewer samples than the normal class and is highly imbalanced.

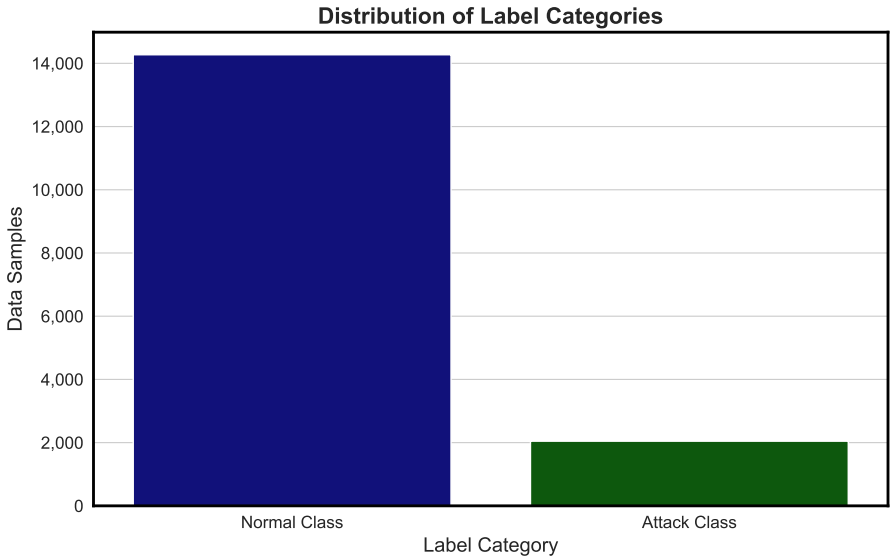


Figure 2. Distribution of a WUSTL dataset's class label.

3.1.2. UNSW 2018 Bot-Net IoT

In their study, Koroniotis presented the bot IoT dataset [30]. At UNSW Canberra's research cyber range lab, they set up a testbed setting. This dataset has 29 features that are related to network traffic. This complete dataset consists of 16.7 GB; however, we used a subset of this dataset for the experiment, and the name of this dataset is UNSW2018IoTBotNetDataset. This CSV file has 972,839 instances; 971,149 belong to a normal class, and 1690 belong to an attack class. The class distribution of a UNSW 2018 dataset is presented in Figure 3.

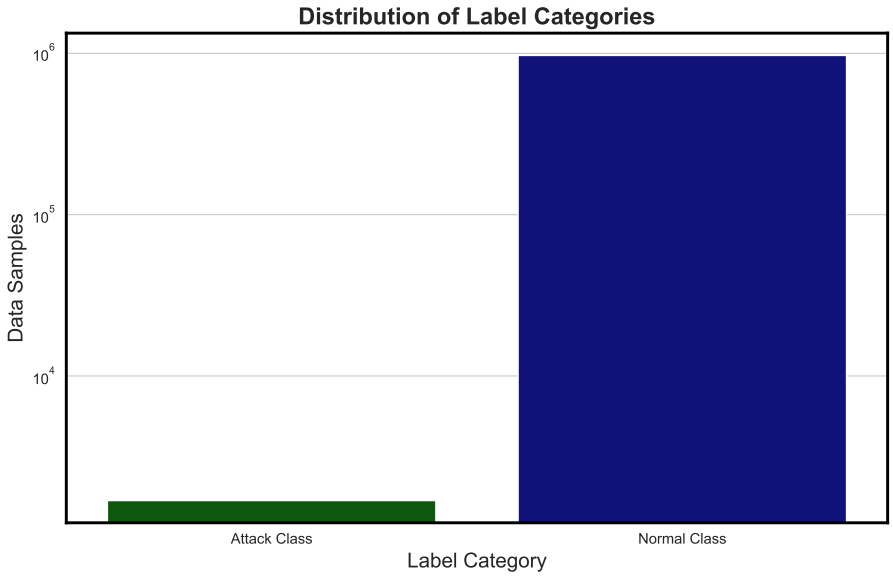


Figure 3. Distribution of a UNSW 2018 dataset's class label.

Figure 3 shows the class distribution of normal and attack classes. In this figure, the x-axis values show the class labels and the y-axis represents the data samples on the log scale. This study used the log scale on a y-axis to show the distribution of an attack class. In the attack class, we have 1690 data samples; in the normal class, we have 971,149 data samples.

3.2. Data Pre-Processing, Feature Selection, Feature Extraction and Data Resampling Methods for Intrusion Detection

Data preprocessing is a process of preparing raw data for ML model training. This study utilizes label encoding and the z-score method to transform text categories into numerics. The z-score standard scaling method converts the data to the same scale. Label encoding is a technique for transforming categorical data into numerical form. The WUSTL and UNSW 2018 datasets have some text categorical features. This study converts those text-categorical features into numeric categories using the label encoding. Data standardization is a method for bringing features to the same scale. Both the WUSTL and UNSW datasets have some features with different scales of values. Standardizing these data reduces the impact of features with larger values and promotes fair learning across all features. It converts the data between 0 and 1. It is computed using Equation (1):

$$Z = \frac{X_i - \mu}{\sigma} \tag{1}$$

where X_i is an ith input feature, and μ and σ are the mean and standard deviation of the data.

3.2.1. Feature Selection Methods for Intrusion Detection

Feature selection is a method in which a subset of relevant features is selected from the original dataset to increase the model's accuracy. It also helps to reduce the possibility of a model over-fitting [31]. This study utilized the RFE, SFS, and statistical feature selection methods to select the relevant feature from the original dataset.

3.2.2. Recursive Feature Elimination (RFE) Method

Recursive Feature Elimination is a selective feature elimination technique for identifying essential features in a dataset. The process involves the model with residual features, removing the less important parts until the desired number of features is identified and eliminating the weakest features [32]. This study utilized this feature selection method to select the most relevant and informative features from the WUSTL and UNSW datasets. This feature selection method selects the top 17 best features from the UNSW and 24 features from the WUSTL dataset.

3.2.3. Sequential Feature Selection (SFS) Method

The sequential feature selection method selects the best K from the original feature sets using the forward and backward feature strategies. The SFS is a greedy feature selection approach that iteratively evaluates the multiple feature sets and selects the best feature set based on cross-validation accuracy. This study applies this method on a WUSTL and UNSW dataset to select the top 24 and 17 best features based on a cross-validation accuracy score, respectively [33]. This study utilized the forward strategy, which adds the features to the selected features until a desired number of features is selected.

3.2.4. Statistical Feature Selection Method

The feature selection technique identifies relevant elements that support the data's learning patterns and are more closely connected with the attack or normal class output labels. This study presents a combined feature rank that analyzes the importance of each feature and quantifies its contribution to the attack classification process [34]. The combined feature rank is calculated using the standard deviation and the mean and median differences of both the WUSTL and UNSW datasets. The highest-valued characteristics have robust matching and minimal redundancy. The steps of this feature selection method are presented below.

- Calculate the dataset's standard deviation σ.

- Sort the characteristics by standard deviation, from highest to lowest. Assign the rank determined from standard deviation σ as $Rank1$.
- Measure the absolute difference D between the dataset's mean and median features.
- Rank characteristics depending on their differential value, from high to low. Assign the rating generated from difference D as $Rank2$.
- Find the combined feature rank: Combined Feature Rank = $Rank1 + Rank2$.
- Recursively add features to the feature subset based on their combined rank until the accuracy is equal to the previously calculated feature subset.

This feature selection method selects the 24 features from the WUSTL dataset and the 17 best features from the UNSW dataset.

3.2.5. Feature Extraction Method for Intrusion Detection

This study used the DAE and PCA to extract features from both datasets. The details of both methods are presented in the subsequent section.

3.2.6. Deep Autoencoder for Feature Extraction

The DAE is a feed-forward neural network that can be used as an unsupervised selection tool. It can identify the most critical storage location identifiers for reconstructing the input data. Autoencoders are specialized neurons that learn how to reconstruct feedback. The WUSTL and UNSW datasets are passed as input to an encoder of this model's module. The encoder learns the initial representation of standard data and its return with minimal errors. The encoder of this model compresses the input data of MITM and botnet attacks. The decoder attempts to rebuild the information from these compressed data of MITM and botnet attack [35].

3.2.7. Principal Component Analysis for Feature Extraction

The PCA is a dimensionality reduction technique used to reduce the dimensionality of large datasets. The main goal of PCA is to reduce the dimensionality of a dataset to preserve important patterns or relationships between variables without prior knowledge of the target variable. This study utilized the PCA method to reduce the dimensionality of the WUSTL and UNSW datasets. Using the PCA method, higher-dimensional space is mapped to data in a lower-dimensional space [36].

3.2.8. Fusion of Principal Component Analysis and Deep Autoencoder Features

This study fused the extracted features after extracting them using the PCA and DAE. It used the early fusion method [37], which fuses information before starting the primary learning process. It helps to preserve the information from both methods to enhance the model performance. Figure 4 shows the architecture of the early fusion method.

3.2.9. Data Resampling to Balance Class Distribution in WUSTL and UNSW Data

Data resampling is the selection of random cases for replacement from the original data sample so that any number of samples drawn contains cases similar to the original data sample. Using data resampling methods can reduce the possibility of over-fitting, which occurs when the model is too complex and fits the training data too well [38].

3.2.10. Synthetic Minority Oversampling Technique and Edited Nearest Neighbor

The Synthetic Minority Oversampling Technique (SMOTE) is a statistical technique that allows us to increase the amount of information in our dataset to make it balanced. This study passes the WUSTL and UNSW data as input to a SMOTE_ENN method. SMOTE clusters the observations in the minority (attack) class by linear interpolation to increase the number of samples in the minority (attack) class. At the same time, the Edited Nearest Neighbors (ENNs) reduces the number of samples in the majority (normal) class by removing noisy samples from the majority (normal) class [39]. The main goal of this method is to

enhance the data points of an attack class and reduce the data points of a normal class for both the WUSTL and UNSW datasets.

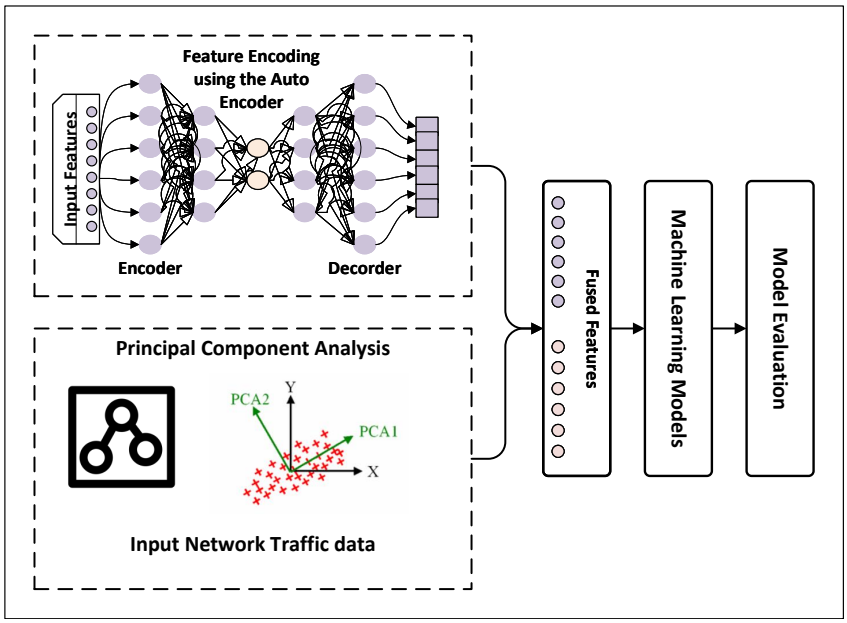


Figure 4. Architecture diagram of DAE and PCA feature's fusion.

3.2.11. Adaptive Synthetic Sampling (ADASYN) for Data Re-Sampling

The ADASYN resampling method is employed to solve the class imbalanced problem of WUSTL and UNSW datasets. This method focuses on the feature space of the original dataset and considers the learning difficulty for each minority sample (attack class). This method generates synthetic samples tailored to that particular class by calculating the density distribution of each minority attack (class). Based on their unique local distributions, it creates new data points, particularly for the minority (attack class) [40].

3.2.12. Synthetic Minority Oversampling Technique–Tomek Links (SMOTE_Tomek)

SMOTE_Tomek is an extended version of SMOTE that handles categorical and numerical data. This method is effective because we can use the SMOTE for numerical features before utilizing the nearest neighbors of the newly created synthetic samples to determine the value of categorical features. This method consists of three steps: select similar samples in a feature space, create a line between the features, and generate the synthetic sample at a point along that line. It utilizes the SMOTE and Tomek under-sampling techniques while removing the overlapping majority (attack) classes. This study used this method to balance the class distribution of an attack and normal class for both WUSTL and UNSW datasets [41].

3.3. ML Models for Intrusion Detection

TabNet, RF, GB, and NODE models are applied to process data to classify attacks. The Bayesian optimization method tunes a hyperparameter, and the K-fold cross-validation method validates the model. Bayesian optimization utilizes probabilistic mechanisms such as the Gaussian process. It uses the acquisition function that controls the balance between exploitation and exploration. Bayesian optimization best suits situations where evaluations are expensive for multiple trials. It utilizes the surrogate model in internal processes that reduce the computation time. The objective is to use the RF and GB models that perform well in the case of tabular data, provide robustness against over-fitting, and handle the large feature set effectively. On the other hand, the TabNet and NODE models offer the benefit of handling high-dimensional data and model flexibility to learning patterns compared to traditional decision trees, respectively.

3.3.1. Random Forest Model for Intrusion Detection

The random forest model is an ensemble learning method commonly used in ML for prediction tasks. It is more suitable for structural tabular data because of its ability to handle high-dimensional data, resistance to over-fitting, and robustness. This study used this model for MITM and botnet attack classification. The features of the WUSTL and UNSW datasets are passed to an RF model, and the input data are split using the bootstrapping method. The multiple decision trees are grown and combined with their output to reach the final result. Each decision tree model gives the output based on the input data. The RF model merges the final output using the voting mechanism [42]. The hyperparameter of the RF model is presented in Table 1.

Table 1. Hyperparameter of random forest.

S.No.	Hyperparameter	Values
1	N Estimators	48
2	Criterion	Gini
3	Bootstrap	True
4	Max Features	15
5	Max Depth	9
6	Min Samples Split	2
7	Random State	None
8	Min Samples Leaf	1

3.3.2. Gradient Boosting Model for Intrusion Detection

Gradient Boosting is another ensemble learning method in ML. This study used the GB model for MITM and botnet attack classification because it works better for tabular structural data such as hand datasets. The GB model consists of combining multiple weak learners into strong learners. This study trained this model sequentially on WUSTL and UNSW datasets, and each new model corrected the errors of the previous one. The idea is gradually approaching the optimal prediction by learning from previous misclassifications of an attack or normal class. In this model, the gradient of the loss function is computed in each iteration, and then a new weak learner is trained to minimize the error. The final model is an ensemble of weak learners, refined and combined to create a better and more accurate model with minimized error on attack and normal classes [43]. The hyperparameter of a GB model is presented in Table 2.

Table 2. Hyperparameter of gradient boosting.

S.No.	Hyperparameter	Values
1	Learning Rate	0.1
2	Max Features	12
3	N Estimators	40
4	Max Depth	7
5	Min Samples Split	2
6	Random State	None
7	Min Samples Leaf	1

3.3.3. TabNet Model for Intrusion Detection

TabNet is a DL model specially designed for tabular data. This study used the TabNet model to classify a botnet and MITM attack. The features of WUSTL and UNSW are passed to a feature transformer of a TabNet model, which transforms them into a more valuable data representation. After the attentive transformer block, the attention transformer block helps the model focus on essential network traffic features and ignore irrelevant ones. At each decision step, the attentive transformer selects which feature to pay attention to

based on its importance for the current input. This step is followed by feature masking that makes the model learn from the informative features and improves its generalization ability, preventing over-fitting. The split block splits the processed network traffic data into multiple parts, facilitating efficient processing and decision-making. In the decoder phase, the TabNet model takes the organized network traffic data from the encoder to make predictions of attack or normal classes [44]. The hyperparameter of a TabNet model is presented in Table 3.

Table 3. Hyperparameter of TabNet model.

S.No.	Name of Parameter	Values
1	Optimizer	Adam
2	Learning Rate	0.02
3	Output Layer Activation Function	Sigmoid
4	Batch Size	512
5	Epochs	35

Neural Oblivious Decision Ensembles (NODE) Model for Intrusion Detection

Neural oblivious decision ensembles are a DL model that combines the strength of neural networks and decision trees. This study employed this model to classify MITM and botnet attacks. The network traffic data are passed to a NODE model, which uses decision trees to consider multiple network traffic features at each decision node. It enables the model to capture complex interactions, leading to a better understanding of relationships between the target class label of normal and attack classes and network traffic input features. This model also utilizes a neural attention mechanism to assign weights to features and decision trees to help identify their relevance and importance. This process allows the models to focus more on the data's essential aspects, which helps improve accuracy. In the prediction phase, it combines the predictions of several pairs of decision trees and neural networks to make the final prediction. By combining the strengths of both models, NODE improves interpretability and accuracy [45]. The NODE model is effective for intrusion detection due to its combination of the interpretability of decision trees and the flexibility and power of neural networks. The hyperparameter of a NODE model is presented in Table 4.

Table 4. Hyperparameter of NODE model.

S.No.	Name of Parameter	Values
1	Optimizer	Adam
2	Learning Rate	0.001
3	Batch Size	32
4	Epochs	30
5	Activation Function	Sigmoid
6	Number of Trees	5

3.4. Evaluation Metrics for Validating ML Model Performance

In an ML paradigm, evaluation metrics are used to evaluate models' performance. These metrics help stakeholders see the effectiveness of the ML model on unseen data. This study used the following evaluation metrics.

3.4.1. Accuracy

Accuracy is a metric by which we can determine the model's true prediction. The formula to determine accuracy is correct prediction divided by total prediction (true and false):

$$Accuracy = CorrectPrediction / TotalPrediction \quad (2)$$

3.4.2. Precision

Precision measures how often a model predicts the correct answer from a training dataset. The formula for precision divides the actual true prediction by the total number of true predictions (true and false):

$$Precision = TP/(TP + FP) \tag{3}$$

3.4.3. Recall

Recall determines how many times the model recognizes the true value. It is computed by dividing the true prediction by the total of true positives and false negative predictions:

$$Recall = TP/(TP + FN) \tag{4}$$

3.4.4. F Score

F score is a harmonic mean of precision and recall. It is computed using the equation

$$F\ score = 2 \times \frac{(Precision \times Recall)}{(Precision + Recall)} \tag{5}$$

4. Results and Discussion

This section discusses the results of our proposed method for classifying attacks. The botnet and MITM attack dataset has many features and an imbalanced class distribution. This imbalanced class distribution affects a model's performance, and redundant features cause over-fitting. This work utilized the feature selection method to select the optimal feature from the dataset, and different variants of the SMOTE data resampling methods were used in this study to balance the class distribution. This work performed the experiments using the two datasets and utilized several feature selection and data resampling methods to accurately predict attack and normal classes. This study monitored the impact of feature selection and data resampling methods on the experimental results. Table 5 shows the details of the experimental setup. Table 6 presents the experimental results of fused features using the three data resampling methods for botnet attack classification. Table 7 presents the results of the RFE feature selection method for botnet attack classification. Similarly, Tables 8 and 9 present the experimental results of ML models using the SFS and statistical feature selection methods for botnet attack classification. The experimental results for MITM attack classification are presented in Tables 10–13, respectively, using the fused features, RFE, SFS, and statistical feature selection.

Table 5. Detail of experimental setup.

S.No.	Components	Detail
1	Hardware System	Intel Core i9 10th Gen
2	Operating System	Window 11
3	RAM	64 GB
4	Data File Format	CSV
5	Programming Language	Python
6	Required Packages	Numpy, Pandas, Hyperopt,imblearn, Scikit, Pytorch, Matplotlib, Seaborn
7	IDE	Vs Code

4.1. Experimental Results of Botnet Attack Classification

This section presents the results of an ML model for botnet attack classification. Table 6 shows the results of ML models using the fused feature with three different resampling techniques: SMOTE_ENN, ADASYN, and SMOTE_Tomek. The random forest model performs better than other ML models with all resampling methods. The NODE model has lower prediction results than other learning models on the three data resampling methods.

The highest prediction accuracy of the RF model is 99.9985% with the ADASYN resampling and statistical feature selection method.

Table 6. Experimental results of botnet classification using fused features.

Method	Accuracy (%)	Precision (%)	Recall (%)	F Score (%)
SMOTE_ENN				
RF	99.9176	99.9175	99.9176	99.9175
NODE	97.3102	97.2849	97.3102	96.8299
TabNet	99.7899	99.7889	99.7899	99.7885
GB	98.7890	98.7869	98.7890	98.7111
Experimental Results of ADASYN				
RF	99.9467	99.9466	99.9467	99.9466
NODE	97.4742	97.9172	97.4742	97.6177
TabNet	99.9501	99.9501	99.9501	99.9500
GB	98.4814	98.4661	98.4814	98.3924
Experimental Results of SMOTE_Tomek				
RF	99.9379	99.9378	99.9379	99.9379
NODE	96.6894	96.4085	96.6894	96.3190
TabNet	99.9365	99.9365	99.9365	99.9363
GB	98.4473	98.4377	98.4473	98.3492

Table 7 presents the results of ML models using the RFE feature selection method. The RF model has better prediction performance than the other models. With this feature selection method, the RF model has the highest prediction performance of 99.99% with a SMOTE_ENN resampling method. On the other hand, the GB and NODE models have lower prediction performance.

Table 7. Experimental results of botnet classification using RFE features.

Method	Accuracy (%)	Precision (%)	Recall (%)	F Score (%)
SMOTE_ENN				
RF	99.9961	99.9961	99.9961	99.9961
NODE	95.4966	95.7001	95.4966	93.6086
TabNet	99.9353	99.9353	99.9353	99.9351
GB	99.2313	99.2195	99.2313	99.2111
Experimental Results of ADASYN				
RF	99.9796	99.9796	99.9796	99.9796
NODE	94.1864	94.5244	94.1864	91.3786
TabNet	99.8958	99.8959	99.8958	99.8953
GB	99.2785	99.2693	99.2785	99.2658
Experimental Results of SMOTE_Tomek				
RF	99.9796	99.9796	99.9796	99.9796
NODE	94.1864	94.5244	94.1864	91.3786
TabNet	99.8958	99.8959	99.8958	99.8953
GB	99.2785	99.2693	99.2785	99.2658

In Table 8, the results of SFS feature selection are presented. The results demonstrate that the RF model has higher prediction results with SMOTE_ENN, ADASYN,

and SMOTE_Tomek data resampling methods. At the same time, the GB model shows variability in performance, especially under the ADASYN technique, where it drops to around 97.13% in precision. The TabNet and NODE models perform better under the ADASYN data resampling method.

Table 8. Experimental results of botnet classification using SFS features.

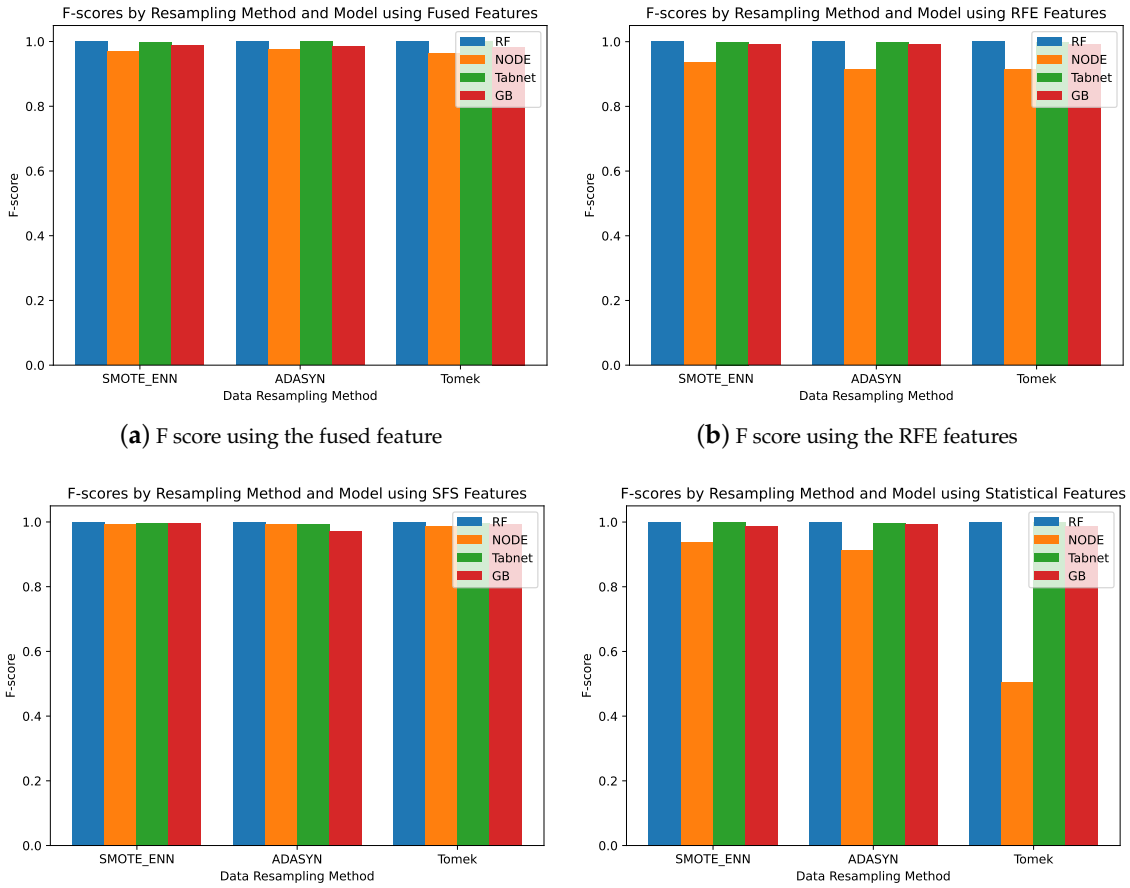
Method	Accuracy (%)	Precision (%)	Recall (%)	F Score (%)
SMOTE_ENN				
RF	99.9234	99.9232	99.9234	99.9232
NODE	99.3578	99.3551	99.3578	99.3394
TabNet	99.7131	99.7112	99.7131	99.7108
GB	99.7651	99.7695	99.7651	99.7665
Experimental Results of ADASYN				
RF	99.9690	99.9690	99.9690	99.9689
NODE	99.5131	99.5108	99.5131	99.5118
TabNet	99.5063	99.5337	99.5063	99.5138
GB	97.1666	97.3110	97.1666	97.2281
Experimental Results of SMOTE_Tomek				
RF	99.9350	99.9349	99.9350	99.9349
NODE	98.8747	98.8530	98.8747	98.8587
TabNet	99.5455	99.5424	99.5455	99.5433
GB	99.3316	99.3247	99.3316	99.3196

Table 9 demonstrates the results of ML models using the statistical feature selection method. The presented results show that under the ADASYN resampling method, the RF model has better results than the other models. On the other hand, the TabNet model maintains the performance and shows robustness across different resampling methods. The presented results highlight the effectiveness of the RF and TabNet model with the statistical feature selection method.

Table 9. Experimental results of botnet classification using statistical features.

Method	Accuracy (%)	Precision (%)	Recall (%)	F Score (%)
SMOTE_ENN				
RF	99.9951	99.9951	99.9951	99.9951
NODE	95.5402	95.7345	95.5402	93.7055
TabNet	99.9539	99.9539	99.9539	99.9539
GB	98.6170	98.8917	98.6170	98.6924
Experimental Results of ADASYN				
RF	99.9985	99.9985	99.9985	99.9985
NODE	94.1864	94.5244	94.1864	91.3786
TabNet	99.7740	99.7745	99.7740	99.7720
GB	99.5079	99.5349	99.5079	99.5152
Experimental Results of SMOTE_Tomek				
RF	99.9976	99.9976	99.9976	99.9976
NODE	39.3586	94.1423	39.3586	50.4864
TabNet	99.9544	99.9544	99.9544	99.9543
GB	98.6102	98.8396	98.6102	98.6723

Figure 5 presents the f score of an ML model for botnet attack classification using different feature selection methods. Each sub-figure's x-axis represents the data resampling method, and the y-axis represents the f-score value of each data resampling method. The f score of the ML model is visualized for each resampling method. The experimental results show that the ADASYN data resampling method has better prediction results for botnet attack classification. It shows that the RF model has higher prediction performance than the ADASYN resampling method using the statistical features. Data resampling, feature selection, and parameter tuning enable the learning model to achieve a high f-score value. Because data resampling methods enable fair class distribution, using a feature selection helps to reduce the noise in the data by selecting the relevant features.



(**a**) F score using the fused feature

(**b**) F score using the RFE features

(**c**) F score using the SFS features

(**d**) F score using the statistical features

Figure 5. F score performance of ML models by feature selection and resampling methods in botnet attack classification.

Figure 6 presents the confusion metrics of an RF model for botnet attack classification with different feature selection methods. This figure only shows the model's confusion metrics, where the RF model has a lower misclassification rate. This figure presents the effectiveness of a feature selection method for botnet attack classification. Figure 6a shows that the RF model has the highest misclassification rate with fused features. However, Figure 6d demonstrates that the misclassification rate of an RF model is more reduced with the statistical feature selection method than with the fused features.The RFE feature selection method also has fewer misclassification errors than the fused and SFS feature selection methods. The results of the RF model demonstrate that the utilization of a statistical feature selection method improves the performance of a model in terms of f score and reduces the misclassification error.

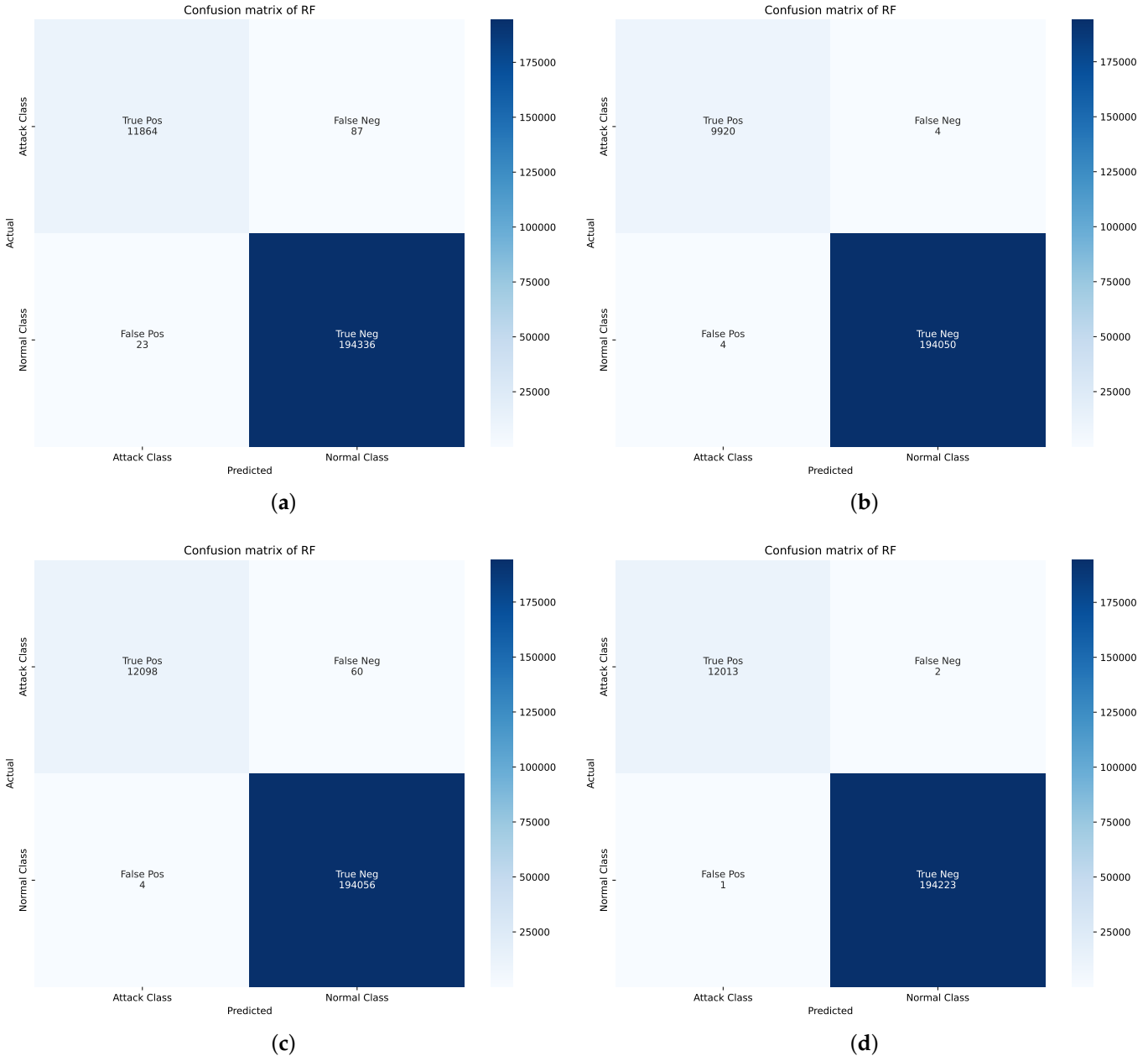


Figure 6. Confusion metrics of RF model using different feature selection method for botnet attack classification. (**a**) Confusion metrics of RF model using fused features with ADASYN resampling. (**b**) Confusion metrics of RF model using the RFF feature selection method with SMOTE_ENN resampling. (**c**) Confusion metrics of RF model using SFS feature selection method with ADASYN resampling. (**d**) Confusion metrics of RF model using statistical feature selection method with ADASYN resampling.

4.2. Experimental Results of MITM Attack Classification

This section shows the results of an ML model for MITM attack classification. In Table 10, the results of the fused feature method are presented for MITM attack classification. The RF model has higher prediction performance with the SMOTE_ENN resampling technique. Further, with an ADASYN resampling method, the performance of a RF model is decreased. However, NODE, TabNet, and GB show variability in performance across the data resampling techniques. It demonstrates that the PCA and DAE do not extract the accurate features from the MITM attack dataset.

Table 10. Experimental results of MITM attack classification using fused features.

Method	Accuracy (%)	Precision (%)	Recall (%)	F Score (%)
SMOTE_ENN				
RF	89.7609	89.7686	89.7609	89.7646
NODE	87.8296	87.7624	87.8296	87.6962
TabNet	85.2544	87.3603	85.2544	84.2683
GB	85.8982	88.3167	85.8982	84.9204
Experimental Results of ADASYN				
RF	84.0812	83.9242	84.0812	83.7798
NODE	71.4543	70.5754	71.4543	69.0156
TabNet	72.4690	77.2061	72.4690	66.9829
GB	74.1601	74.0567	74.1601	71.9525
Experimental Results of SMOTE_Tomek				
RF	87.8889	87.8329	87.8889	87.6954
NODE	81.6908	82.9196	81.6908	80.3392
TabNet	81.9045	85.3895	81.9045	79.9333
GB	81.9520	85.6751	81.9520	79.9349

Table 11 presents the results of MITM attack detection with the RFE feature selection method. With this feature selection method, the results of an RF model improve more than the fused features. It shows the effectiveness of an RF model for MITM attack detection. The NODE model shows more variability in performance, but it has poor performance in terms of f score and accuracy.

Table 11. Experimental results of MITM attack classification using RFE features.

Method	Accuracy (%)	Precision (%)	Recall (%)	F Score (%)
SMOTE_ENN				
RF	99.7069	99.7076	99.7069	99.7067
NODE	38.7496	76.3518	38.7496	26.2167
TabNet	75.6431	78.4080	75.6431	76.1825
GB	89.7428	91.1092	89.7428	89.2304
Experimental Results of ADASYN				
RF	99.9777	99.9777	99.9777	99.9777
NODE	67.6792	68.3309	67.6792	60.3809
TabNet	98.9279	98.9351	98.9279	98.9252
GB	73.4197	75.6285	73.4197	69.5236
Experimental Results of SMOTE_Tomek				
RF	99.9694	99.9694	99.9694	99.9693
NODE	79.0612	79.0394	79.0612	79.0501
TabNet	98.0585	98.0793	98.0585	98.0486
GB	95.7483	95.9207	95.7483	95.6847

Table 12 shows the experimental results of an SFS feature selection. With the ADASYN data resampling method, the RF model achieves better prediction results regarding accuracy score metrics of 99.9694%. With this feature selection method, the NODE model improves the prediction results more than the RFE feature selection method. However, the TabNet model shows more variability with all data resampling methods.

Table 12. Experimental results of MITM attack classification using SFS features.

Method	Accuracy (%)	Precision (%)	Recall (%)	F Score (%)
SMOTE_ENN				
RF	97.4095	97.4963	97.4095	97.3853
NODE	89.8252	90.7715	89.8252	89.3886
TabNet	47.5031	59.6647	47.5031	46.7274
GB	94.6385	94.9188	94.6385	93.9774
Experimental Results of ADASYN				
RF	99.6648	99.6665	99.6648	99.6644
NODE	74.1899	75.1949	74.1899	74.5118
TabNet	64.4916	77.1142	64.4916	50.6862
GB	96.2316	96.2145	96.2316	96.0030
Experimental Results of SMOTE_Tomek				
RF	95.5815	95.8015	95.5815	95.5113
NODE	82.3501	86.0670	82.3501	80.3985
TabNet	66.0112	77.6455	66.0112	53.2020
GB	95.2512	95.4270	95.2512	94.7680

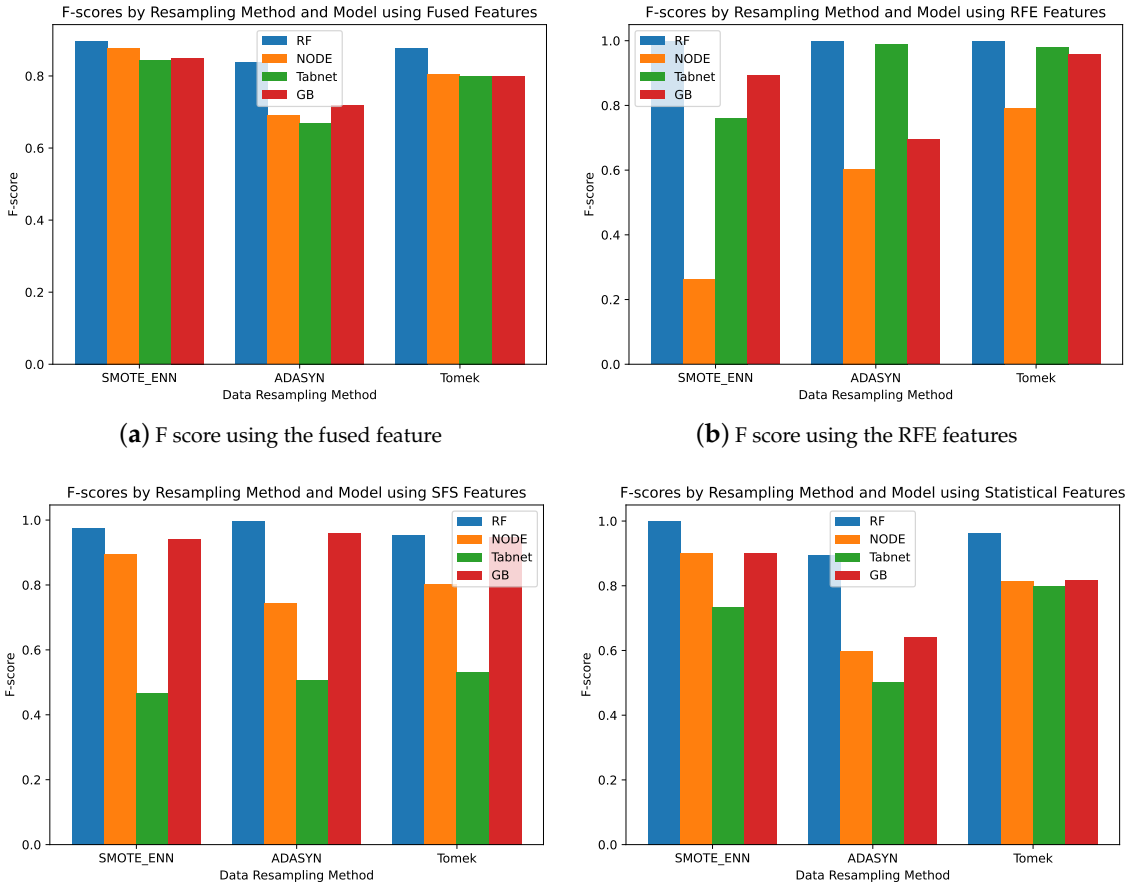
Table 13 describes the experimental results of the statistical feature selection method. The RF model performs better with the SMOTE_ENN method. With this feature selection method, the performance of an ADASYN feature selection method is slightly lower than that of the SMOTE_Tomek data re-sampling method. However, the TabNet model has a lower f score and accuracy rate than the other models.

Table 13. Experimental results of MITM attack classification using statistical features.

Method	Accuracy (%)	Precision (%)	Recall (%)	F Score (%)
SMOTE_ENN				
RF	99.9674	99.9675	99.9674	99.9674
NODE	90.3646	91.4674	90.3646	89.9474
TabNet	72.8190	75.5367	72.8190	73.4035
GB	90.3971	91.2436	90.3971	90.0286
Experimental Results of ADASYN				
RF	89.7900	90.1209	89.7900	89.5319
NODE	68.8114	78.0416	68.8114	59.9513
TabNet	64.2538	41.2855	64.2538	50.2704
GB	71.1796	80.1038	71.1796	64.0868
Experimental Results of SMOTE_Tomek				
RF	96.2399	96.3236	96.2399	96.1993
NODE	83.1408	86.4244	83.1408	81.3483
TabNet	81.7154	84.4897	81.7154	79.7624
GB	83.4357	86.7584	83.4357	81.6907

Figure 7 presents the f score of an ML model with different feature selection methods for an MITM attack classification. The x-axis of a figure represents the data resampling method, and the y-axis represents the f score of models for the data resampling technique. It shows that the RF model has a higher f-score value with the statistical and RFE feature selection methods. However, the TabNet and NODE models show more variability regarding

the f score value for all feature selection methods. Data resampling and feature selection methods enable the learning models to achieve better prediction results.



(**a**) F score using the fused feature

(**b**) F score using the RFE features

(**c**) F score using the SFS features

(**d**) F score using the statistical features

Figure 7. F score performance of ML models by feature selection and resampling methods in MITM attack classification.

Figure 8 shows the confusion metrics of an RF model for MITM attack classification. In this figure, we only represent confusion, where we have a lower classification rate of the feature selection method. The results in Figure 8b,d show that the RF model achieves a low classification error with the RFE and statistical feature selection methods. However, the fused features and SFS feature selection method have a higher misclassification error than the RFE and statistical feature selection method. The presented results show the effectiveness of the feature selection method in reducing the misclassification error for MITM attack classification.

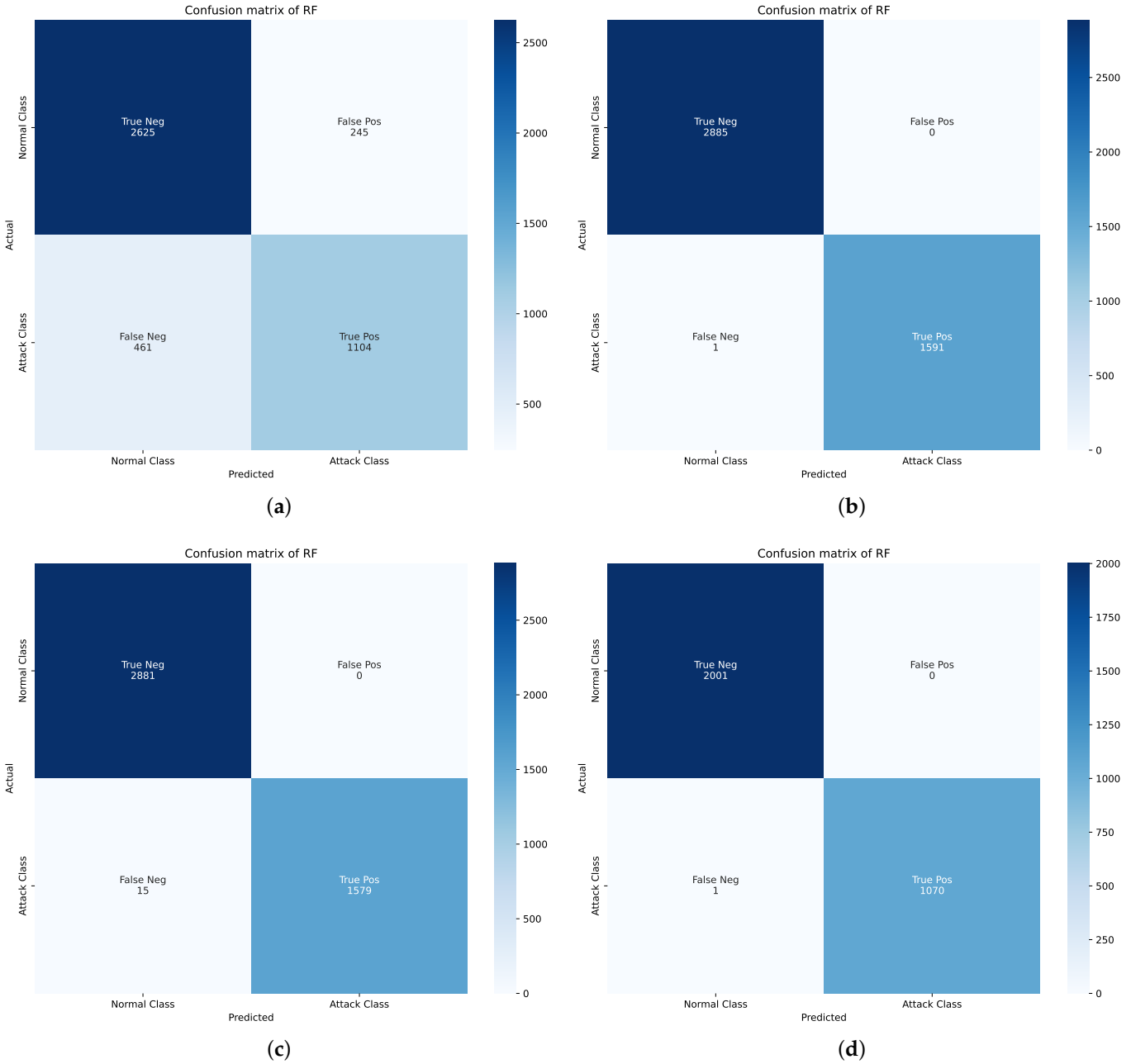


Figure 8. Confusion metrics of RF model using different feature selection method for MITM attack classification. (**a**) Confusion metrics of RF model using fused features. (**b**) Confusion metrics of RF model using the RFF feature selection method. (**c**) Confusion metrics of RF model using SFS feature selection method. (**d**) Confusion metrics of RF model using statistical feature selection method.

4.3. Comparative Analysis of a Proposed Method with the Existing Methods

Works in the literature have applied many methods to classify a botnet and MITM attack. However, the existing techniques have lower prediction performance and high misclassification rates. Table 14 presents the results of the existing and proposed method for botnet and MITM attack classification. The experimental findings of the proposed work demonstrate an improved performance that surpasses that of existing approaches.

Table 14. Comparative analysis of proposed method with the existing methods.

Method	Accuracy
UNSW Dataset	
ANN [46]	99.42%
DL-RL [47]	97.29%
Proposed method (RF with statistical feature and ADASYN)	99.9985
WUSTL Dataset	
CNN-LSTM [48]	99%
MLP [49]	96.39%
CNN-Focal [50]	93.08%
Proposed method (RF with RFE and ADASYN)	99.9777

Table 14 concludes that the proposed strategy has significantly improved prediction results on the WUSTL and UNSW datasets than the existing methods. This study achieved these better and improved results by utilizing the feature selection, data resampling, and optimization of hyperparameters that enabled the learning model to capture the complex pattern of network traffic data. In the proposed strategy, the feature selection method selects the relevant features that reduce the noise and probability of over-fitting. Further, using the data resampling method enables fair learning in class distribution by balancing the class distribution of WUSTL and UNSW data.

5. Conclusions

The intrusion detection system is crucial in safeguarding network infrastructures against cyber threats. However, they frequently encounter challenges such as elevated false positive rates and model bias caused by imbalanced data and irrelevant feature sets. This study utilized SFS, RFE, statistical feature selection methods and SMOTE_ENN, SMOTE_Tomek, and ADASYN data resampling methods to enhance the performance of intrusion detection. A feature selection method aims to reduce the feature of in-hand data and select the subsets of a feature that contribute to improving the performance of a model. The purpose of a data resampling method is to balance the data distribution and reduce the possibility of a model over-fitting towards the majority class. This study applied the TabNet, RF, NODE, and GB models to pre-processed data and performed the experiments on WUSTL and UNSW datasets. The experimental result shows that the RFE and statistical feature selection methods have better prediction results than the ADASYN data resampling method. Amongst the learning models, the RF model has fewer misclassification errors on both datasets, reducing the misclassification errors compared to the existing methods. The findings of the proposed study conclude that we can utilize the RF model with statistical feature selection and ADASYN data resampling methods to analyze real-time traffic in the IoT network and track abnormal traffic. In the future, we can utilize the federated learning paradigm to consider the non-independent and identically distributed environment, where some clients have some attack classes and are absent from some classes. We collaboratively learn from other clients about these absence classes.

Author Contributions: Conceptualization, F.M. and Q.W.K.; methodology, F.M., Q.W.K. and R.A.; software, F.M.; validation, G.A. and R.A.; formal analysis, Q.W.K., A.R. and F.M.; investigation, Q.W.K. and F.M.; resources, G.A. and R.A.; data curation, Q.W.K., A.R. and F.M.; writing—original draft preparation, F.M., Q.W.K. and R.A; writing—review and editing, R.A. and G.A.; visualization, G.A.; supervision, G.A. and R.A.; project administration, R.A. All authors have read and agreed to the published version of the manuscript.

Funding: Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2024R408), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia

Data Availability Statement: Data is available.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

IDS	Intrusion Detection System
RFE	Recursive Feature Elimination
SFS	Sequential Feature Selection
SMOTE_ENN	Synthetic Minority Oversampling Technique and Edited Nearest Neighbor
ADASYN	Adaptive Synthetic Sampling
NODE	Neural Oblivious Decision Ensembles
GB	Gradient Boosting
RF	Random Forest
MITM	Man-in-the-Middle
ML	Machine Learning
IoT	Internet of Things
DL	Deep Learning
LDA	Linear Discriminant Analysis
KNN	k-Nearest Neighbor
SVM	Support Vector Machine
WSN	Wireless Sensor Network
BI-LSTM	Long Short-term Memory Network
TCN	Temporal Convolutional Network
SSAE	Stacked Sparse Autoencoder
CNN	Convolutional Neural Network

References

1. Rahman, Z.; Haque, M.A.; Aziz, D.A.B. Internet Usage During and Post COVID-19 Pandemic: A Study on the Students of Information Science and Library Management in the University of Rajshahi, Bangladesh. *Libr. Philos. Pract.* **2023**, 1–15. Available online: https://digitalcommons.unl.edu/libphilprac/7621/ (accessed on 28 April 2024).
2. Javaid, M.; Haleem, A.; Singh, R.P.; Suman, R. Towards insighting cybersecurity for healthcare domains: A comprehensive review of recent practices and trends. *Cyber Secur. Appl.* **2023**, *1*, 100016. [CrossRef]
3. Liu, X.; Ahmad, S.F.; Anser, M.K.; Ke, J.; Irshad, M.; Ul-Haq, J.; Abbas, S. Cyber security threats: A never-ending challenge for e-commerce. *Front. Psychol.* **2022**, *13*, 927398. [CrossRef]
4. Aswathy, S.; Tyagi, A.K. Privacy Breaches through Cyber Vulnerabilities: Critical Issues, Open Challenges, and Possible Countermeasures for the Future. In *Security and Privacy-Preserving Techniques in Wireless Robotics*; CRC Press: Boca Raton, FL, USA, 2022; pp. 163–210.
5. Arogundade, O.R. Network security concepts, dangers, and defense best practical. *Comput. Eng. Intell. Syst.* **2023**, *14*, 25–38.
6. Vaigandla, K.; Azmi, N.; Karne, R. Investigation on intrusion detection systems (IDSs) in IoT. *Int. J. Emerg. Trends Eng. Res.* **2022**, *10*, 158–166.
7. Bediya, A.K.; Kumar, R. A novel intrusion detection system for internet of things network security. In *Research Anthology on Convergence of Blockchain, Internet of Things, and Security*; IGI Global: Hershey, PA, USA, 2023 ; pp. 330–348.
8. Thakkar, A.; Lohiya, R. A survey on intrusion detection system: Feature selection, model, performance measures, application perspective, challenges, and future research directions. *Artif. Intell. Rev.* **2022**, *55*, 453–563. [CrossRef]
9. Momand, A.; Jan, S.U.; Ramzan, N. A systematic and comprehensive survey of recent advances in intrusion detection systems using machine learning: Deep learning, datasets, and attack taxonomy. *J. Sensors* **2023**, *2023*, 6048087. [CrossRef]
10. Ponnusamy, V.; Yichiet, A.; Jhanjhi, N.; Almufareh, M.F. IoT wireless intrusion detection and network Traffic Analysis. *Comput. Syst. Sci. Eng.* **2022**, *40*, 865. [CrossRef]
11. Umar, M.A.; Chen, Z.; Shuaib, K.; Liu, Y. Effects of feature selection and normalization on network intrusion detection. *Authorea Prepr.* **2024**. [CrossRef]
12. Latif, S.; Dola, F.F.; Afsar, M.; Esha, I.J.; Nandi, D. Investigation of Machine Learning Algorithms for Network Intrusion Detection. *Int. J. Inf. Eng. Electron. Bus.* **2022**, *14*, 1–22. [CrossRef]
13. Thamilarasu, G.; Chawla, S. Towards deep-learning-driven intrusion detection for the internet of things. *Sensors* **2019**, *19*, 1977. [CrossRef] [PubMed]
14. Pajouh, H.H.; Javidan, R.; Khayami, R.; Dehghantanha, A.; Choo, K.K.R. A two-layer dimension reduction and two-tier classification model for anomaly-based intrusion detection in IoT backbone networks. *IEEE Trans. Emerg. Top. Comput.* **2016**, *7*, 314–323. [CrossRef]

15. Yahyaoui, A.; Abdellatif, T.; Attia, R. Hierarchical anomaly based intrusion detection and localization in IoT. In Proceedings of the 2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC), Tangier, Morocco, 24–28 June 2019; pp. 108–113.
16. Zhang, Y.; Li, P.; Wang, X. Intrusion detection for IoT based on improved genetic algorithm and deep belief network. *IEEE Access* **2019**, *7*, 31711–31722. [CrossRef]
17. Osa, E.; Orukpe, P.E.; Iruansi, U. Design and implementation of a deep neural network approach for intrusion detection systems. *E-Prime Electr. Eng. Electron. Energy* **2024**, *7*, 100434. [CrossRef]
18. He, Z.; Wang, X.; Li, C. A Time Series Intrusion DetectionMethod Based on SSAE, TCN and Bi-LSTM. *Comput. Mater. Contin.* **2024**, *78*. [CrossRef]
19. Alotaibi, A.; Rassam, M.A. Adversarial machine learning attacks against intrusion detection systems: A survey on strategies and defense. *Future Internet* **2023**, *15*, 62. [CrossRef]
20. Aljehane, N.O.; Mengash, H.A.; Eltahir, M.M.; Alotaibi, F.A.; Aljameel, S.S.; Yafoz, A.; Alsini, R.; Assiri, M. Golden jackal optimization algorithm with deep learning assisted intrusion detection system for network security. *Alex. Eng. J.* **2024**, *86*, 415–424. [CrossRef]
21. Akhiat, Y.; Touchanti, K.; Zinedine, A.; Chahhou, M. IDS-EFS: Ensemble feature selection-based method for intrusion detection system. *Multimed. Tools Appl.* **2024**, *83*, 12917–12937. [CrossRef]
22. Nanjappan, M.; Pradeep, K.; Natesan, G.; Samydurai, A.; Premalatha, G. DeepLG SecNet: Utilizing deep LSTM and GRU with secure network for enhanced intrusion detection in IoT environments. *Clust. Comput.* **2024**, 1–13. [CrossRef]
23. Ahmed, A.A.; Jabbar, W.A.; Sadiq, A.S.; Patel, H. Deep learning-based classification model for botnet attack detection. *J. Ambient Intell. Humaniz. Comput.* **2022**, *13*, 3457–3466. [CrossRef]
24. Putra, M.A.R.; Ahmad, T.; Hostiadi, D.P. B-CAT: A model for detecting botnet attacks using deep attack behavior analysis on network traffic flows. *J. Big Data* **2024**, *11*, 49. [CrossRef]
25. Alshaeaa, H.Y.; Ghadhban, Z.M. Developing a hybrid feature selection method to detect botnet attacks in IoT devices. *Kuwait J. Sci.* **2024**, *51*, 100222. [CrossRef]
26. Dash, S.K.; Dash, S.; Mahapatra, S.; Mohanty, S.N.; Khan, M.I.; Medani, M.; Abdullaev, S.; Gupta, M. Enhancing DDoS attack detection in IoT using PCA. *Egypt. Inform. J.* **2024**, *25*, 100450. [CrossRef]
27. Soliman, S.; Oudah, W.; Aljuhani, A. Deep learning-based intrusion detection approach for securing industrial Internet of Things. *Alex. Eng. J.* **2023**, *81*, 371–383. [CrossRef]
28. Almazroi, A.A.; Ayub, N. Deep learning hybridization for improved malware detection in smart Internet of Things. *Sci. Rep.* **2024**, *14*, 7838. [CrossRef]
29. Angelin, J.A.B.; Priyadharsini, C. Deep Learning based Network based Intrusion Detection System in Industrial Internet of Things. In Proceedings of the 2024 2nd International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT), Bengaluru, India, 4–6 January 2024; pp. 426–432.
30. Koroniotis, N.; Moustafa, N.; Sitnikova, E.; Turnbull, B. Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset. *Future Gener. Comput. Syst.* **2019**, *100*, 779–796. [CrossRef]
31. Turukmane, A.V.; Devendiran, R. M-MultiSVM: An efficient feature selection assisted network intrusion detection system using machine learning. *Comput. Secur.* **2024**, *137*, 103587. [CrossRef]
32. Sharma, N.V.; Yadav, N.S. An optimal intrusion detection system using recursive feature elimination and ensemble of classifiers. *Microprocess. Microsystems* **2021**, *85*, 104293. [CrossRef]
33. Polat, H.; Polat, O.; Cetin, A. Detecting DDoS attacks in software-defined networks through feature selection methods and machine learning models. *Sustainability* **2020**, *12*, 1035. [CrossRef]
34. Thakkar, A.; Lohiya, R. Fusion of statistical importance for feature selection in Deep Neural Network-based Intrusion Detection System. *Inf. Fusion* **2023**, *90*, 353–363. [CrossRef]
35. Song, Y.; Hyun, S.; Cheong, Y.G. Analysis of autoencoders for network intrusion detection. *Sensors* **2021**, *21*, 4294. [CrossRef]
36. Almaiah, M.A.; Almomani, O.; Alsaaidah, A.; Al-Otaibi, S.; Bani-Hani, N.; Hwaitat, A.K.A.; Al-Zahrani, A.; Lutfi, A.; Awad, A.B.; Aldhyani, T.H. Performance investigation of principal component analysis for intrusion detection system using different support vector machine kernels. *Electronics* **2022**, *11*, 3571. [CrossRef]
37. Khan, Q.W.; Ahmad, R.; Rizwan, A.; Khan, A.N.; Park, C.W.; Kim, D. Multi-modal fusion approaches for tourism: A comprehensive survey of data-sets, fusion techniques, recent architectures, and future directions. *Comput. Electr. Eng.* **2024**, *116*, 109220. [CrossRef]
38. Bagui, S.; Mink, D.; Bagui, S.; Subramaniam, S.; Wallace, D. Resampling Imbalanced Network Intrusion Datasets to Identify Rare Attacks. *Future Internet* **2023**, *15*, 130. [CrossRef]
39. Abdelmoumin, G.; Rawat, D.B.; Rahman, A. Studying Imbalanced Learning for Anomaly-Based Intelligent IDS for Mission-Critical Internet of Things. *J. Cybersecur. Priv.* **2023**, *3*, 706–743. [CrossRef]
40. Bagui, S.; Li, K. Resampling imbalanced data for network intrusion detection datasets. *J. Big Data* **2021**, *8*, 6. [CrossRef]
41. Sams Aafiya Banu, S.; Gopika, B.; Esakki Rajan, E.; Ramkumar, M.; Mahalakshmi, M.; Emil Selvan, G. SMOTE Variants for Data Balancing in Intrusion Detection System Using Machine Learning. In Proceedings of the International Conference on Machine Intelligence and Signal Processing, Raipur, India, 12–14 March 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 317–330.

42. Alshamy, R.; Ghurab, M.; Othman, S.; Alshami, F. Intrusion detection model for imbalanced dataset using SMOTE and random forest algorithm. In Proceedings of the Advances in Cyber Security: Third International Conference (ACeS 2021), Penang, Malaysia, 24–25 August 2021; Revised Selected Papers 3; Springer: Berlin/Heidelberg, Germany, 2021; pp. 361–378.
43. Mishra, S. An optimized gradient boost decision tree using enhanced African buffalo optimization method for cyber security intrusion detection. *Appl. Sci.* **2022**, *12*, 12591. [CrossRef]
44. Arik, S.Ö.; Pfister, T. Tabnet: Attentive interpretable tabular learning. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtually, 2–9 February 2021; Volume 35, pp. 6679–6687.
45. Popov, S.; Morozov, S.; Babenko, A. Neural oblivious decision ensembles for deep learning on tabular data. *arXiv* **2019**, arXiv:1909.06312.
46. Kerrakchou, I.; Abou El Hassan, A.; Chadli, S.; Emharraf, M.; Saber, M. Selection of efficient machine learning algorithm on Bot-IoT dataset for intrusion detection in internet of things networks. *Indones. J. Electr. Eng. Comput. Sci.* **2023**, *31*, 1784–1793. [CrossRef]
47. Zaman, S.; Iqbal, M.M.; Tauqeer, H.; Shahzad, M.; Akbar, G. Trustworthy communication channel for the iot sensor nodes using reinforcement learning. In Proceedings of the 2022 International Conference on Emerging Trends in Electrical, Control, and Telecommunication Engineering (ETECTE), Lahore, Pakistan, 2–4 December 2022; pp. 1–6.
48. Ravi, V.; Pham, T.D.; Alazab, M. Deep Learning-Based Network Intrusion Detection System for Internet of Medical Things. *IEEE Internet Things Mag.* **2023**, *6*, 50–54. [CrossRef]
49. Judith, A.; Kathrine, G.J.W.; Silas, S. Efficient Deep Learning-Based Cyber-Attack Detection for Internet of Medical Things Devices. *Eng. Proc.* **2023**, *59*, 139. [CrossRef]
50. Dina, A.S.; Siddique, A.; Manivannan, D. A deep learning approach for intrusion detection in Internet of Things using focal loss function. *Internet Things* **2023**, *22*, 100699. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.