CHAPTER 4

ENSEMBLE BASED COMBINED FILTER FOR FEATURE SELECTION (CFFS) WITH DECISION TREE (DT) CLASSIFIER FOR SINGLE VECTOR DDOS FLOODING ATTACKS DETECTION

4.1 Introduction

In a single vector DDoS flooding assault, a target system, network, or service is bombarded with requests or traffic from several sources. The objective is to exhaust resources of the target such as network connections, CPU, memory, and bandwidth so that it can no longer support authorized users. A network of hacked devices, often known as a "botnet," is managed by the attacker in a classic DDoS flooding assault in order to create and transmit the traffic deluge (Patani, N. and Patel, R., 2017). This traffic can be sent in a number of ways, such as sending a flood of TCP (Transmission Control Protocol) packets to use up all of the target's resources, sending a flood of ICMP (Internet Control Message Protocol) packets, which are frequently used in ping flood attacks, sending a flood of HTTP (Hypertext Transfer Protocol) requests to use up all of the target's web server resources, and sending a flood of UDP (User Datagram Protocol) packets, which are less dependable but require fewer resources to generate much more. A single vector DDoS flooding attack refers to a scenario where the attacker orchestrates the flood of traffic from multiple sources but initiates it as a single coordinated event. This can still have a significant impact on the target, causing service disruption or downtime.

To decrease computational complexity and increase detection accuracy (Venkatesh B and J. Anuradha, 2019) feature selection may be used to discover key characteristics of a dataset in the preprocessing phase prior to classification. The existing defense strategies often have redundant or irrelevant characteristics, which demands a long time to train and classify the attacks and are inefficient for handling large volumes of data. Many applications, including statistical pattern recognition, machine learning, and data mining for reducing data make use of feature selection techniques to enhance performance and identify outliers.

Filter, wrapper, and embedding methods can be used to categorize existing feature selection techniques. In filter approaches, characteristics are categorized in accordance with the inherent information of the data, and they are not classified according to any particular

classification algorithm (Aamir, M. and Zaidi, S.M.A., 2019). Following that, features are assessed and graded in accordance with their underlying characteristics using straightforward metrics like distance, dependence, and information. Comparing these techniques to wrapper approaches, which provide a more accurate output but take longer to perform, these techniques are particularly successful when working with huge datasets. To identify the significance of a feature subset, wrapper and embedding approaches require certain classification algorithms. Combining determination algorithms may enhance classifier performance by detecting traits that are strong but poor on their own collectively, removing extraneous components, and identifying features that, as shown by early discovery, have a good association with the final class (Kasongo, S.M. and Sun, Y., 2019).

As a result, feature selection strategy is employed in this study, such as the CFFS approach, which selects the key characteristics by combining the gain ratio, information gain (IG), ReliefF and chi-squared outputs. The objective of this research is to substantially decrease the feature set while preserving or increasing accuracy in classification using a decision tree classifier. The proposed methodology is assessed using the CICDDOS2019 intrusion detection benchmark dataset with 80 characteristics.

The major contributions of Phase I include,

(i) A Combined filter for feature selection (CFFS) method is proposed with the ML algorithms to identify DDoS attacks

(ii) Designing of an Intelligent IDS to detect Single Vector DDOS flooding attack and evaluated using area-user-curve analysis to avoid the over fitting problems

4.2 Proposed Methodology in Phase I

The importance of the preprocessing stage in feature selection for systems to detect DDoS attacks has been recognized. By highlighting crucial features from the initial dataset, it improves classification accuracy and decreases computing complexity. To provide an optimal selection, it integrates the results of four filtering techniques. Each filter method's output as well as the combined filter are used, with decision tree (C 4.5) classifiers each utilizing their own set of ranking features. The findings are examined and contrasted. Given that it learns more quickly than other classifiers, the DT classifier is often utilized (Kousar, H. et al., 2021) The results demonstrate that, in comparison to other classification procedures, the recommended model has a good classification accuracy and can successfully decrease the number of features.

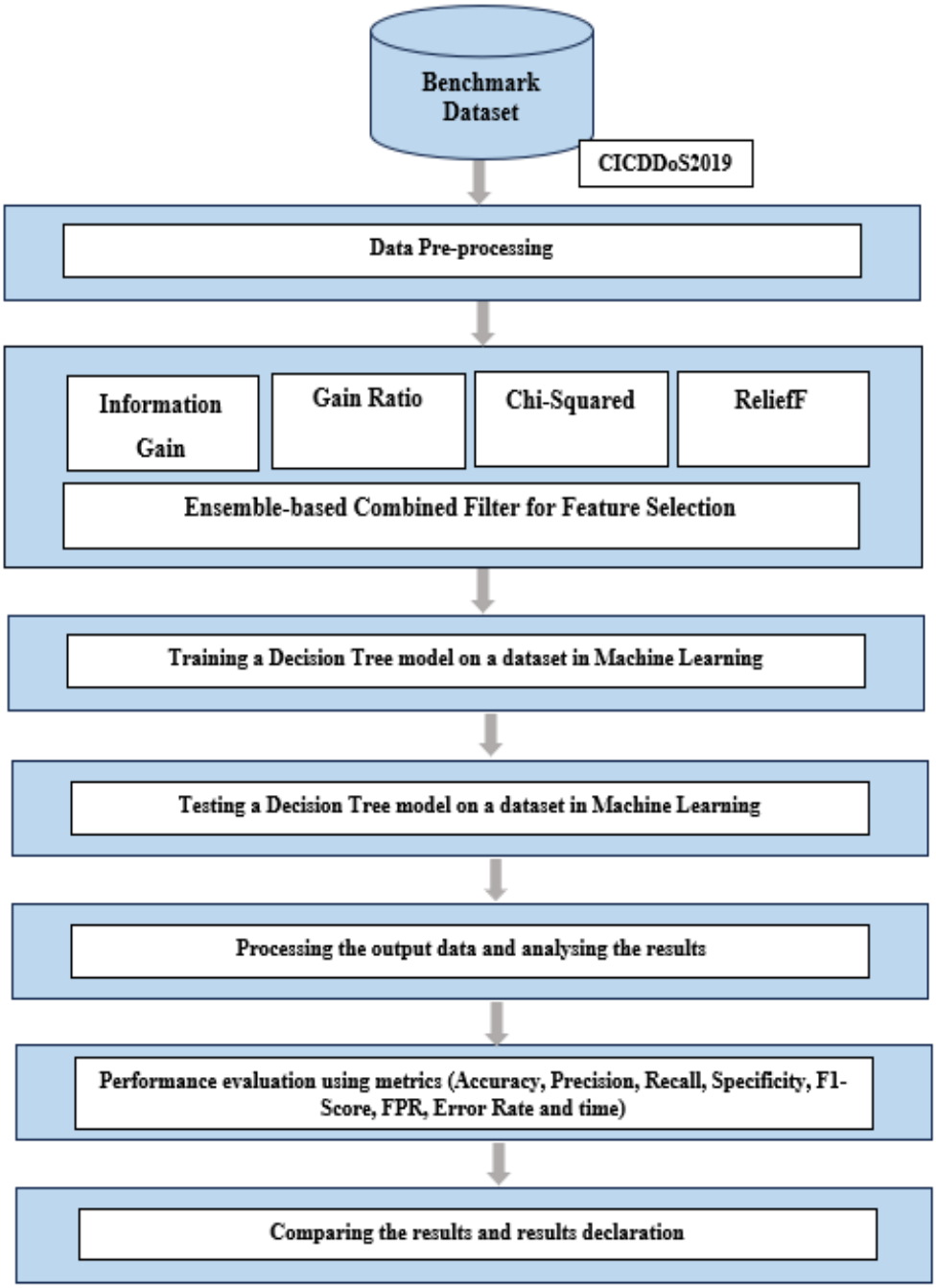

Figure 4.1 Proposed Methodology for Phase I

The outcome from every filter method is combined to get the final chosen feature, which is then decided by the majority vote in the CFFS. To find the most common features amongst the four filtration techniques, a threshold is established and set as 3 (i.e., T = 3). The threshold T = 3 is justified as it balances the inclusion of features strongly supported by filter methods while minimizing irrelevant ones. Requiring a feature to be selected by at least three methods ensures its importance across different criteria, improving the finished

feature set's efficacy and dependability. By reducing overfitting to specific filters, this method produces a more accurate and broad feature selection for categorization. A counter is employed to identify regular characteristics with regard to the specified threshold after integrating all of the chosen feature sets. The final feature set for categorization is made up of features that satisfy the threshold requirements. Figure 4.1 displays the diagrammatic representation of the proposed methodology in Phase I.

4.2.1 Benchmark Dataset

The dataset CICDDoS2019 is widely used by universities, businesses, and researchers as a labeled standard for DDoS attack detection. It includes benign and current DDoS attacks, with traffic analysis results from CICFlowMeter-V3. The dataset contains 172,647 records (58.6%) of attack traffic and 121,980 records (41.4%) of normal traffic. The recently released dataset CICDoS2019 is used to assess the suggested model. The collection includes a variety of DDoS flooding assaults that may be executed over TCP/UDP network and application layer protocols. The dataset was gathered for training and testing assessment on two distinct days. On "March 11th, 2019 the testing set was developed which comprises 7 DDoS attacks, whereas the training set was captured on January 12th, 2019, and contains 12 distinct types of DDoS attacks. The CICDoS2019 dataset is accessible in both flow and PCAP file formats". The initial features in the CICDDOS2019 dataset includes a wide array of network traffic attributes, 78 features are included in the overall number of features utilized (Khalid, S., Khalil, T. and Nasreen, 2014).

4.2.2 Data Pre-Processing

The process of transforming raw data into information that may be consumed is called preprocessing. The information is cleansed to eliminate any subpar information, eliminate any erroneous information, and eliminate superfluous details. The first stage converts raw data into a format suitable for analysis by preprocessing the source datasets (Bouke, Mohamed Aly, et al., 2023) (Osanaiye, O et al., 2016).

There are 78 characteristics in the CICDDOS2019 dataset. There are anomalies in this data that may be substituted with different values. For example, the least attribute values can be used to substitute missing values, and the highest attr ibute values can be applied to find infinite values. For instance, the outliers 'Infinity' and 'NaN' are present in

'FlowBytes', while only 'Infinity' is present in 'Flowpackets' (Alghazzawi, D., et al., 2021). A dataset attribute is referred to be a zero-adjectives type when both its lowest and highest values are zero. Ten zero-adjectives with the same value for every entry were found in the CICDDOS2019 dataset after analysis.

Therefore, eliminating the zero- adjectives will increase the accuracy of the model (Batchu, Raj Kumar, and Hari Seetha., 2022). Last but not least, CICDDOS2019 has nominal qualities. Since it is impossible to train ML models with small amount of data prior to any procedure, data labels are substituted with numerical values instead of nominal features. The framework is trained to classify input traffic as two categories ("Benign" and "attack") using the benign "0" and attack "1" labels, which encode the nominal amount of data.

4.2.3 Ensemble-based Combined Filter for Feature Selection

Regardless of the classification technique, the filter based feature selection approach is an essential step for locating important characteristics in a dataset (Yu, L. and Liu, H., 2003). These techniques use a ranking mechanism to rate features according to their relevance and perform statistical analyses straight to the original training dataset. The selection of the most relevant traits is ensured by excluding features that fall below a certain threshold. Filter techniques are frequently used in real-world applications because of their effectiveness and simplicity. The suggested strategy finds 16 common traits that are essential for improving classification performance by using the combined advantages of various techniques.

The selection of the Combined Filter Feature Selection (CFFS) approach, integrated with the Decision Tree (DT) classifier, is guided by the need to balance detection accuracy, model interpretability, and computational efficiency. CFFS leverages multiple statistical measures, Information Gain, Gain Ratio, Chi-Square, and ReliefF to capture diverse perspectives on feature importance, leading to more consistent and generalizable feature subsets. This combination is particularly effective in single-vector DDoS flooding scenarios, where rapid identification of core features can significantly reduce processing time without compromising performance. The DT classifier further complements this by offering transparent decision paths and fast training, which are critical for practical intrusion detection systems. While more complex classifiers such as SVM or deep learning models could be considered, they typically involve higher computational overhead and longer

training times. The CFFS-DT pairing, in contrast, achieves high classification accuracy with reduced dimensionality and low latency, making it a suitable choice for environments where real-time detection and resource constraints are important considerations.

4.2.3.1 Information Gain (IG)

To extract relevant qualities from a set of features, IG is a feature selection method that lowers the uncertainty involved in class attribute determination while the actual value associated with the feature is unknown (Venkatesh, B., and J. Anuradha 2019). The variability is caused by the distribution's entropy, the sample's entropy, and the dataset's anticipated entropy. It is possible to define the X (entropy of variable) as:

$$H(X) = -\sum_{i} P(x_i) \log_2(P(x_i)) \tag{4.1}$$

Let $p(x_i)$ gives the previous probability for X's value. Following the observation, the entropy of X is defined as given:

$$H(X/Y) = -\sum_{j} P\left(y_j\right) \sum_{i} P\left(x_i|y_j\right) \log_2\left(P\left(x_i|y_j\right)\right) \tag{4.2}$$

$p(x_i|y_j)$ in the equation above is the subsequent probability of X given Y. The quantity that the entropy of X falls to reflect new insight about X that was offered by Y is known as the information gain, and it is calculated as follows:

$$IG(X/Y) = H(X) - H(X|Y) \tag{4.3}$$

4.2.3.2 Gain Ratio

IG bias is reduced by using gain ratios to highlight characteristics with high variance values, which can dominate the selection process despite not being inherently significant. A high gain ratio value indicates that information is evenly distributed, while a low value shows that information is concentrated in one part of the attribute (Baig, Z.A., Sait, S.M. and Shaheen, A., 2013). The entropy distribution of a feature's occurrence can be employed for determining its intrinsic information (L Devi, P Subathra and P Kumar, 2015). The gain ratio among a given feature (x) and its value (y) can be calculated using the following formulas.

$$\text{Gain Ratio } (y,\ x) = \frac{\text{Information Gain}(y,\ x)}{\text{Intrinsic Value}(x)},$$

where (4.4)

$$\text{Intrinsic Value } (x) = -\sum \frac{|S_i|}{|S|} * \text{Log}_2 \frac{|S_i|}{S}$$

the |S| shows count of possible feature x values and |Si| shows how many real values feature x could take.

4.2.3.3 Chi-Square

The chi-square (χ2) statistic evaluates the degree of independence between these two variables by calculating a score. The symbol χ2 (chi-square) specifically denotes the squared differences normalized by expected frequencies, which is central to evaluating statistical relationships. Chi-square is characterized by:

$$\chi^2(r,\ c_i) = \frac{N[P(r,\ c_i)P(\bar{r},\bar{c}_i) - P(r,\bar{c}_i)P(\bar{r},\ c_i)]^2}{P(r)P(\bar{r})P(c_i)P(\bar{c}_i)} \quad (4.5)$$

Where "N → entire dataset, r → the presence of the feature ($\bar{r}$ its absence) and c_i refers to the class. $P(r, c_i)$ is the probability that feature r occurs in class c_i, and $P(\bar{r}, c_i)$ is the probability that the feature r does not occur in class c_i. Moreover, $P(r, \bar{c}_i)$ and $P(\bar{r}, \bar{c}_i)$ are the probabilities that the features do or do not occur in a class that is not labelled c_i and so on.

4.2.3.4 ReliefF

The ReliefF feature choosing approach makes use of ongoing testing to identify the closest hit and the closest miss. Each characteristic is given weight by the attribute evaluator in accordance with how well it can discriminate between the various classes. The weight of features that are greater than a user-defined threshold is picked as relevant features. To address its flaws, ReliefF was created as an evolution of the original Relief algorithm (Moradkhani M et al., 2015. ReliefF differs from other filter techniques in that it has a low bias and may be used in all circumstances. The element determination approach's suggested consolidated channel makes use of the 33% split of positioning highlights from the aforementioned channel operations.

Prior to training, there is a preprocessing step called CFFS where unique filter approaches are utilized for the initial selecting process. The first dataset's components are positioned by applying the IG, gain-proportion, chi-squared and ReliefF channel techniques before selecting a 33% split of the placement highlights. These characteristics are considered to be the most relevant ones for each channel process. The output of each channel technique is combined to produce the CFFS yield, which selects the final component using a simple greater part vote. To find the attributes that recur often, a limit is chosen from the four filtering techniques. Once all the capabilities have been aggregated, a counter is employed to separate the typical characteristics from the edge. The final feature set for classification is chosen from among those features that satisfy the threshold requirements.

4.2.3.5 Pseudocode for proposed combined filter for feature selection method

The algorithms listed below are used to develop the CFFS technique. Table 4.1 provides the pseudocode for the suggested Combined filter feature selection approach.

Table 4.1 Pseudocode for the suggested Combined filter Feature Selection Approach

Algorithm 1 (Filter feature ranking methods)

Step 1a: Let X i = {X 1, X 2, X 3 … … … … , X 78 } be the feature set in the CICDDOS dataset and C i represents the class (i.e. normal or anomaly).

Step 2a: Sort and rank the features X i for every filter based on its importance in determining the output class C i .

Step 3a: Choose one-third split of the output from every filter selection approach.

Algorithm 2 (Merge output features)

Step 1b: Merge specified output features from every filtering approach.

Step 2b: Compute the threshold T for feature counts.

Step 3b: Determine the feature occurrence rate for each filter technique.

Algorithm 3 (Ensemble selection)

Step 1c: Select the common features intercepts from Algorithm.2

Step 2c: If the feature count is below the threshold, drop the feature orelse select the feature.

Step 3c: For every feature in the one-third split subgroup, repeat step 2.

4.3 Steps involved in Proposed Approach

The proposed method introduces a unique three-step ensemble feature selection process that integrates multiple filter feature ranking methods, combines their outputs, and applies an ensemble selection strategy. This approach helps in guaranteeing that the chosen features are important according to filter methods and stable across the methods.

4.3.1 Pseudocode for Decision Tree and SVM Classifier

The pseudocode given in Table 4.2 and Table 4.3 depicts the working of a Decision Tree (C 4.5) Classifier and SVM classifier, which are two of the popular classification ML algorithm. It was seen that DT can extract meaningful features based on its feature-based approach (Al-Omari et al, 2021) and SVM (Gu, J. and Lu, S., 2021) is very good in finding the boundary between classes which makes it very effective in dealing with the significant characteristics of DDoS attacks.

Table 4.2 Pseudocode for the Decision Tree Classifier

Step 1. Function preprocess_data(data):

- Some of the general preprocessing steps include handling of missing values, Ouliers and scaling numerical features.

- Divide the data into two sets: training and testing.

Step 2. Function train_J48(training_data):

- Train a J48 decision tree classifier using the training_data.

- Set parameters such as tree depth, minimum number of instances per leaf, and others based on domain knowledge and experimentation.

- Return the trained decision tree model.

Step 3. Function test_J48(model, testing_data):

- Apply the trained J48 model to the testing_data to make predictions.

- Analyze the performance of the model using various metrics.

- Return the evaluation results.

Step 4. Function detect_DDoS_attacks(new_data, model):

- Apply the trained J48 model to new_data to classify instances as normal or DDoS attacks.

- Return the predictions.

Step 5. Main program:

- Load the dataset containing features related to network traffic.

- Preprocess the dataset using the preprocess_data function.

- Train the J48 model using the train_J48 function.

- Test the efficiency of the model using the test_J48 function.

- Use detect_DDoS_attacks function to detect DDoS attacks

Table 4.3 Pseudocode for the SVM Classifier

Step 1. Initialize hyperparameters:

- Kernel type (RBF)

- Regularization parameter (C)

- Kernel coefficient (Gamma=0.05)

- Degree of the polynomial kernel (3)

Step 2. Load the training dataset:

- Separate features (X) and labels (Y)

Step 3. Preprocess the data:

- Normalize or scale features

Step 4. Initialize the SVM model with chosen hyperparameters:

- For example, in scikit-learn: svm_model = SVC(kernel='linear', C=1.0)

Step 5. Train the SVM model:

- Call the fit() function with the training data:

svm_model.fit(X_train, y_train)

Step 6. Evaluate the model:

- Use metrics such as accuracy, precision, recall, F1-score, FPR, Error Rate.

- ROC Curve is utilized.

Step 7. Prediction:

- Given data, call the predict() function:

predicted_labels = svm_model.predict(X_new)

Step 8. Use the predicted labels for decision making or further analysis.

In the context of the CICDDOS2019 database, which is commonly applied for intrusion detection system (IDS) research, feature extraction is particularly important because of the complex traffic data. The most significant fourteen features from each filter method were found by selecting the top one-third of the rated features shown in Table 4.4a. After applying the combined filter method to the outputs of each filter method, features 22, 53, 60, 64, 45, 12, 27, 51, 50, 42, 36, 33, 15, and 46 showed up across over three filter

techniques. This indicates the significance of those features in distinguishing the output class shown in Table 4.4b. Table 4.4b displays the 14 selected features from the combined filter method's top-ranked features of the CICDDoS2019 dataset.

Table 4.4a. Features selected using Filter methods

Filter Method	Selected Features (Top 14)
Information Gain (IG)	1, 3, 8, 12, 15, 22, 27, 33, 45, 51, 53, 60, 64, 70
Gain Ratio	1, 4, 10, 18, 22, 25, 30, 42, 50, 55, 57, 62, 68, 72
Chi-squared	2, 5, 9, 14, 20, 22, 29, 34, 40, 49, 52, 58, 64, 75
ReliefF	1, 7, 13, 19, 22, 28, 36, 41, 46, 53, 60, 65, 71, 76

Table 4.4b. Combined Filter for Feature Selection (CFFS)

Filter Method	Selected Features (Top 14)
CFFS	22, 53, 60, 64, 45, 12, 27, 51, 50, 42, 36, 33, 15, 46

4.4 Experimental Results and Discussion

CICDDOS2019 dataset is applied to assess the performance of CFFS method with Decision tree (C 4.5) classifier and SVM dataset to assess the effectiveness of the CFFS approach utilizing 10-fold cross-validation using Decision tree (C 4.5) classification and SVM Classifier. Data is split into ten equal-sized sets for the cross-validation before ten training and validation rounds are carried out. From the 10 folds, one fold is utilized for validation in each iteration, with the remaining being used for learning. A 64-bit Windows 8.1 computer of Intel Core i5-4210U CPU and 6 GB of RAM is used for all tests. To classify data as either normal or attack, the suggested CFFS approach chooses the most crucial characteristics for the Decision tree (C 4.5) and SVM classification algorithm. Weka (http://www.cs.waikato.ac.nz/ml/weka/) , a program that includes a number of ML algorithms for data mining tasks, was used to do the research. The experiments' categorization settings are set to Weka's default values. The proposed CFFS method selects the most important features for the Decision tree (C 4.5) and SVM classification algorithm, which classifies data as either an attack or normal. In the experiments, the parametric terms for classification are set to the default values in Weka shown in Table 4.5. The list of Single-vector DDoS Flooding attacks considered in this research is (UDP Flood, ICMP Flood, DNS Flood, SYN Flood, ACK Flood).

Table 4.5. Parameters setup in Weka

Parameter	Default Value
Confidence Factor (C)	0.25
Minimum Number of Instances per Leaf (M)	2
Use Unpruned Tree	FALSE
Number of Folds for Pruning	3
Seed for Randomization	1
Binary Splits	FALSE
Debug Mode	FALSE
Save Instance Data	FALSE
Use Laplace Smoothing	FALSE
Collapse Tree	TRUE

These measurements compare the method to other machine learning-based filtering strategies. The outcomes, which are shown in Tables 4.6–4.12 and illustrated in comparison bar graphs (Figures 4.2–4.8), highlight the efficacy of the approach. When applied to the CICDDoS2019 dataset, the suggested method outperforms other feature selection techniques with noteworthy performance measures, including high accuracy (97.69%), precision (99%), recall (97%), and F-score (0.983), coupled with a low FPR (0.73).

Classifier performance relies on metrics derived from TP, TN, FP, and FN. TP corresponds to correctly identified attacks, TN to correctly classified normal samples, FP to false alarms, and FN to attacks misclassified as normal. Furthermore, the amount of time needed to construct classification models indicates how well learning processes work after the use of each feature selection technique. Performance indicators are comprehensively illustrated across the figures and tables mentioned.

The classification models implemented are,

i) Information Gain – Decision Tree (IG-DT)

ii) Gain Ratio – Decision Tree (GR-DT)

iii) Chi-Square – Decision Tree (CT-DT)

iv) ReliefF – Decision Tree (RF-DT)

v) Combined Filter for Feature Section – Decision Tree (CFFS-DT)

vi) Information Gain – Support Vector Machine (IG-SVM)

vii) Gain Ratio – Support Vector Machine (GR-DT)

viii) Chi-Square – Support Vector Machine (CT-DT)

ix) ReliefF – Support Vector Machine (RF-DT)

x) Combined Filter for Feature Section – Support Vector Machine (CFFS-SVM)

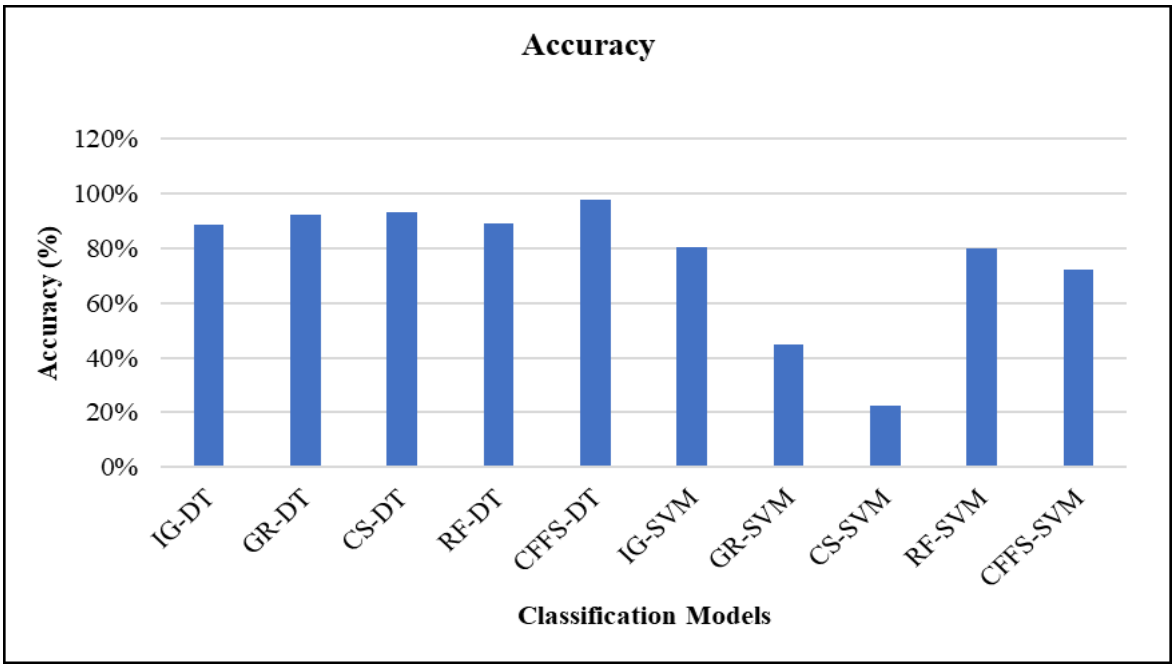

Figure 4.2 Accuracy rates

Table 4.6 Accuracy rate values

Approaches	Accuracy (%)
IG-DT	88.64
GR-DT	92.23
CS-DT	93.26
RF-DT	89.16
CFFS-DT	**97.69**
IG-SVM	80.63
GR-SVM	44.88
CS-SVM	22.47
RF-SVM	79.86
CFFS-SVM	72.16

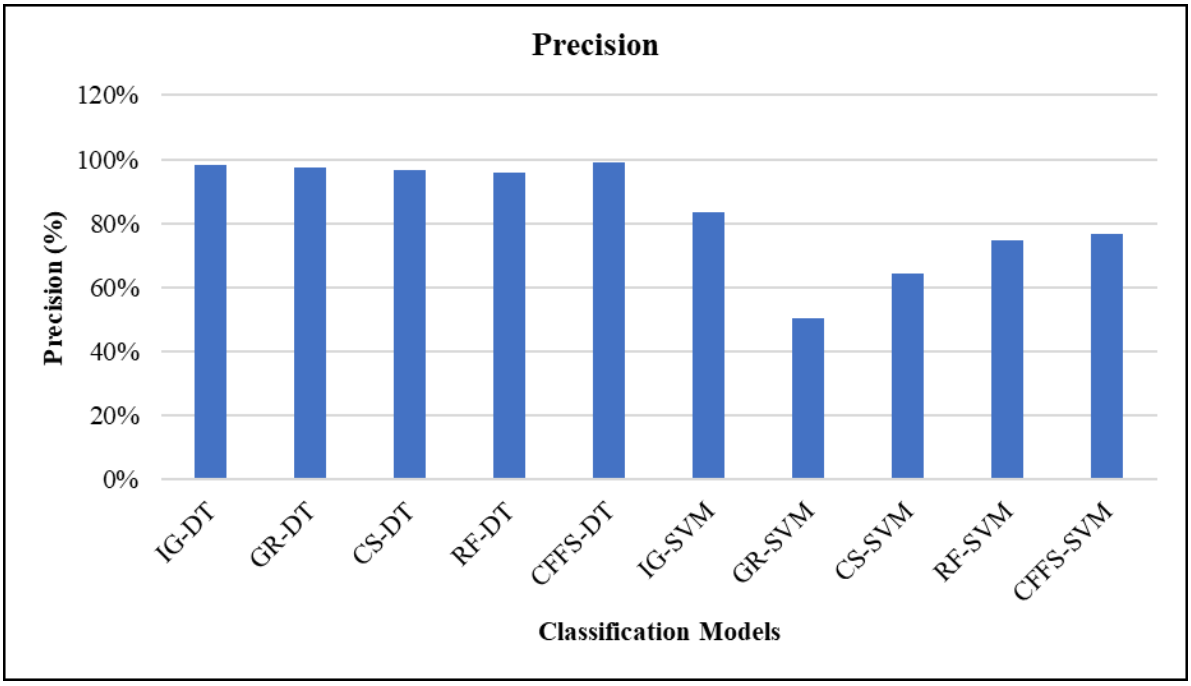

Figure 4.3 Precision rates

Table 4.7 Precision rate values

Approaches	Precision (%)
IG-DT	98.25
GR-DT	97.51
CS-DT	96.62
RF-DT	96.01
CFFS-DT	**99.13**
IG-SVM	83.50
GR-SVM	50.46
CS-SVM	64.52
RF-SVM	74.70
CFFS-SVM	76.59

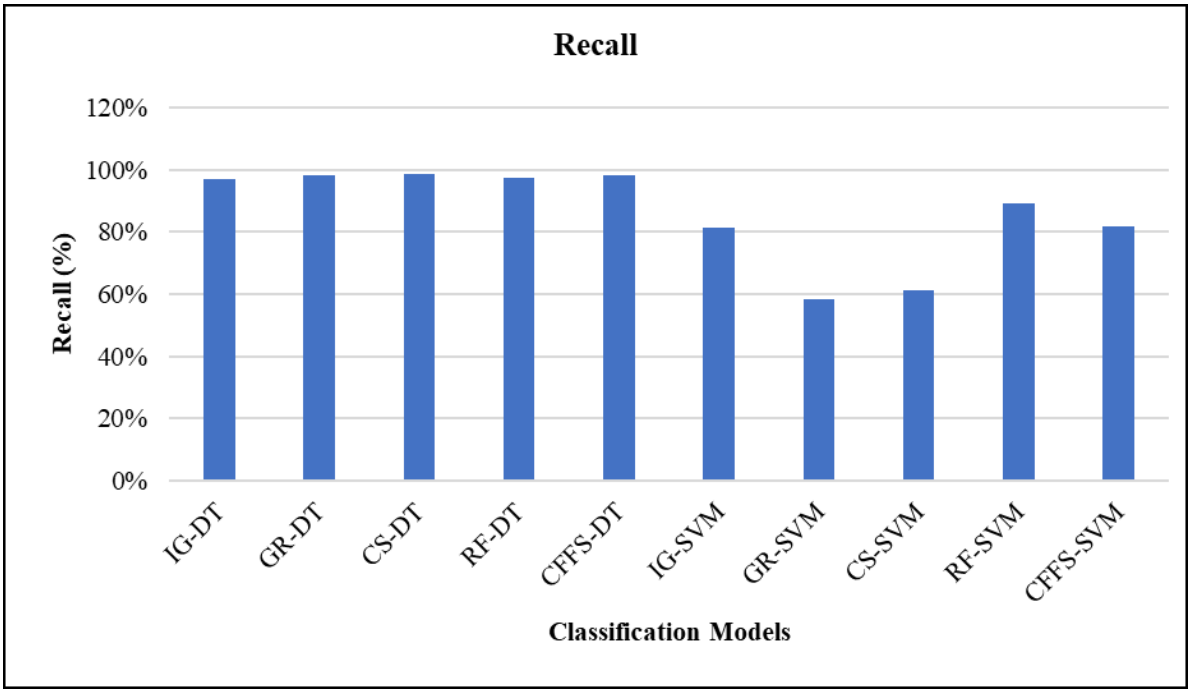

Figure 4.4 Recall Rates

Table 4.8 Recall Rate Values

Approaches	Recall (%)
IG-DT	99
GR-DT	98.50
CS-DT	99.70
RF-DT	97.50
CFFS-DT	**98.50**
IG-SVM	81.25
GR-SVM	58.20
CS-SVM	61.20
RF-SVM	89.25
CFFS-SVM	81.75

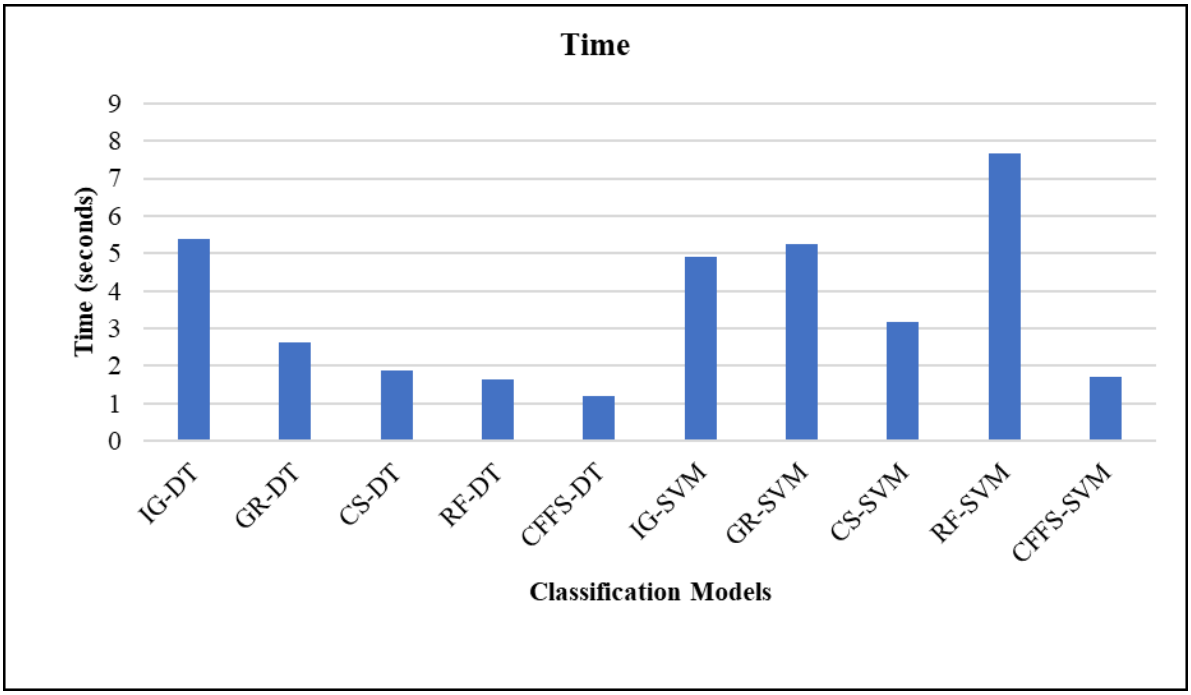

Figure 4.5 Time of proposed Methods

Table 4.9 Time Values

Approaches	Time (s)
IG-DT	5.37
GR-DT	2.64
CS-DT	1.87
RF-DT	1.64
CFFS-DT	**1.2**
IG-SVM	4.92
GR-SVM	5.24
CS-SVM	3.18
RF-SVM	7.66
CFFS-SVM	1.7

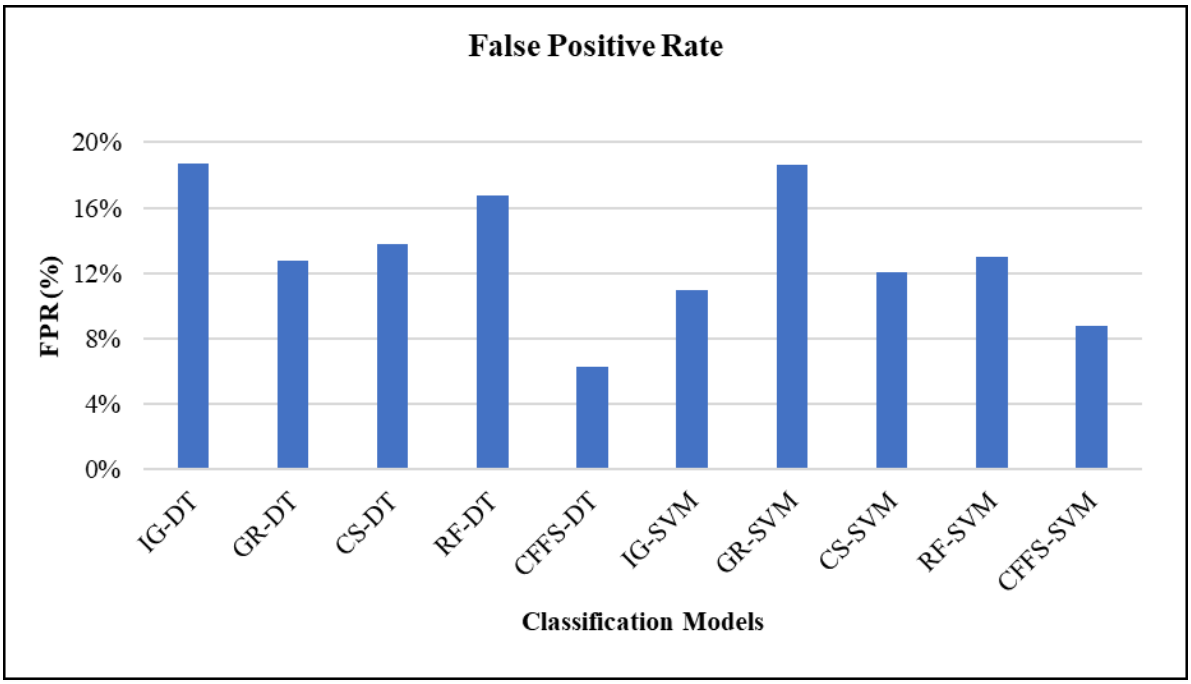

Figure 4.6 False Positive Rates

Table 4.10 False Positive Rate values

Approaches	FPR (%)
IG-DT	18.71
GR-DT	12.80
CS-DT	13.79
RF-DT	16.74
CFFS-DT	**6.32**
IG-SVM	10.94
GR-SVM	18.62
CS-SVM	12.04
RF-SVM	13.00
CFFS-SVM	8.81

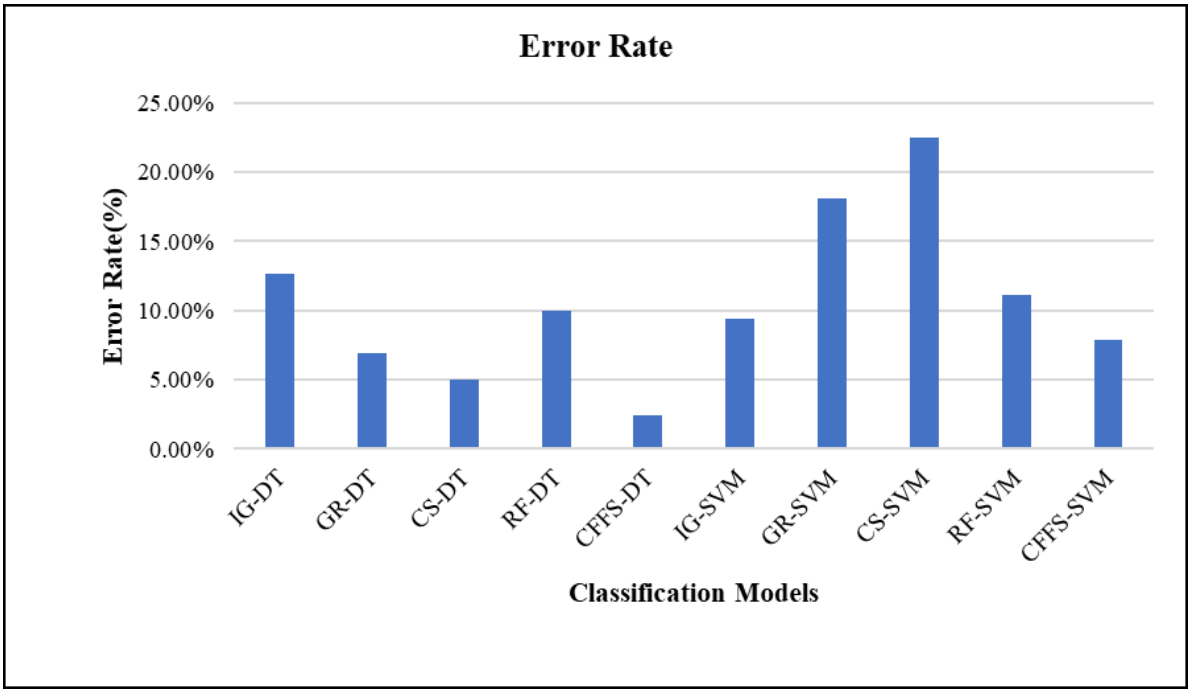

Figure 4.7 Error rate

Table 4.11 Error rate values

Approaches	Error Rate (%)
IG-DT	12.61
GR-DT	6.96
CS-DT	5.01
RF-DT	10.00
CFFS-DT	**2.40**
IG-SVM	9.36
GR-SVM	18.12
CS-SVM	22.52
RF-SVM	11.14
CFFS-SVM	7.84

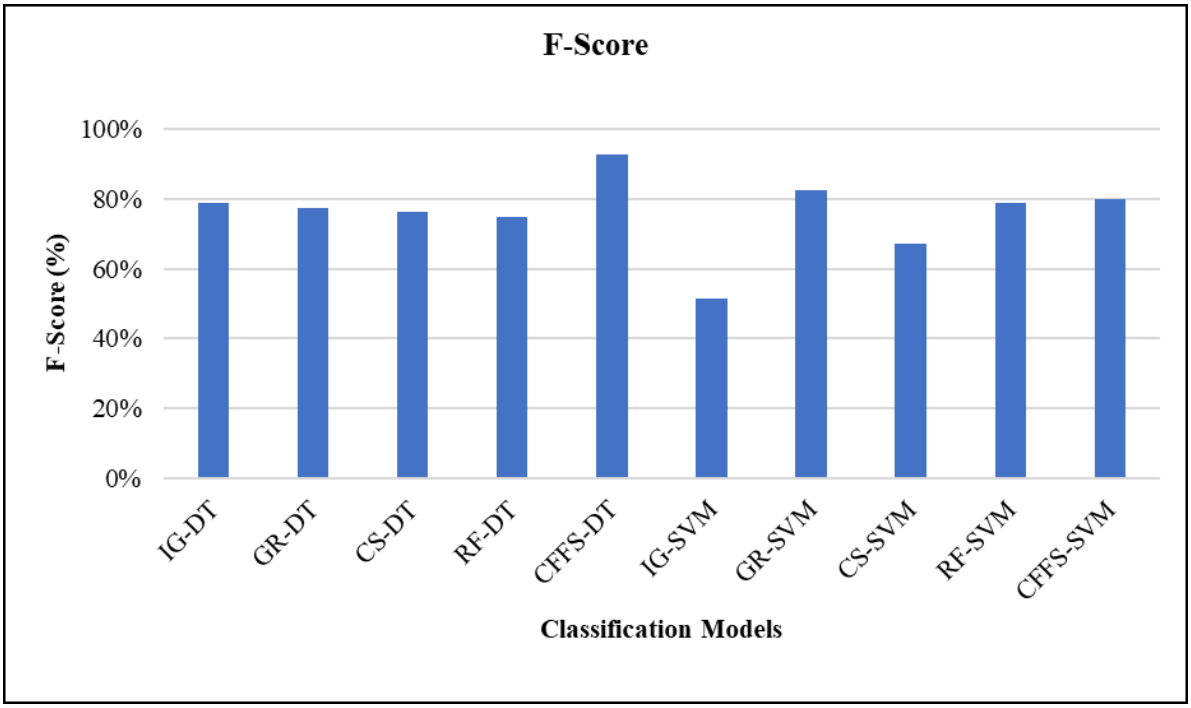

Figure 4.8 F-Score rates

Table 4.12 F-Score rate Values

Approaches	F-Score (%)
IG-DT	78.76
GR-DT	77.45
CS-DT	76.16
RF-DT	74.80
CFFS-DT	**92.90**
IG-SVM	51.34
GR-SVM	82.43
CS-SVM	67.33
RF-SVM	78.67
CFFS-SVM	79.82

Single Vector DDoS flooding attacks further demonstrate the viability of machine learning in link with ensemble-based feature selection as a way to recognize these dangers. The performance metrics of the SVM and Decision Tree (C 4.5) classifiers employing different feature selection methods are displayed in Table 4.13. alongside the proposed CFFS model. The classification models implemented include Decision Tree (C 4.5), SVM, and others used in comparison.

Table 4.13 Performance Measures

Approaches	Accuracy (%)	Precision (%)	Recall (%)	FPR (%)	F-Score (%)	Time (S)	Error Rate (%)
IG-DT	88.64	98.25	97	18.71	78.76	5.37	12.61
GR-DT	92.23	97.51	98.50	12.80	77.45	2.64	6.96
CS-DT	93.26	96.62	98.70	13.79	76.16	1.87	5.01
RF-DT	89.16	96.01	97.50	16.74	74.80	1.64	10.00
CFFS-DT	**97.69**	**99.13**	**98.50**	**6.32**	**92.90**	**1.2**	**2.40**
IG-SVM	80.63	83.50	81.25	10.94	51.34	4.92	9.36
GR-SVM	44.88	50.46	58.20	18.62	82.43	5.24	18.12
CS-SVM	22.47	64.52	61.20	12.04	67.33	3.18	22.52
RF-SVM	79.86	74.70	89.25	13.00	78.67	7.66	11.14
CFFS-SVM	72.16	76.59	81.75	8.81	79.82	1.7	7.84

Among the various approaches evaluated, CFFS-DT stands out for its exemplary performance. Using the same dataset, it has 97.69% accuracy, precision, and recall qualities of 99.13% and 98.50% respectively. Interestingly, it presents good features in differentiation between true and false positive measures; its FPR is astonishingly low at only 6.32%. In addition, the F-score of 92.90% demonstrates the exceptional accuracy and recall of the suggested model. CFFS-DT has a short execution time of 1.2 seconds; in addition to the enhanced performance, it presents.

The ROC curve (Meng, Y., 2012) shows the relationship across two variables, true and false classifications, and is used to evaluate the model's performance. The Area under the ROC Curve (AUC) measures the difference between false positive and true positive rates.

The suggested model's AUC value of 0.976, which is shown in Figure 4.9, shows that it successfully distinguishes between 97.69% of positive and negative classes.

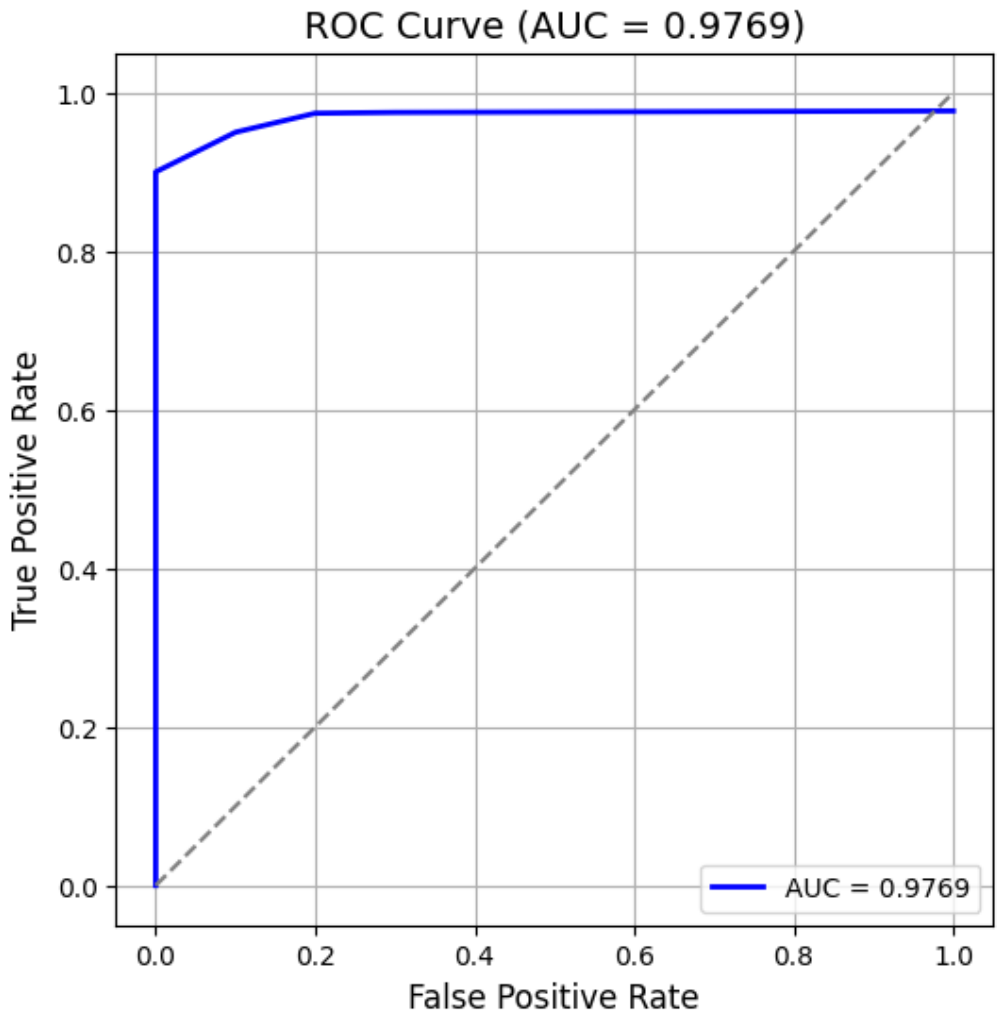

Figure 4.9 ROC Curve of proposed methods in Phase I

The values that the proposed framework accurately predicts are represented by the TP and TN. In difference, the misclassification is indicated by FP and FN (Ho et al., 2012).

When evaluating the proposed approach against Multi Vector DDoS Flooding attacks, its performance decreases by 37.7%. Therefore, it becomes essential to introduce a more intelligent approach to counteract Multi vector Flooding attacks. As depicted in Figure 4.10 below, the accuracy of detection noticeably declines, indicating ample room for enhancing performance.

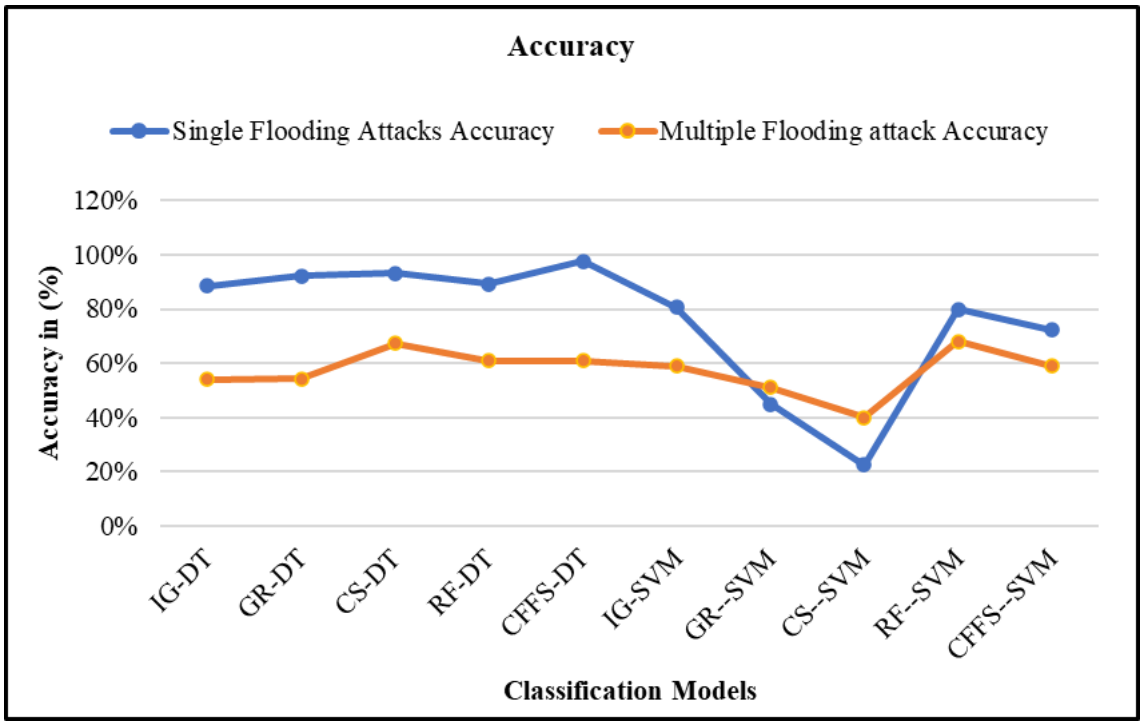

Figure 4.10 Performance of the Proposed Approach with Single and Multi vector Flooding attacks detection

4.5 Chapter Summary

The Combined Filter for Feature Selection (CFFS) method integrated with the decision tree C 4.5 (DT) classification demonstrates clear efficacy in identifying Single Vector DDoS Flooding attacks, as indicated by its performance metrics. Integrating filters at this stage enhances the feature selection process significantly, proving to be a crucial element in the successful detection of Single Vector DDoS Flooding attacks during Phase I. This methodology is also complexity aware, adapting to the intricate data nature to ensure improved detection while maintaining efficiency. Therefore, by using the advantage of the simple and simple to understand DT Due to SVM's robust performance, a thorough examination is conducted under various assault scenarios. In particular, CFFS-DT produces a lower false positive rate of 6.32% with an accuracy of 97.69% with precision and recall over 99%. This is supported by the F-score showing that the measure had a precision and recall in equal measure pegged at 92.90%.

Also, it has a short execution time of 1.2 seconds to support its efficiency and better performance. However, while handling Multi Vector Flooding attacks, performance degradation indicates the need for better strategies to counter such threats. This decline in the

detection accuracy was observed and the search for better tactics to improve performance was seen, thus making a case for more adaptive approaches to counter ever changing threats. These outcomes stress the significance of combining ML and feature selection techniques algorithms in enhancing Multiple DDoS attack defense capabilities of IDSs and opening the path to more robust and flexible cybersecurity solutions in response to new and emerging cyber threats. The next chapter provides a Strategic Framework for addressing the detection of Multiple DDoS attacks described above.