Ensemble Feature Selection for Network Intrusion Detection: Combining Information Gain and Random Forest with Recursive Feature Elimination

Stephen Kahara Wanjau
School of Computing and Information Technology
Murang’a University of Technology
Murang’a, Kenya
Email: steve.kahara [AT] gmail.com

Gabriel Ndung’u Kamau
School of Computing and Information Technology
Murang’a University of Technology
Murang’a, Kenya
Email: kamau.gabriel [AT] gmail.com

Abstract—**Network intrusion detection systems (NIDS) are essential for protecting computer networks against cyberattacks. The selection of a nominal set of essential features that may adequately discriminate malicious traffic from the normal traffic is indispensable while developing a NIDS. As such, a more reliable and accurate detection result may be realized when intrusion detection is carried out on a dataset based on an inclusive feature representation. This work presents the pre-processing and feature selection workflow as well as its results in the case of the CIC-IDS-2017 dataset with a focus on two cyber-attacks namely Denial-of-Service (DoS) and PortScan. The study applied an ensemble feature selection method based on information gain and Random Forest to filter out important features. Recursive Feature Elimination method was then applied to the reduced features to optimize the selected feature subset. The selected feature subset was experimented with using two classification algorithms, namely support vector machine and multi-layer perceptron. In the evaluation process, four widely used performance metrics were considered. The study results demonstrated the efficacy of the proposed ensemble approach to optimize the selected feature subset for detecting PortScan and DoS attacks in network traffic. Experimental results revealed that the support vector machine had a slight advantage in accuracy and could train more quickly. According to the study's evaluation, the NIDS may be able to shorten processing times without sacrificing the ability to detect PortScan and DoS attacks accurately by choosing a narrow subset of informative features. This suggests the approach might be applicable to real-world NIDS scenarios involving these attacks. The study also provides encouraging perspectives on how ensemble feature selection utilizing MLP and SVM can enhance the effectiveness of NIDS. Building on these findings, more research can create NIDS solutions that are even more reliable and efficient for the dynamic field of cybersecurity.**

Keywords- Classification; ensemble; feature selection; network intrusion detection system; pre-processing; recursive feature elimination.

I. INTRODUCTION

Cybersecurity is an important concern for organizations globally owing to the high cost associated with security threats and vulnerabilities. Computing products used in these organizations, ranging from small mobile devices to large cloud-based platforms, are practically all networked, and susceptible to network intrusion. Such organizations are utilizing Network Intrusion Detection Systems (NIDSs), which are capable of efficiently detecting attacks against their information assets. The NIDSs constantly monitor and evaluate the network parameters with the goal of detecting malicious activities likely to damage computer systems or even the network. When malicious activities are detected, the NIDS triggers an alarm message that is sent to a pre-configured monitoring system capable of preventing any additional attacks by re-configuring network devices [1].

Existing network intrusion detection approaches are largely classified as either anomaly-based or signature-based contingent on the methods of intrusion detection [2, 3]. Signature-based detection systems (also known as misuse-based NIDS) are subject to pre-defined signatures and filters when determining network attacks. These detection systems greatly rely exclusively on human input while executing regular updates on the rules and signature database [4]. The main advantage of this method is that it accurately discovers renowned attacks. However, it is largely ineffective in detecting any new or unknown attacks [5]. On the other hand, anomaly-based detection systems (also known as behaviour-based NIDS) leverage on heuristic mechanisms to identify and process unknown malignant activities [6] distinguishing between normal and abnormal behaviour. Anomaly-based detection algorithms are normally exploited to model patterns of normal behaviour for devices and networks and proceed to scrutinize patterns that deviate from normal behaviour at a much faster pace [7].

The growing complexity of cyber threats demands continuous improvement in NIDS. Nevertheless, the "curse of dimensionality," which impairs NIDS performance, can result from high-dimensional network traffic data. To overcome this difficulty, feature selection techniques pinpoint a subset of features that considerably improve the accuracy of intrusion detection. This paper reports on a study that focused on anomaly-based NIDSs that are well-known to operate in two key modes; learning mode and detection model. In the learning mode, the NIDS is provided with data containing benign network traffic and malignant attack data. The classification unit is trained and tested using the labels linked to the data records. Thereafter, in detection mode, the fully trained classification module is used to determine whether or not the current activity is harmful.

As a whole, the classification module of a NIDS is the most significant component. Nonetheless, its speed and efficiency are to a large extent affected by the identification and accurate selection of a nominal set of essential features that are monitored and used to perform the classification [8]. The study focused on determining these features in the case of the CIC-IDS-2017 dataset, a benchmark network intrusion detection dataset that comprises both normal network traffic data as well as data associated with different attack types. Specifically, the focus was on PortScan and Denial-of-Service (DoS) attacks. These attacks are the most common and widely recognized threats to network security. They can cause major disruptions to regular operations and pose serious risks to network availability. Attackers using port scanning probe ports to find open services on a network. Selecting features that prioritize detecting odd connection attempts and service requests is necessary when analyzing network traffic patterns for PortScan detection. The goal of denial-of-service (DoS) attacks is to flood a system with traffic so that legitimate users are unable to access it. Finding unusual traffic volumes, packet characteristics, or explicit protocols used in these attacks could be the key focus of feature selection for DoS detection.

The rank of each feature for the attack type was determined based on the average score attained from the results of the application of the three feature selection methods. In addition, two classification algorithms were utilized in evaluating a series of rank threshold values to determine the optimal threshold and feature set for each attack type.

The rest of the paper is organized as follows: Section 2 presents a review of related works on feature selection methods for intrusion detection systems. Section 3 describes feature selection methods. Section 4 presents the proposed methodology used, the dataset and data pre-processing, classification algorithms used and the performance evaluation metrics. In Section 5, the experiment results are discussed. Section 6 draws conclusions and makes recommendations for future work.

II. Related Work

A review of related work in the literature reveals that several feature selection methods used to rank features for use in the development of NIDSs have been investigated intensively in the last five years. The work of [8] reported six feature evaluation techniques which were performed on CSE-CIC-IDS 2018 dataset where an average score was computed after normalization and used to rank individual features. Six ranking thresholds were defined, which led to the selection of several relevant feature collections for each attack type. Pearson Linear Correlation and Information Gain methods were used by Javadpour, et al. [9] to select the features of the KDD99 dataset. The researchers used Artificial Neural Network (ANN), Random Forest (RF), CART, and Decision Tree algorithms for classification with ANN obtaining the best result (99.98% accuracy). Ren, et al. [10] proposed a network intrusion detection model (ID-RDRL) that used Decision Tree-based Recursive Feature Elimination (RFE) and deep reinforcement learning. The RFE feature selection technique was used to filter the optimum subset of features from the CSE-CIC-IDS2018 dataset and train a classifier using DRL to recognize network intrusions. The proposed model achieved an accuracy of 96.2% and an F1-score of 94.9%, respectively.

Ali, et al. [11] adopted the correlation-based feature selection (CFS) and classifier subset evaluation methods to select the relevant features from the CIC-IDS-2017 dataset. The CFS method reduced the total 78 attributes to 5 attributes, whereas the classifier subset evaluation method reduced the attributes to 3. Then, Multi-Layer Perceptron (MLP) and Instance-Based Learning (IBK) algorithms were applied to a reduced number of features, where the IBK performed better than MLP. Swe, et al. [12] used gain-ratio and chi-squared ranking methods to select optimal feature subset from the CIC-IDS 2017 and CSE-CIC-IDS 2018 datasets for training a DDoS attacks intrusion detection model. The experimental results showed that the proposed mechanism could detect slow rate attacks with 99% accuracy and with very low false-negative rate. Patil & Kshirsagar [13] used two-step hybrid feature selection method on the CICIDS 2017 dataset containing 84 features. Information gain, gain ratio, and correlation filter-based algorithms were utilized to rank features and then the forward selection approach was used to reduce the features up to 32. The experiment yielded a higher accuracy of 88.7373% in classifying DDoS attack. In a relatively recent study, Kshirsagar & Kumar [14] proposed an ensemble of filter feature selection techniques to obtain a significant feature subset for web attack detection and selected one-fourth split of the ranked features using Information Gain, Gain Ratio, Correlation coefficient and Relief. Experiments were conducted on the CICIDS 2017 dataset where the proposed technique yielded a detection rate of 99.9909%, with J48 algorithm using 24 features.

In their work, Pelletier & Abualkibash [15] ran the CIC-IDS-2017 dataset through an automated test of importance to determine the relative importance of each individual network feature in the dataset using Boruta package. Their study identified the top 10 most important features from this determination for use in design the predictive intrusion detection model. The study by [16] utilized Information Gain (IG) method to rank and cluster features contained in the CIC-IDS-2017 dataset. The authors applied J48, Bayes Net (BN), Random Forest (RF), Random Tree (RT) and Naive Bayes (NB) classification algorithms in the selection of features, which generated worthy classification results. The work of Reis, et al. [17] used several feature selection and ensemble methods on the CIC-IDS-2017 dataset to develop valid models to detect intrusions as soon as they occur. The authors used permutation importance to reduce the original 69 dataset features to only 10 features, which permitted the reduction of Random Forest based model execution time, thus leading to a faster intrusion detection system. Despite their reported success, these methods may not capture complex non-linear relationships between features that could be crucial for identifying intrusions.

III. Overview of Feature Selection

High-dimensional data is a term used to describe datasets with numerous characteristics; it has also drawn more attention from researchers [18]. Complex data engulfs effective information, making it challenging to identify the key elements of the data. The current challenge is to extract meaningful reduction data from the high-dimensional data set while preserving the essential features of the original data to satisfy recognition accuracy and storage requirements [19]. Feature selection, also known as variable selection, or feature subset selection refers to the practice of removing unnecessary, repetitive, or noisy features from the original features so as to select a small subset of the pertinent features. [20, 21]. The large number of redundant and irrelevant data in network traffic brings serious challenges to intrusion detection [22] besides the curse of dimensionality that often leads to increased costs of data storage and computing [23]. Therefore, feature selection is aimed at selecting an optimal subset of features (based on a specified criterion) from the initial set, where two steps are normally performed: The first one involves a search strategy to pinpoint candidate subsets; the second one involves an objective function to evaluate the selected candidate subsets.

Researchers have continually focused on feature selection as one of the important tasks of data pre-processing that is essential for the efficiency and performance of model training [19]. According to Ren, et al. [10] feature selection methods may be categorized into three groups: wrapper, filtering and embedding. A filter method utilizes static measures to compute a score for each feature whereby the decision to either select or reject feature from dataset is dependent on the feature score. Examples include, information gain, Chi-squared test, and correlation coefficient score. A wrapper method works in a comparable manner to a search problem whereby features are prepared in different combination, evaluated, and compared to other combinations. Typically, a predictive model is employed to assign a score based on the accuracy of a model while evaluating the features. The search process could be heuristics, such as forward and backward pass, or may be stochastic in nature, for instance, random hill-climbing algorithm to add and remove features. An example of a wrapper method is a recursive feature elimination algorithm. On the other hand, an embedded method evaluates each feature in the dataset that increases the accuracy of a model while it is being created.

The study by Yin, et al. [24] proposed IGRF-RFE for intrusion detection, a feature reduction strategy based on a combination of two filter methods, information gain and random forest (RF) respectively, to reduce the feature subset search space. Then, a machine learning-based wrapper method that provides a recursive feature elimination was used to further reduce feature dimensions in the UNSW-NB15 dataset while considering the relevance of similar features. The features were reduced from 42 to 23 with multi-classification accuracy of MLP improving from 82.25% to 84.24%.

In another study, Patgiri, et al. [25] used support vector machine (SVM) and random forest in combination with recursive feature elimination to choose features from the NSL-KDD dataset and evaluated the two machine learning algorithms for intrusion detection. From the literature, a comparative analysis of the aforementioned feature selection methods, has demonstrated that the wrapper method, Recursive Feature Elimination may iteratively select feature subsets and is better appropriate for NIDS datasets that contains colossal data volume and numerous features.

The proposed study proposes a three-step ensemble feature selection approach: Initial Feature Selection with Information Gain, Feature Importance Ranking with Random Forest, and Feature Subset Optimization with Recursive Feature Elimination. The following is a discussion of this approach.

A. *Ensemble Random Forest and Information Gain Feature Subset Selection*

Information gain is a filter method that is based on information entropy [26]. There may be features that are highly skewed or contain little information, particularly when working with high-dimensional datasets that normally impact performance in machine learning. The information gain feature selection takes the amount of information as the importance metric by calculating the information entropy of each feature in classification tasks. The information gain of a feature is equivalent to the entropy of the class label minus the conditional entropy of the class label under the feature [24].

Random Forest is a machine learning algorithm based on multiple decision trees that is regularly used for classification and regression tasks [27]. The algorithm combines several randomized decision trees and aggregates their predictions by averaging, thus avoiding overfitting and achieving better generalizability. When used as a classifier, the random forest first establishes how many trees to construct, and then it generates a random subset of the data for each decision tree using the bootstrap sampling technique. After training, using a voting mechanism based on each tree's prediction, the classifier produces a prediction with a higher probability. The importance score for each feature is provided by the resulting model, which can also be used as an embedded feature selection method using the random forest algorithm. As a result, the most crucial features can be chosen, and the features that have no bearing on the model's performance are eliminated. The random forest's feature importance is primarily dependent on the decision trees' node impurity property. A decision tree is generated by determining a feature's position and priority in each node based on entropy or the Gini index. Higher feature importance corresponds with less impurity, which is indicated by a lower entropy or Gini index. Each tree's impurity is calculated, and an average importance score is produced.

The ensemble feature selection, which combines information gain and random forest importance is considered the first step in the feature selection. The ensemble method is applied to all the 78 features in the dataset and computes the importance of each feature using information gain and random forest respectively. The importance scores were normalized to a value between 0 and 1. By ranking and visualizing the importance scores, the study set a threshold of 0.3 and 0.02 for the two feature selection methods respectively to filter the important features. Only the feature whose importance was larger than the threshold was retained.

A. *Feature Subset Optimization Using Recursive Feature Elimination*

Recursive Feature Elimination (RFE) is a wrapper method that fits a model and determines how significant features explain the variation in the dataset [28]. RFE uses an iterative process that removes the least significant features from the initial features to create a set of candidate subsets during training. When using RFE, there are two key configuration options available: selecting the number of features to choose from and selecting the algorithm to assist in feature selection [29]. The desired number of features are specified by the researcher as a stopping criterion. This number may be determined by domain expertise, computational efficiency considerations, or by tracking the model's performance as features are removed. Feature ranking and selection for elimination in each iteration of RFE are done using an algorithm. Feature importance scores calculated by a Random Forest model are used in the method suggested in this paper.

As soon as the feature importance has been determined, RFE gets rid of the less important features one by one in each iteration. This iterative process continues repeatedly until a definite threshold (optimal number of features needed) is attained. According to Darst, et al. [28] leveraging a machine learning algorithm and an importance-ranking metric, RFE evaluates each of the feature's impact on model performance.

IV. Methodology

The methodology and experimental setup for assessing the suggested ensemble feature selection strategy for the NIDS are described in this section. The outcomes shed light on the advantages of combining Recursive Feature Elimination, Random Forest, and Information Gain when choosing a smaller feature set that improves computational efficiency and NIDS performance.

B. *Research Design*

The study assesses the efficacy of an ensemble feature selection technique for the NIDS using an empirical methodology. Experimental research design was used in this study as the study required careful planning and control to ensure the results are robust and meaningful. The research design uses a scientific method to conduct the research and quantitative data is collected and used to perform statistical analysis during the study process. The efficacy of the suggested ensemble approach was measured using the following performance metrics accuracy, precision, recall, F1-score, and detection rate.

C. *Experimental Environment and Setup*

Randomized experimental setups were implemented to provide the highest levels of internal validity [29]. Stratified random sampling was done to ensure that each class (attack and benign traffic) are well represented in each sample.

Information-gathering experiments were conducted on an Intel Core i5 2.50 GHz CPU and 16 GB of RAM machine running on Microsoft Windows 10, Professional. The Anaconda Integrated Development Environment (IDE) pre-processes, analyses and creates the predictive models. Python 3.8 was used as the experimental programming environment and the MLP and SVM models were created on TensorFlow 2.4.1. Scikit-Learn, NumPy, Pandas, and Matplotlib packages provided data processing, feature selection, and visualization functions for the experiments.

D. Dataset

The CICIDS-2017 dataset was constructed by abstracting the behaviour of 25 users across a range of network protocols, collected using the NetFlowMeter Network Traffic Flow analyser and provided by the Canadian Institute of Cybersecurity [30]. The dataset resembles real-world data and contains network traffic data for both benign and 8 types of attacks that were collected over a span of 5 days.

TABLE 1: STATISTICS OF THE CIC-IDS 2017 DATASET

S/No.	Class	Records
1	Benign (Normal flow)	2,273,097
2	DDOS	128,027
3	DOS	252,661
4	Bot	1966
5	Patator (SSH & FTP)	13,835
6	Heartbleed	11
7	Infiltration	36
8	PortScan	158,930
9	Web Attacks	2180
	Total	**2,830,743**

Source: Zhang, et al. [22]

E. Data Pre-processing

The study considered only the DOS and PortScan attacks. The comma separated values (CSV) format of the CIC-IDS-2017 dataset was used for all the processing. During data pre-processing, several techniques were performed including data cleaning, oversampling, encoding, and normalization of the dataset. Then, the dataset was divided into a training set, a validation set, and a test set. Both the training and validation sets were used in the feature selection and training process while the test set was used to verify the final performance of the model.

Data cleaning: In total, there are 3,119,345 network flows labelled as one of the classes, each with 84 features. Among them, flowid, sourceIP, sourceport, destinationIP, destinationport and timestamp are features used for manual labelling of the flow and they do not contain any information about the content. Hence, these 6 data columns were excluded and the experiments carried out considering the remaining 78 features.

Oversampling: Over sampling is used in cases where the amount of data collected is insufficient. To improve the class balance between the benign and intrusion samples in the dataset that would subsequently yield better performance of the model, the Synthetic Minority Oversampling Technique (SMOTE) proposed by Chawla, et al. [31] and Batista, et al. [32] that selects features in close feature space was used.

Encoding: One Hot Encoding was used to transform all the categorical features into binary vectors for training and testing purposes. The categorical values were mapped to an integer value, and then, each integer represented in binary vector.

Normalization: Given that some features have very large scope in the difference between the minimum and maximum values, a logarithmic scaling method was applied for scaling to obtain the features, which are mapped to a range of (0, 1). The min-max scaling method [33] was used for normalization. The transformation parameters (maximum and minimum values of each feature) obtained were used to apply the same scaling on the features of the training, validation, and test sets.

The final step in the pre-processing was data splitting such that each resulting file contained only the records that match with one attack type as well as records describing normal traffic. In this case, the operation resulted in three data files as shown in Table 2.

TABLE 2: DATASETS GENERATED DURING DATA PRE-PROCESSING

S/No.	File Name	Number of columns	Number of Records
1	dataset-benign.csv	78	2,273,097
2	dataset-portscan.csv	78	158,930
3	dataset-dos.csv	78	252,661

For model training and testing, the datasets were split into two parts, i.e., 70% for training and the remaining 30% for testing. Further, the training samples were added 20% of records belonging to the benign traffic and the test samples compiled by adding 10% of the benign traffic records. Using supervised learning, two classifiers were trained to categorize the network traffic. The final performance on the test set was used to demonstrate the effectiveness of the proposed model. Fig. 1 illustrates the flowchart of the proposed intrusion detection model.

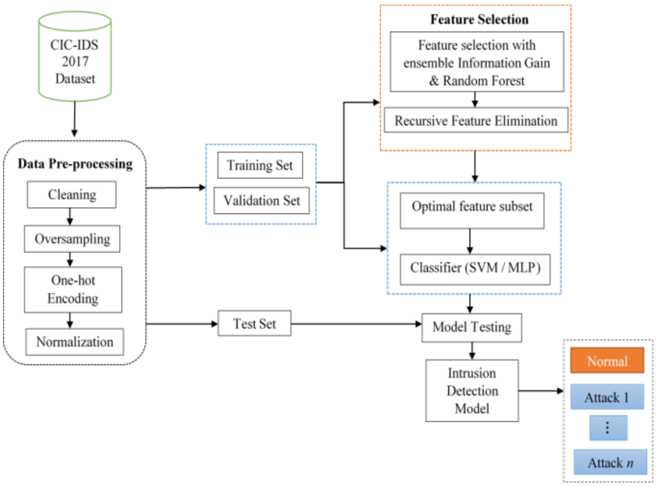

Fig. 1: Schematic Model of the Proposed Intrusion Detection Approach

F. Classification Algorithms for Evaluation

Classification algorithms are used to predict the class of an instance based on the input feature vector. Two machine learning-based classification algorithms namely, Multi-layer Perceptron (MLP) and Support Vector Machine (SVM), were used to construct models that could learn from the labelled datasets and finally make predictions based on of new, unseen data. The features in network traffic data have intricate, non-linear relationships with one another. SVMs and MLPs are equally capable of managing these non-linearities. NIDS frequently work with multifeatured, high-dimensional datasets. SVMs and MLPs work well in these kinds of situations. The goal of NIDS is to spot particular patterns in network traffic that do not match typical usage patterns. MLPs and SVMs perform well on tasks requiring pattern recognition.

i. Multi-layer Perceptron (MLP)

MLP networks are feed-forward artificial neural networks composed of input, output, and several hidden layers [34]. Because of their layered architecture and activation functions, MLPs have the capacity to comprehend intricate feature relationships, which may enable them to detect subtle nuances in attack traffic patterns. The input layer of MLP is associated with the number of features whereas the number of neurons in the output layer is equivalent to the number of classes to be classified. The layers appearing amidst the input and output layers are usually fully connected layers which are trained by backpropagation. In addition, hyperparameter tuning allows MLPs to be adapted to specific NIDS requirements and datasets

ii. Support Vector Machine (SVM)

Support Vector Machine is a statistically based supervised classification algorithm that can be used to efficiently handle high-dimensional data [37]. SVMs mostly concentrate on a subset of data points (support vectors) during training. This conceivably saves a lot of memory for large datasets. The algorithm creates a multi-dimensional hyperplane which separates two classes in the case of binary classification tasks while multi-class classification problems are reduced to multiple binary classification tasks. The goal of SVMs is to maximize the difference between attack classes and regular traffic. For intrusion detection, this emphasis on distinct separation may prove advantageous.

G. Performance Evaluation

To analyze the models five performance parameters are evaluated. A confusion matrix, a technique for summarizing the performance of the classification algorithms, where the outputs are presented in a table layout as shown in Table 3 was used. Model performance metrics including the accuracy, precision, recall, f1-score, and detection rate were calculated using the information given in a confusion matrix as shown in Table 3.

TABLE 3: CONFUSION MATRIX

		Predicted Class	
		Attack class	**Normal (Benign)**
Actual class (Ground truth)	**Attack class**	True Positives (TP)	False Negatives (FN)
	Normal	False Positives (FP)	True Negatives (TN)

The specific metrics used for evaluation are:

Accuracy: Proportion of correctly classified network traffic instances (normal vs. attack).
Precision: Proportion of true positives among identified attacks.
Recall: Proportion of actual attacks correctly identified by the NIDS.
F1-score: Harmonic mean of precision and recall.
Detection Rate: Proportion of attack instances successfully detected by the NIDS.

V. RESULTS AND DISCUSSION

This section presents the findings from the experiment evaluating the proposed ensemble feature selection approach for NIDS. The results present the performance evaluation of the proposed approach for detecting PortScan and DoS attacks using SVM and MLP classifiers and highlight the effectiveness of the ensemble feature selection method in improving the detection performance. The models were implemented in Python.

The experiments began with determining the outcomes for the NIDS using an ensemble feature selection technique, with a particular focus on identifying PortScan and DoS attacks in the CICIDS 2017 dataset. Four experiments were carried out on the different feature selection methods. All the features included in the CICIDS 2017 dataset were used in the first experiment. To distinguish between traffic from PortScan/DoS attacks and regular traffic, the second experiment involved selecting features according to each feature's information gain (IG). The third experiment involved selecting features using the feature importance scores that a trained Random Forest (RF) model estimated, while the final experiment employed the suggested ensemble approach that combines IG, RF, and Recursive Feature Elimination (RFE). Table 4 presents a comparison of the effects of each feature selection technique on the NIDS performance metrics for PortScan and DoS attack detection.

TABLE 4: PERFORMANCE COMPARISON OF FEATURE SELECTION METHODS ON PORTSCAN AND DOS ATTACKS DETECTION

Features Selection Method	Accuracy	Precision	Recall	F-1 Score	Detection Rate
Full Feature Set	0.972	0.968	0.970	0.969	0.965
Information Gain (IG)	0.965	0.960	0.963	0.961	0.958
Random Forest (RF)	0.967	0.962	0.965	0.963	0.960
Ensemble (IG+RF+RFE)	0.975	0.971	0.973	0.972	0.970

The top 10 features were selected. The RF algorithm constructs several decision trees during training where data is iteratively split between each tree according to which feature (in this case, PortScan and DoS) best discriminates between legitimate traffic and attack traffic. Table 5 lists the important features selected. These features are considered the most informative for identifying PortScan and DoS attacks based on the combined analysis using IG, RF, and RFE.

TABLE 5: RANDOM FOREST FEATURE IMPORTANCE SCORES (TOP 10)

Feature Name	Importance Score
Total Length of Flow Packets	0.28
Flow Bytes/s	0.25
Packets Per Flow	0.18
Source Port	0.12
Destination Port	0.10
Protocol	0.08
Flags	0.05
Forward Packets	0.03
Backward Packets	0.02
Total Length of Forward Packets	0.01

Given their high importance scores, "Total Length of Flow Packets" and "Flow Bytes/s" appear to be the most useful features for the RF model in terms of distinguishing between legitimate traffic and PortScan and DoS attacks. These characteristics may be useful in detecting attack patterns as they may record the total amount and size of network traffic packets. Additionally, features like "Source Port," "Destination Port," and "Protocol" are very important. These characteristics probably contribute to the detection of explicit port usage patterns or protocols that are frequently linked to DoS or PortScan attacks.

When comparing the ensemble approach (IG+RF+RFE) to using the entire feature set or individual selection methods (IG or RF alone), the results demonstrates that the ensemble approach achieves the highest accuracy (0.975) and competitive performance across other metrics (precision, recall, F1-measure, detection rate). These results suggest that a smaller feature subset that preserves or even enhances NIDS performance in identifying PortScan and DoS attacks can be successfully identified by the proposed ensemble approach.

Experiments were further conducted to compare SVM and MLP classifiers in detecting PortScan and DoS attacks using the ensemble feature selection approach in the CICIDS 2017 dataset. The Random Forest model's feature importance scores were used to determine which features are most useful for intrusion detection. These features likely provide more discriminative power for the NIDS model. Table 6 presents the results of the experiments and the models detection performance.

TABLE 6: EXPERIMENT RESULTS FOR THE PORTSCAN AND DOS ATTACKS USING FEATURES SELECTED USING ENSEMBLE APPROACH

Classifier	Attack type	Training Time (Sec)	Accuracy	Precision	Recall	F1-Score	Detection Rate
MLP	DOS	32.1	0.965	0.960	0.963	0.962	0.968
	PortScan	38.7	0.977	0.972	0.975	0.974	0.979
SVM	DOS	23.8	0.972	0.968	0.970	0.969	0.975
	PortScan	25.4	0.980	0.975	0.978	0.977	0.982

The finding indicates that when employing the ensemble features for PortScan and DoS attack detection, both SVM and MLP classifiers achieve high accuracy (above 0.96). For both attack kinds, SVM seems to have a slight advantage in accuracy and could train more quickly. The results suggest that the SVM can work especially well with high-dimensional data, such as the potentially big feature set found in NIDS. It is possible that the feature space produced by the ensemble feature selection was still well suited to SVM's capabilities. Again, the margins between classes (normal traffic vs. attack types) are the focus of SVM by default. Therefore, for tasks like intrusion detection, where it's critical to distinguish clearly between normal behaviour and attacks, this feature is helpful. The results further demonstrates that by using an ensemble approach, the number of features needed for the NIDS training are greatly decreased, thereby improving computational efficiency as shown in the training time.

iii. Conclusions and Recommendations

Network Intrusion Detection Systems play a critical role in safeguarding computer networks against cyberattacks. The NIDS are evolving and will likely continue to evolve as network attack methods change and new computational capabilities are achieved. However, high-dimensional network traffic data can hinder their performance. The number of traffic features is huge with irrelevant and redundant features likely to have a great impact on their performance results. Feature selection techniques offer a solution by identifying the most relevant features for intrusion detection. In this paper, an ensemble of Information Gain with Random Forest algorithm was used to select relevant and significant features in the CIC-IDS 2017 dataset and then Recursive Feature Elimination method applied to the reduced features to optimize the selected feature subset.

The study implemented MLP and SVM classifier algorithms in experiments to detect and classify the PortScan and DoS network attacks. Experiment results on these two machine learning algorithms using the selected features, was observed to be less time consuming for the SVM classifier and the model performance improved. The key findings of the study underscore the significance of the proposed ensemble feature selection approach for detecting PortScan and DoS attacks in network traffic. An increasingly secure network environment may result from NIDS's ability to detect PortScan and DoS attacks more effectively by utilising MLP and SVM's strengths and concentrating on informative features. These findings can inform the development of more efficient and accurate NIDS for network security professionals.

Future work on this project could include performing statistical tests to assess the significance of the observed differences between SVM and MLP performance, compare the effectiveness of the suggested ensemble method with other cutting-edge feature selection strategies, and examine how well the chosen features identify new or zero-day variations of PortScan and DoS attacks.

References

[1] M. Ring, S. Wunderlich, D. Scheuring, D. Landes and A. Hotho, “A Survey of Network-based Intrusion Detection Data Sets,” *arXiv preprints,* vol. arXiv:1903.02460v2, p. 17, 6 July 2019.

[2] D. Berman, A. Buczak, J. Chavis and C. Corbett, “A Survey of Deep Learning Methods for Cyber Security,” *Information,* vol. 10, no. 122, pp. 1-35, 2019.

[3] V. Jyothsna and K. Prasad, “Anomaly-Based Intrusion Detection System,” in *Computer and Network Security*, IntechOpen, 2019, pp. 1-15.

[4] R. Vinayakumar, M. Alazab, K. Soman, P. Poornachandran, A. Al-Nemrat and S. Venkatraman, “Deep Learning Approach for Intelligent Intrusion Detection System,” *IEEE Access,* pp. 41525-41550, 2019.

[5] E. Min, J. Long, Q. Liu, J. Cui and W. Chen, “TR-IDS: Anomaly-Based Intrusion Detection through Text-Convolutional Neural Network and Random Forest,” *Security and Communication Networks,* vol. 2018, pp. 1-10, 2018.

[6] K. Patel and B. Buddhadev, “Machine Learning based Research for Network Intrusion Detection: A State-of-the-Art,” *International Journal of Information & Network Security (IJINS),* vol. 3, no. 3, pp. 31-50, June 2014.

[7] P. Angelov, “Anomaly Detection based on Eccentricity Analysis,,” in *IEEE Symposium on Evolving and Autonomous Learning Systems (EALS)*, 2014.

[8] L. Göcs and Z. C. Johanyák, “Identifying Relevant Features of CSE-CIC-IDS2018 Dataset for the Development of an Intrusion Detection System,” *arxiv preprint,* vol. arxiv 2307.11544v1, p. 24, 21 Jul7 2023.

[9] A. Javadpour, S. Abharian and G. Wang, “Feature Selection and Intrusion Detection in Cloud Environment Based on Machine Learning Algorithms,” in *2017 IEEE International Symposium on Parallel and Distributed Processing with Applications and 2017 IEEE International Conference on Ubiquitous Computing and Communications (ISPA/IUCC),* Guangzhou, China, 2017.

[10] K. Ren, Y. Zeng, Z. Cao and Y. Zhang, “ID-RDRL: a deep reinforcement learning-based feature selection intrusion detection model,” *Scientific Reports,* vol. 12, p. 17, 2022.

[11] A. Ali, S. Shaukat, M. Tayyab, M. Khan, J. Khan, Arshad and J. Ahmad, “Network Intrusion Detection Leveraging Machine Learning and Feature Selection,” in *2020 IEEE 17th International Conference on Smart Communities: Improving Quality of Life Using ICT, IoT and AI (HONET),* Charlotte, NC, USA,, 2020.

[12] Y. Swe, P. Aung and A. Hlaing, “A Slow DDoS Attack Detection Mechanism using Feature Weighing and Ranking,” in *Proceedings of the 11th Annual International Conference on Industrial Engineering and Operations Management*, Singapore, 2021.

[13] A. Patil and D. Kshirsagar, “An approach towards hybrid feature selection for detection of DDoS attack,” *An approach towards hybrid feature selection for detection of DDoS attack,* vol. 3, no. 3-4, pp. 274-289, 2021.

[14] D. Kshirsagara and S. Kumar, “Towards an intrusion detection system for detecting web attacks based on an ensemble of filter feature selection techniques,” *CYBER-PHYSICAL SYSTEMS,* vol. 9, no. 3, pp. 244-259, 2023.

[15] Z. Pelletier and M. Abualkibash, “Evaluating the CIC IDS-2017 Dataset Using Machine Learning Methods and Creating Multiple Predictive Models in the Statistical Computing Language R,” *International Research Journal of Advanced Engineering and Science ,* vol. 5, no. 2, pp. 187-191, 2020.

[16] Kurniabudi, D. Stiawan, Darmawijoy, M. Idris, A. Bamhdi and R. Budiarto, “CICIDS-2017 Dataset Feature Analysis With Information

Gain for Anomaly Detection," *IEEE Access,* vol. 8, pp. 132911-132921, 2020.

[17] B. Reis, E. Maia and I. Praça, "Selection and Performance Analysis of CICIDS2017 Features Importance," in *Foundations and Practice of Security: 12th International Symposium, FPS 2019,* Toulouse, France, 2019.

[18] J. Cunningham and B. Yu, "Dimensionality reduction for large-scale neural recordings," *Nat Neurosci,* vol. 17, no. 11, pp. 1500-1509, 2014.

[19] J. Miao and L. Niu, "A Survey on Feature Selection," *Procedia Computer Science,* vol. 91, pp. 919-926, 2016.

[20] Z. Hira and D. Gillies, "A Review of Feature Selection and Feature Extraction Methods Applied on Microarray Data," *Advances in Bioinformatics,* pp. 1-14, 2015.

[21] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Computers and Electrical Engineering,* vol. 40, no. 1, pp. 16-28, 2014.

[22] Y. Zhang, H. Zhang and B. Zhang, "An Effective Ensemble Automatic Feature Selection Method for Network Intrusion Detection," *Information,* vol. 13, no. 7: 314, 2022.

[23] W. Jia, M. Sun, J. Lian and S. Hou, "Feature dimensionality reduction: a review," *Complex & Intelligent Systems*, vol. 8, p. 2663–2693, 21 January 2022.

[24] Y. Yin, J. Jang-Jaccard, W. Xu, A. Singh, J. Zhu, F. Sabrina and J. Kwak, "IGRF-RFE: A hybrid feature selection method for MLP-based network intrusion detection on UNSW-NB15 Dataset.," *Journal of Big Data,* vol. 10, no. 15, p. 26, 2023.

[25] R. Patgiri, U. Varshney, T. Akutota and R. Kunde, "An Investigation on Intrusion Detection System Using Machine Learning," in *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, Bangalore, India, 2018.

[26] P. Dhal and C. Azad, "A comprehensive survey on feature selection in the various fields of machine learning," *Applied Intelligence,* vol. 52, p. 4543–4581, 2022.

[27] G. Biau and E. Scornet, "A random forest guided tour," *TEST,* vol. 25, no. 2, pp. 197-227, 2016.

[28] R. Su, X. Liu and L. Wei, "MinE-RFE: determine the optimal subset from RFE by minimizing the subset-accuracy-defined energy," *Brief Bioinform,* vol. 21, no. 2, pp. 687-698, 2020.

[29] J. Brownlee, "Recursive Feature Elimination (RFE) for Feature Selection in Python," Machine Learning Mastery, 28 August 2020. [Online]. Available: https://machinelearningmastery.com/rfe-feature-selection-in-python/. [Accessed 28 December 2023].

[30] B. Darst, K. Malecki and C. Engelman, "Using recursive feature elimination in random forest to account for correlated variables in high dimensional data.," *BMC Genomic Data,* Vols. 19 (Suppl. 1), 65 , 2018.

[31] O. Mitchell, "Experimental Research Design," *The Encyclopedia of Crime and Punishment,* 2 October 2015.

[32] I. Sharafaldin, A. Gharib, A. H. Lashkari and A. Ghorbani, "Towards a Reliable Intrusion Detection Benchmark Dataset," *Journal of Software Networking,* vol. 2017, no. 1, p. 177–200, 2017.

[33] N. Chawla, K. Bowyer, L. Hall and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research,* vol. 16, pp. 321-357, 2002.

[34] G. E. A. P. A. Batista, A. L. C. Bazzan and M. C. Monard, "Balancing Training Data for Automated Annotation of Keywords: a Case Study," in *WOB*, S. Lifschitz, N. F. A. Jr., G. J. P. Jr. and R. Linden, Eds., 2003, pp. 10-18.

[35] E. Bisong, "Introduction to scikit-learn," in *Building Machine Learning and Deep Learning Models on Google Cloud Platform*, Berlin, Germany, Springer, 2019, p. 215–229.

[36] D. Svozil, V. Kvasnicka and J. Pospichal, "Introduction to multi-layer feed-forward neural networks," *Chemometrics and Intelligent Laboratory Systems,* vol. 39, no. 1, pp. 43-62, 1997.

[37] J. Gu and S. Lu, "An effective intrusion detection approach using SVM with naïve Bayes feature embedding," *Computers & Security*, vol. 103, 2021.