CHAPTER 5

FEATURE ENGINEERING TECHNIQUES WITH HYBRIDIZATION OF IMPROVED DRAGONFLY OPTIMIZATION ALGORITHM (IDOA) AND DECISION TREE CLASSIFICATION (DT) FOR MULTI-VECTOR DDOS FLOODING ATTACKS DETECTION

5.1 Introduction

As a result of the limitations outlined in the first contribution, contribution two presents a Strategic-Framework with computational intelligence for both feature selection and classification using ML The suggested framework will decrease computing latency and improve the multi-vector DDoS Flooding attack's detection accuracy.

Multi-vector DDoS Flooding attacks are an organized attack that employs a number of flooding methods at the same time ICMP, HTTP, SYN, and UDP floods are among the techniques used to overload the target's server, network, or application resources (Brahma, K.K., et al. 2019). These assaults are designed to consume bandwidth, processing capacity and memory to the extent that the availability of the services is significantly impacted or indeed, the services are unavailable. The complexity and intensity of multi-vector DDoS Flooding attacks make them particularly challenging to detect and mitigate, as they exploit different layers of the network stack and often resemble legitimate traffic patterns. Implementing multi-layered protection measures such as traffic filtering, rate limitation, anomaly detection systems, and utilizing DDoS protection mechanisms which may absorb and neutralize attack traffic are examples of effective mitigation solutions. Addressing this threat is essential for safeguarding against financial loss, reputational damage, and ensuring the continuity of critical online operations (Bhardwaj et al., 2021).

Because of their effectiveness in exploring vast, complicated search spaces and locating optimum or nearly ideal solutions, CI optimization methods are crucial in feature selection. CI approaches may be used for a variety of feature selection problems and performance metrics, and they are especially well-suited for large numbers of input variables since they prevent entrapment in local optima. Besides, they also provide parallelism and scalability which greatly enhance the selection process. High relevance, low dimensionality and better orographic by selecting features that significantly contribute

towards the generation of the dataset. It also improves model interpretability and its dynamic stops irrelevant data from swaying the outcome and noise from affecting the model.

This chapter provides a high-level framework designed to improve feature selection approaches effectively, especially for the detection of DDoS flooding assaults with many vectors. This chapter's primary objectives include:

(i) Introducing a unified machine learning framework for general DDoS detection, with a special emphasis on design improvements for better detection.

(ii) Dynamic strategy and procedure to tackle the new threat models where the machine learning models capabilities are found optimal.

(iii) Applying Computational Intelligence methods together with machine learning to identify multi-vector DDoS flooding assaults effectively, enhancing detection performance using the provided framework.

5.2 Proposed Framework and Approach

The new framework proposes a strategic-level approach that aims at improving the feature selection methods for better identification of multi-vector DDoS flooding attacks. Through integrating the IDOA with the DT, this framework is expected to enhance the feature selection methodology which will improve the choice of the most suitable traits for categorization. Before supplying data to the machine learning model, preprocessing is crucial. This includes scaling, depreciating outliers, and eliminating missing values. IDOA is used for feature engineering and selection. It designs two new features that enhance model performance and chooses the best features to enhance model performance. The Decision Tree Algorithm is then employed for training and testing the feature set, analyze the output data and assess the outcome. Validation guarantees that when further risks enter the system, the light feature selection will continue to work. Several parameters are utilized to access the performance of the model and comparisons are made with other models to make the final conclusion about the effectiveness of the presented method.

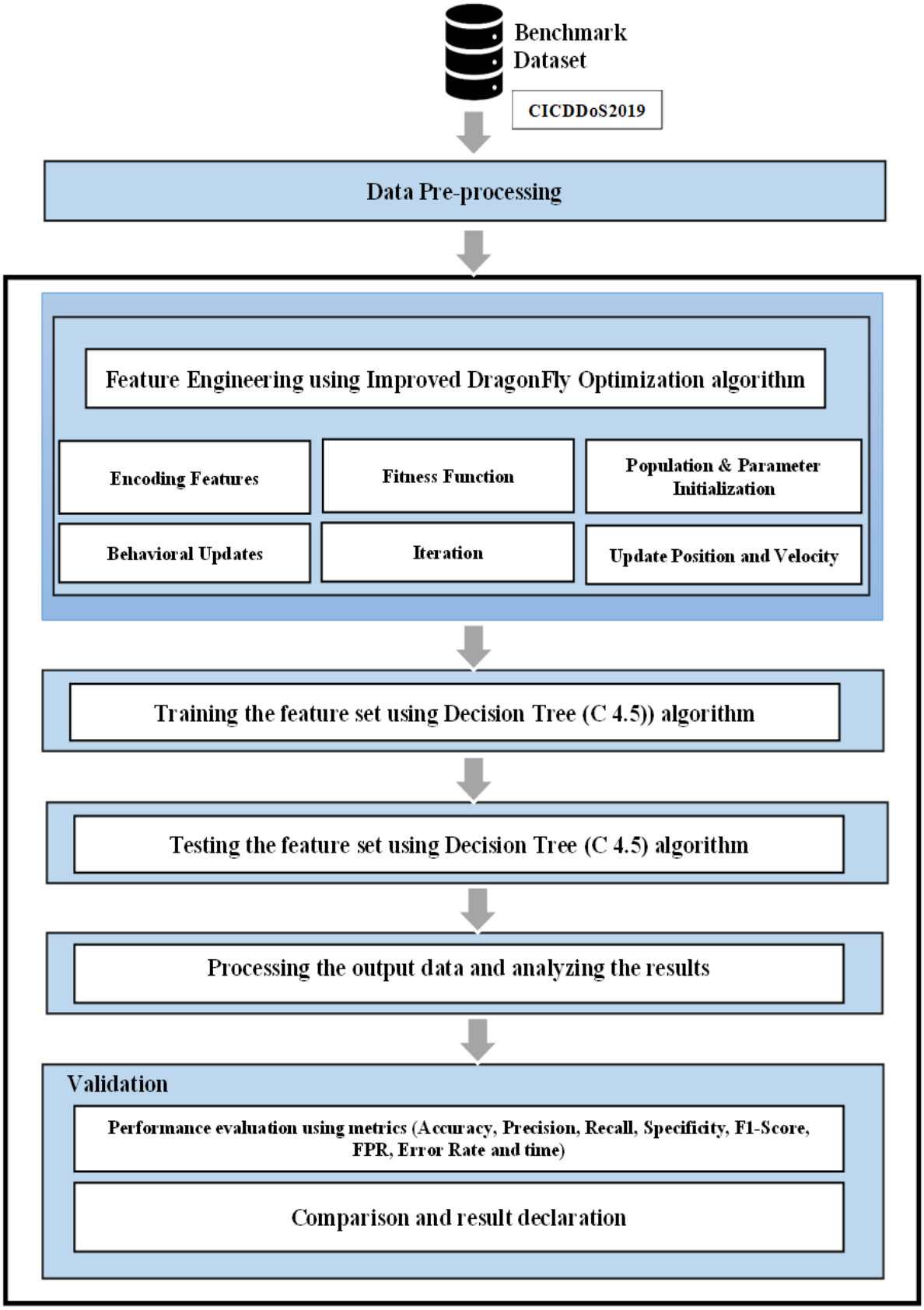

Figure 5.1 Proposed Framework for Phase II

This integrated machine learning approach focuses on the design improvements that strengthen the flexibility and stability of DDoS detection systems and includes optimization techniques and validation processes that correspond to the shift in attack patterns. Using Computational Intelligence in combination with machine learning, the proposed framework provides a robust solution for identifying important features, and improving the effectiveness of systems that detect DDoS attacks. The goal of the suggested strategic framework is to improve the created algorithm's classification performance and efficacy for detecting multi-vector DDoS flooding assaults. Below, in Figure 5.1, is the suggested framework.

The proposed strategy guarantees a systematic way of treating features based on the characteristics of given data. This approach can be applied universally to any type of cyber intrusion including DDoS attacks and addresses inherent data issues including skewness, collinearity and multicollinearity for subsequent machine learning phases. In the feature engineering module, one of the most important things to address is the missing values. Different methods like imputation, where data is replaced by average, maximum or minimum value is used. If the percentage of the missing values is high compared to the assigned values, then such a feature may have to be excluded. Furthermore, the proposed framework also considers the Dragonfly Optimization Algorithm (Sayed, G.I., Tharwat, A. and Hassanien, A.E., 2019) one among the most advanced feature selection approach. When used to create feature selection datasets with fewer features, this actually helps to enhance the feature selection process.

5.2.1 Data Set

The framework used the well-known and globally utilized CICDDoS2019 dataset. These consist includes common network traffic and different kinds of DDoS assaults; and the data are analyzed using CICFlowMeter-V3 to extract the attack types, ports, IP addresses of source and destination and timestamps recorded in CSV files format. It has about 121,980 records of normal traffic and the other 172,647 records' traffic data is of attack traffic; the attack scenarios include DDoS with different types; and this dataset will be updated frequently. It include TCP/UDP network and application layer protocols captured for testing and training over two consecutive days. As we also observed in CICDDoS2019 dataset, as depicted in Figure 5.2 there were total of 41.4% (121,980 records) of normal traffic while 58.6% (172,647 records) of traffic was categorized as attack traffic.

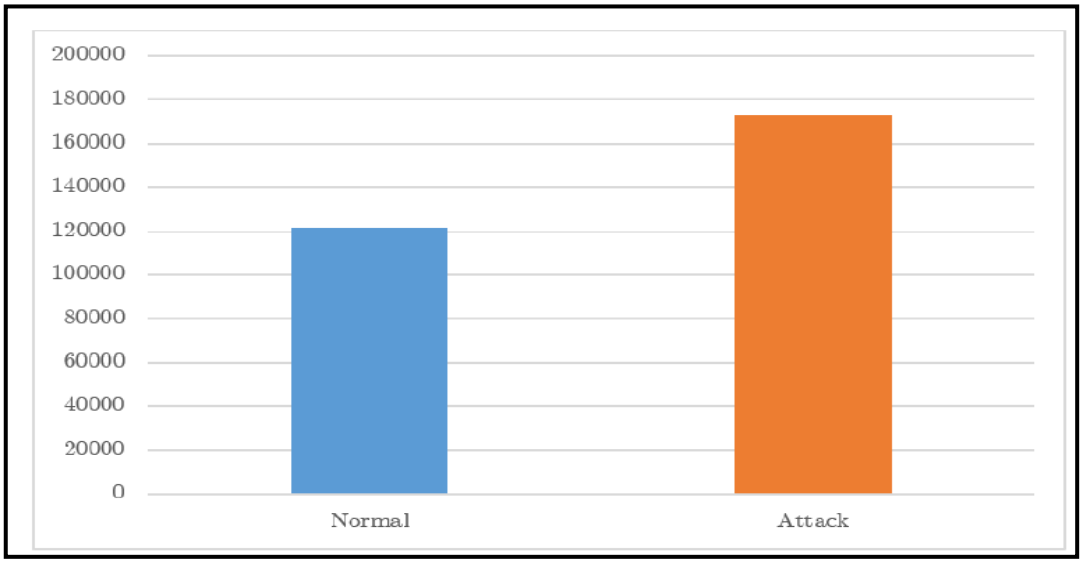

Figure 5.2 A bar graph illustrating the traffic type distribution in the CICDDoS2019 dataset

5.2.2 Enhanced Feature Engineering and Extraction using Improved Dragonfly Optimization Algorithm (IDOA)

Feature engineering is the process that involves generating new features or modifying already existing ones so as to improve the performance of ML models especially when used with datasets such as CICDDoS2019. The models are better able to identify patterns suggestive of malicious activity by converting unprocessed network traffic data into more relevant and meaningful information. Normalization, encoding, and statistical transformation are examples of feature engineering approaches (Aamir, M. and Zaidi, S.M.A., 2019) (Tesfahun, A. and Bhaskari, D.L., 2013)). Furthermore, model accuracy may be greatly increased by creating additional features that capture intricate connections in the data. To enhance the intrusion detection model's detection capabilities for the CICDDoS2019 dataset, feature engineering may assist in finding crucial characteristics that distinguish between malicious and normal traffic.

Feature creation is the process of defining new features while reducing or expanding existing once to enhance the ability of ML. It has great significance in IDS construction especially when used on dataset such as CICDDoS2019. By analyzing a number of network traffic features derived from raw packet data, the models are able to improve identification and discretization of features in order to identify patterns reflective of malicious activities. Those feature engineering techniques include normalization, encoding, and statistical

transformation on the features. Moreover, building new features that reflect others' intricate relationships aspect spikes the accuracy.

Within this phase, two features were built utilizing feature engineering and selection by feature IDFOA. The first feature, the A network flow's forward and backward traffic are compared using the ratio of total forward packet length to total backward packet length. Finding the flow of the columns Total FPL and Total BPL) yields this characteristic. The second feature, the Product of Flow length and Total Forward Packets, attempts to take into account the relationship between the flow length and the forward-pointing packing of the provided data. This functionality was developed using the "Flow" table analysis, where the column attributes include the Flow Duration (FD) and Total Forward Packets (TFP) and their product is calculated. These newly created features, therefore, are posed to yield more information regarding traffic behaviour hence enhancing the models capability of detecting DDoS attacks effectively.

(i) The Ratio Of The Forward And Backward Packets Total Lengths

This feature is defined as the total forwarded packet length and total reverse packet length, therefore forward to backward packet length ratio. The steps to construct this feature are as follows:

Step 1. Identify relevant columns:

- Total Forward Packet Length (let's denote it as FPL)
- Total Backward Packet Length (let's denote it as BPL)

Step 2. Compute the ratio:

- Create a new feature, Ratio_FP_to_BP, is calculated as:

$$Ratio\_FP\_to\_BP = \frac{FPL}{BPL}$$

(ii) Flow Duration multiplied by Total Forward Packets

This feature is calculated by the increasing flow duration by the total count of packets in moving forward. The steps to construct this feature are as follows:

Step 1. Identify relevant columns:

- Flow Duration (let's denote it as FD)
- Total Forward Packets (let's denote it as TFP)

Step 2. Compute the product:

- Create a new feature, Product_FD_TFP, which is calculated as: $Product\_FD\_TFP = FD * TFP$

Both dynamic and static swarming behaviors of dragonflies in the wild served as the inspiration for the Dragonfly Optimization method. It demonstrates how the individuals are moved throughout the search region in relation to the adversary, food supply, and one another (Sayed, G.I., Tharwat, A. and Hassanien, A.E., 2019) (Rahman C.M et al., 2023).

Table 5.1 Pseudocode for the Dragonfly Optimization Algorithm

Step 1. Initialize parameters: population_size, max_iterations, initial_step_size, initial_attraction_coefficient, initial_randomness_coefficient, learning_rate_decay_factor, inertia_weight.

Step 2. Initialize population of dragonflies randomly.

Step 3. Define fitness function to evaluate subsets of features.

Step 4. Initialize adaptive learning rates: step_size, attraction_coefficient, randomness_coefficient.

Sep 5. For each iteration from 1 to max_iterations:

- For each dragonfly:

- For each other dragonfly:

- If higher fitness, move towards it with adaptive attraction coefficient.

- Else, move randomly with adaptive randomness coefficient

- Update position based on movement rules with adaptive step size.

- Evaluate fitness; if higher, update position and fitness.

- Sort dragonflies by fitness.

- Select best-performing dragonflies for next iteration.

-Adjust adaptive learning rates: decay step_size, attraction_coefficient, randomness_coefficient.

Step 6. Select dragonfly with highest fitness as the optimal feature subset.

Step 7. Return the optimal feature subset.

In the pseudocode shown in Table 5.1, the Dragonfly Optimization Algorithm iteratively optimizes feature subsets by simulating the movement of dragonflies within a search space. Dragonflies adjust their positions based on the attractiveness of neighboring

solutions and their own exploration behavior. The algorithm converges towards depending on the performance of a classifier trained with the chosen features, the ideal feature subset that maximizes a predetermined fitness function.

The swarming behavior depends on three fundamental principles: -

- Separation: It denotes of avoiding static collision between individuals.
- Alignment: It denotes the velocity matching the individual to the neighborhood individual.
- Cohesion: It denotes the tendency of individuals to the neighboring mass center.

Since survival is the primary objective, all individuals are disturbed by external enemies and drawn to food sources. These parameters are statistically defined (Mirjalili S., 2016).

Separation: This is calculated using below equation 5.1:

$$S_i = -\sum_{k=1}^{M} Y - Y_k, \tag{5.1}$$

Alignment: *This describes the average of the velocities of all neighbors using below equation 5.2:*

$$A_i = \frac{\sum_{k=1}^{M} V_k}{M}, \tag{5.2}$$

Cohesion: this is calculated using the below equation 5.3:

$$C_i = \frac{\sum_{k=1}^{M} Y_k}{M} - Y \tag{5.3}$$

Attraction towards a food source is determined by using equation 5.4 below:

$$F_i = Y^+ - Y \tag{5.4}$$

Distraction An adversary is estimated externally using equation 5.5 below:

$$E_i = Y^- - Y \tag{5.5}$$

The behavior of the dragonfly is a combination of these five shown in Equation 5.6.

$$\Delta Y_{t+1} = (aA_i + sS_i + cC_i + eE_i + fF_i) + w\Delta Y_t \tag{5.6}$$

When A denotes alignment weight, C represents cohesion weight, e gives enemy weight, food weight is denoted by f, separation weight by s, and inertia weight by w. t stands for multiplication counter. There are 2000 iterations. There are forty dragonflies. d = the overall size of vector positions of charging flight step.

5.2.2.1 Improved Dragonfly Optimization Algorithm

The IDOA extends the learning rates to make movements and step sizes flexible. Through the acquisition of a feedback mechanism that depends on the current optimization performance, learning rates are thus adjusted. This seem to balance exploration and exploitation better and thus converge much faster while at the same time selecting good features. The modified equation for incorporating adaptive learning rates is:

$$\Delta Y_{t+1} = (\alpha_t aA_i + \sigma_t sS_i + \gamma_t cC_{i+} \epsilon_t eE_{i+} \phi_t fF_i) + \omega_t w\Delta Y_t$$

Where:

- α_t is the adaptive learning rate for alignment.
- σ_t is the adaptive learning rate for separation.
- γ_t is the adaptive learning rate for cohesion.
- ϵ_t is the adaptive learning rate for attraction towards food.
- ϕ_t is the adaptive learning rate for distraction from enemy.
- ω_t is the adaptive inertia weight for the previous step size.

This enables the algorithm to alter its behavior for acceleration of convergence as well as optimization of the features selection.

The feature selection technique depends on the IDOA and feature engineering is also applied to the CICDDoS2019 dataset. The first data preprocessing step was to address categorical variables, missing values, and standardizing numerical data for comparison value ranges. Two novel features are engineered: the forward to backward packet size ratio, and the flow duration times total number of forward packets which capture some vital interactivity in the traffic flow. Moreover, if required, the IDOA algorithm is used to choose a collection of characteristics that would be most helpful for the correctness of the model. To minimize classification errors, the fitness function leading the IDOA in the feature space was developed. The following are the chosen features:

(i) Flow Duration

(ii) Total Fwd Packets

(iii) Fwd Packet Length Mean

(iv) Flow Duration x Total Fwd Packets

(v) Fwd/Bwd Packet Length Ratio

(vi) Total Length of Fwd Packets

(vii) Bwd Packet Length Mean

(viii) Packet Length Mean

(ix) Max Packet Length

Explanations of Features:

(i) **Flow Duration:** How long the flow lasted expressed in microseconds, an inherent measure of session and potential DDoS attack length.

(ii) **Total Fwd Packets:** The sum of all packets sent forward determines the quantity of traffic transmitted.

(iii) **Fwd Packet Length Mean:** An estimate of the overall forward packet size to get the general idea of the typical packet size.

(iv) **Flow Duration x Total Fwd Packets:** The newly developed metric that shows the relationship between flow duration and the volume of forward packets indicating when the transmission sustains long bursts of data.

a. **Fwd/Bwd Packet Length Ratio:** Another new feature to represent the traffic, both forward and backward, which shows that the asymmetric flows occur in DDoS attack traffic are included.

(v) **Total Length of Fwd Packets:** The overall data volume, which is the sum of the lengths of all packets pushed forward.

(vi) **Bwd Packet Length Mean:** The mean length of packets transferred backwards, which serves as a benchmark for forwarding traffic.

(vii) **Packet Length Mean:** A generic traffic characteristic that is the mean length of all the packets in the flow.

(viii) **Max Packet Length:** The longest one packet in the flow may be, identifying anomalously large packets which could signal attack traffic.

These features, refined and ranked through the DFO and feature engineering process, in the long term, it will significantly enrich the dataset in terms of distinguishing and predicting DDoS attacks, which will improve the current level of cybersecurity protection.

The selection of the Dragonfly Optimization Algorithm (DOA) combined with the Decision Tree (DT) classifier was driven by the need to balance detection performance with computational efficiency, particularly for complex multi-vector DDoS flooding attacks. DOA is well-suited for high-dimensional search spaces and enables effective feature subset selection by simulating natural swarming behavior, which helps avoid local optima and enhances convergence. The improved version (IDOA) further refines this process through adaptive learning rates, resulting in faster optimization and better feature relevance. On the classification side, the Decision Tree model offers quick execution, interpretability, and low computational cost, making it an ideal fit for scenarios. While more complex models could potentially offer marginal accuracy gains, the chosen IDOA-DT hybrid achieves high accuracy with significantly lower training time and complexity, making it a robust and practical solution for DDoS detection using the CICDDoS2019 dataset.

5.3 Decision Tree Classifiers for Training and Testing

These features have been also used in identifying DDoS attacks in the CICDDoS2019 dataset to improve performance of a Decision Tree classifier. Amongst those, Flow

Duration, Total Forward Packets, and Forward Packet Length Mean are particularly relevant to understand the properties of transmission processes. Furthermore, the variables like, Flow duration x total forward packets, the Fwd/Bwd packet length ratio, are capturing the interaction terms as well as one-sided traffic that are signs of attacks or abnormal traffic which enhances the classifiers capability to separate normal from possible attack traffic.

This model is utilized by the Decision Tree classifier through the construction of decision nodes that carry out a division of the data set under consideration according to several features. For instance, a node might see 'traffic' as a DDoS potential traffic if the Flow Duration is high and the Total Forward Packets are equally high. This process of recursive splitting continues until the tree can classify all the data correctly. Information such as the mean and maximum of the packet sizes assists the classifier in detecting unusual packets sizes, particularly in DDoD attacks. Through these comprehensive and selected features, the Decision Tree can construct a comprehensive model that played an important role in identifying DDoS assaults and strengthens the endeavor to strengthen network security.

5.4 Validation

A key part of the proposed strategic framework is developing Intelligent IDS and using analysis to verify results and prevent misleading accuracy. Often, training data mirrors test data too closely, leading to overly optimistic performance results, a problem known as overfitting. To address this, k-fold cross-validation is commonly used. It divides the dataset into k parts, using k–1 for training and 1 for testing in each round, rotating until all folds have been used for testing. This method helps ensure the model's reliability and prevents overfitting. The average error across all k rounds, called the cross-validation error, is used as a performance metric. In this case, 10-fold cross-validation was used to evaluate the classifier, with the MISE forecast error calculated as per equation (5.7) (Behal & Kumar, 2016).

$$\widehat{MISE}_{CV} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i^{-k(i)})^2 = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{f}(x_i, \alpha^{-k(i)}))^2 \quad (5.7)$$

For each round accuracy score is measured and the test split is modified. Ensures that the boundaries applied don't result in over foundation in the case that the precision adjustment stays in a particular value range.

5.4.1 Performance Evaluation

The last phase of the proposed framework is the test stage, where different exactness focuses and readings from other chosen measures may be compared to ascertain the optimum way to set the proposed model for the greatest results. By organizing the data according to just one categorization in all information circumstances, testing can avoid the exactness chaos that leads to a false sense of model precision. It offers no benefits of both positive and negative deceptive reading, yet it can still demonstrate high levels of accuracy. The recipient's presentation factor diagram (ROC) is set up between the positive and negative readings, with the lower bend (AUC) providing real precision to the model characterization, to avoid this uncertainty.

5.5 Experimental Results and Discussion

This section analyzes the experimental findings from using the IDOA in combination with other machine learning methods to identify multi-vector DDoS flooding assaults. The experiments were carried out using CICDDoS2019 dataset. The objective is to assess the degree to which the proposed framework improves feature selection and, therefore, increases the likelihood of accurate detection. These are then quantitatively analyzed in detail to give out performance information of the hybrid approach and its possible impacts on cybersecurity applications. Table 5.2 shows the parameters of an Improved Dragonfly Optimization which is an optimization tool derived from dragonflies' inherent static and dynamic characteristics. The alteration of these parameters like the population size (N), separation weight (s) and the inertia weight (w) is important in regulating the dragonflies’ movement in the search space, exploitation/exploration properties as well as the convergence to solutions bounded by limits LB and UB.

Table 5.2 Dragonfly Optimization Algorithm Parameter Configurations

Parameter	Value
Population Size	30
Max Iterations	100
Initial Step Size	0.1
Initial Attraction Coefficient (a)	0.5
Initial Separation Coefficient (s)	0.5
Initial Cohesion Coefficient (c)	0.5
Initial Attraction Coefficient (e)	0.5
Initial Distraction Coefficient (f)	0.5
Learning Rate Decay Factor	0.99
Initial Inertia Weight (w)	0.5

Various combinations of the suggested framework with Improved Dragonfly optimization for detecting multi-vector ML methods are used to test and evaluate the detection of DDoS attacks. An accuracy rate comparison bar graph for the proposed architecture trained on the CICDDoS2019 dataset (Khalid, S., Khalil, T. and Nasreen, S., 2014). The gathered outcomes indicate that the proposed framework works well when combined with the Decision Tree classification model and the Improved Dragonfly Optimization method.

The classification models evaluated in this chapter are,

i) Improved Dragonfly Optimization - Naive Bayes (IDOA-NB)

ii) Improved Dragonfly Optimization - Booster (IDOA-Booster)

iii) Improved Dragonfly Optimization - Random Forest (IDOA-RF)

iv) Improved Dragonfly Optimization - Support Vector Machine (IDOA-SVM)

v) Improved Dragonfly Optimization - Logistic Regression (IDOA-LR)

vi) Improved Dragonfly Optimization - Decision Tree (IDOA-DT)

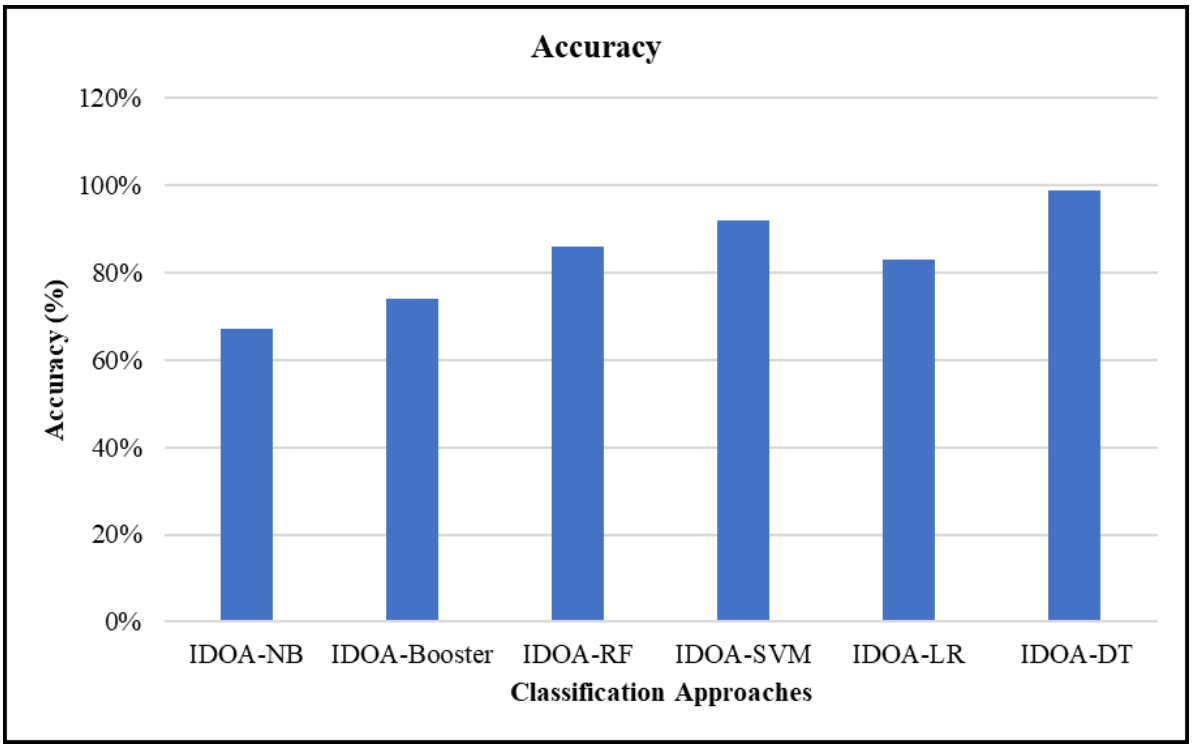

Figure 5.3 Accuracy of the proposed Strategic level framework

Table 5.3 Accuray values of the proposed Strategic level framework

Approaches	Accuracy (%)
IDOA-NB	67.00
IDOA-Booster	74.00
IDOA-RF	86.00
IDOA-SVM	92.00
IDOA-LR	83.00
IDOA-DT	**98.89**

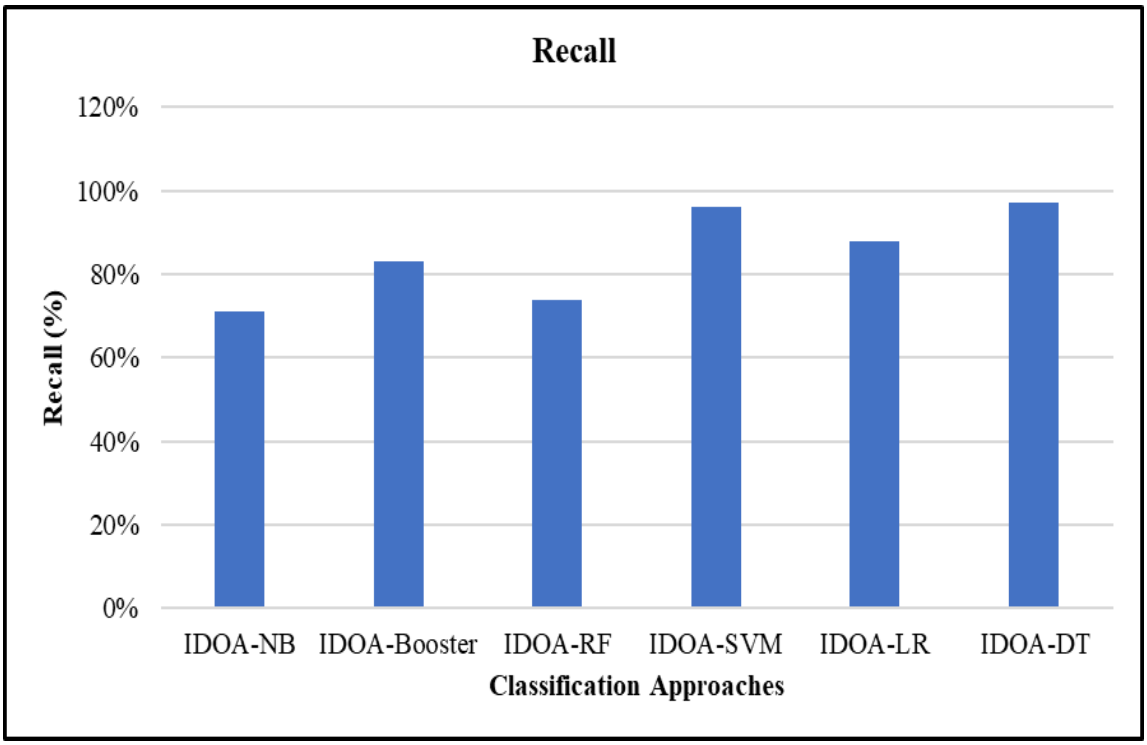

Figure 5.4 Recall of the proposed Strategic level framework

Table 5.4 Recall values of the proposed Strategic level framework

Approach	Recall (%)
IDOA-NB	71.00
IDOA-Booster	83.00
IDOA-RF	74.00
IDOA-SVM	96.00
IDOA-LR	88.00
IDOA-DT	**97.00**

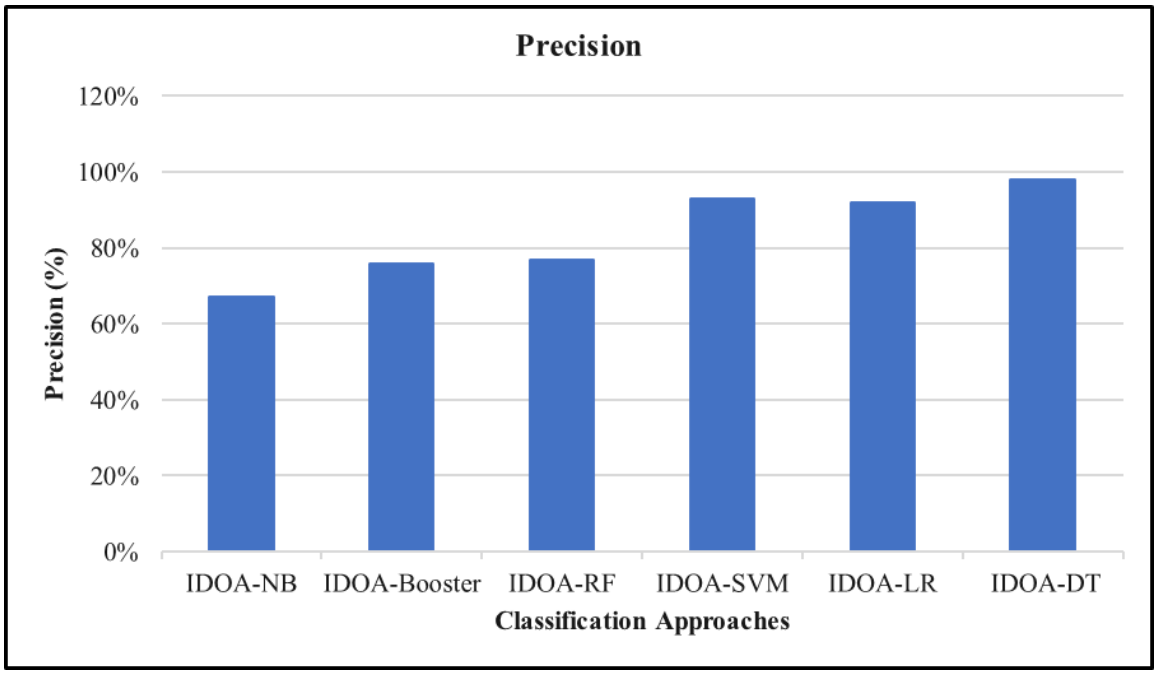

Figure 5.5 Precision of the proposed Strategic level framework

Table 5.5 Precision values of the proposed Strategic level framework

Approach	Precision (%)
IDOA-NB	67.00
IDOA-Booster	76.00
IDOA-RF	77.00
IDOA-SVM	93.00
IDOA-LR	92.00
IDOA-DT	**98.00**

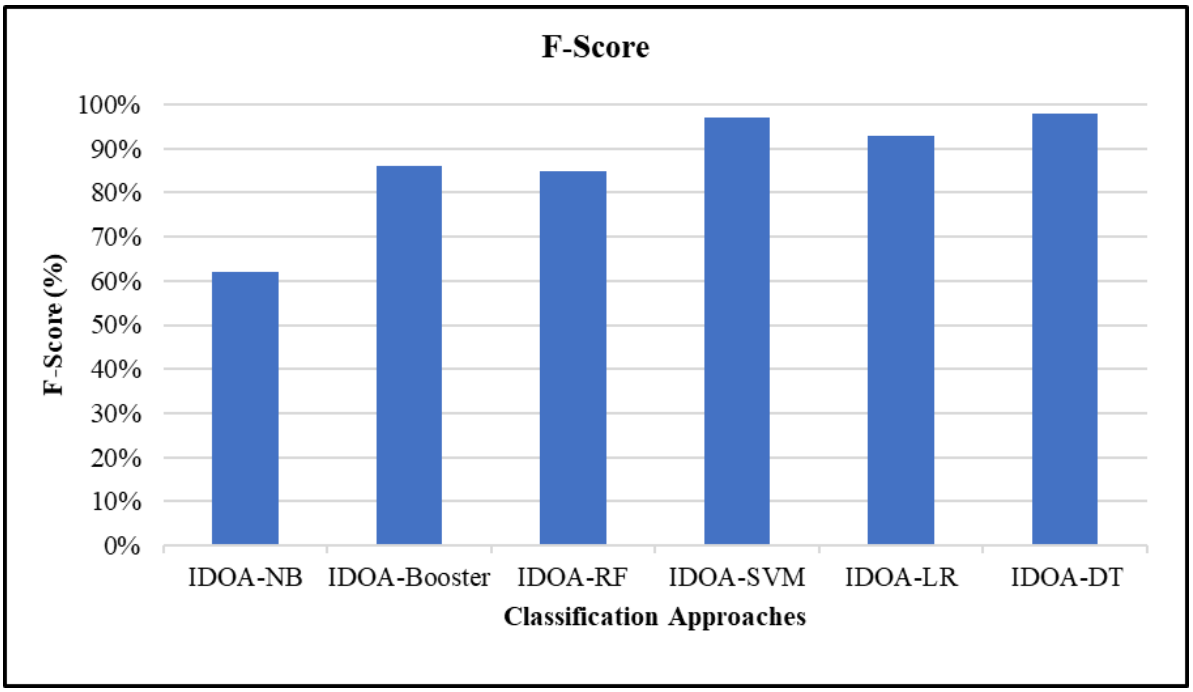

Figure 5.6 F-Score of the proposed Strategic level framework

Table 5.6 F-Score values of the proposed Strategic level framework

Approach	F-Score (%)
IDOA-NB	62.00
IDOA-Booster	86.00
IDOA-RF	85.00
IDOA-SVM	97.00
IDOA-LR	93.00
IDOA-DT	**98.00**

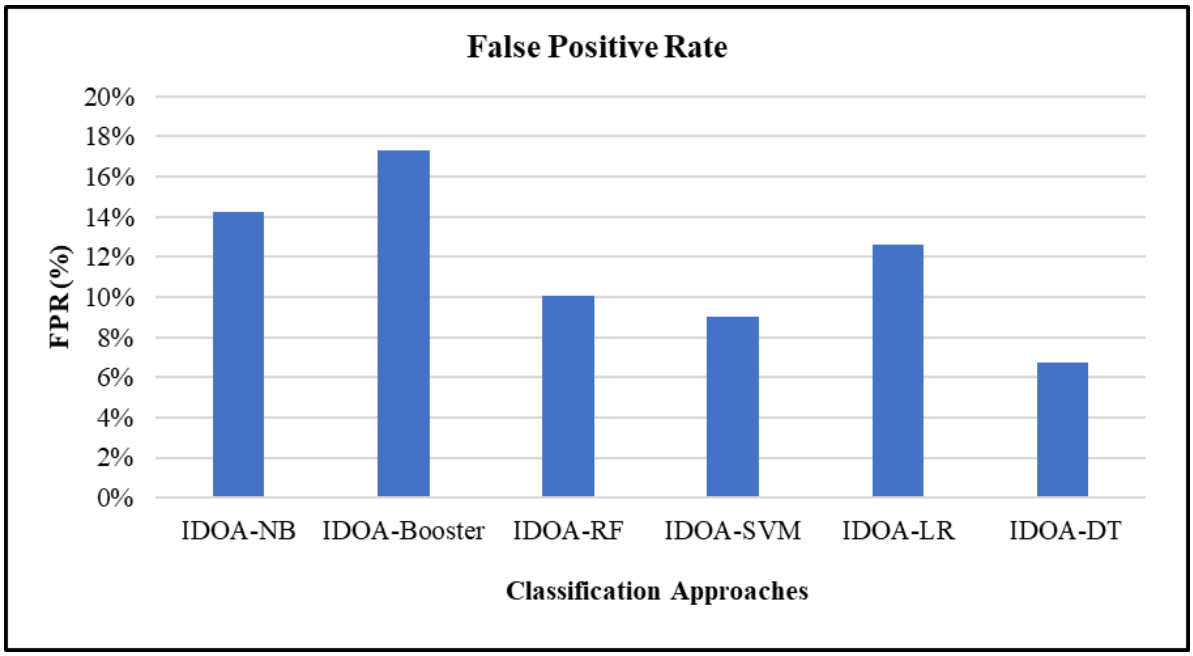

Figure 5.7 False Positive Rates of the proposed Strategic level framework

Table 5.7 False Positive Rate values of the proposed Strategic level framework

Approach	FPR (%)
IDOA-NB	14.22
IDOA-Booster	17.28
IDOA-RF	10.06
IDOA-SVM	9.00
IDOA-LR	12.59
IDOA-DT	**6.71**

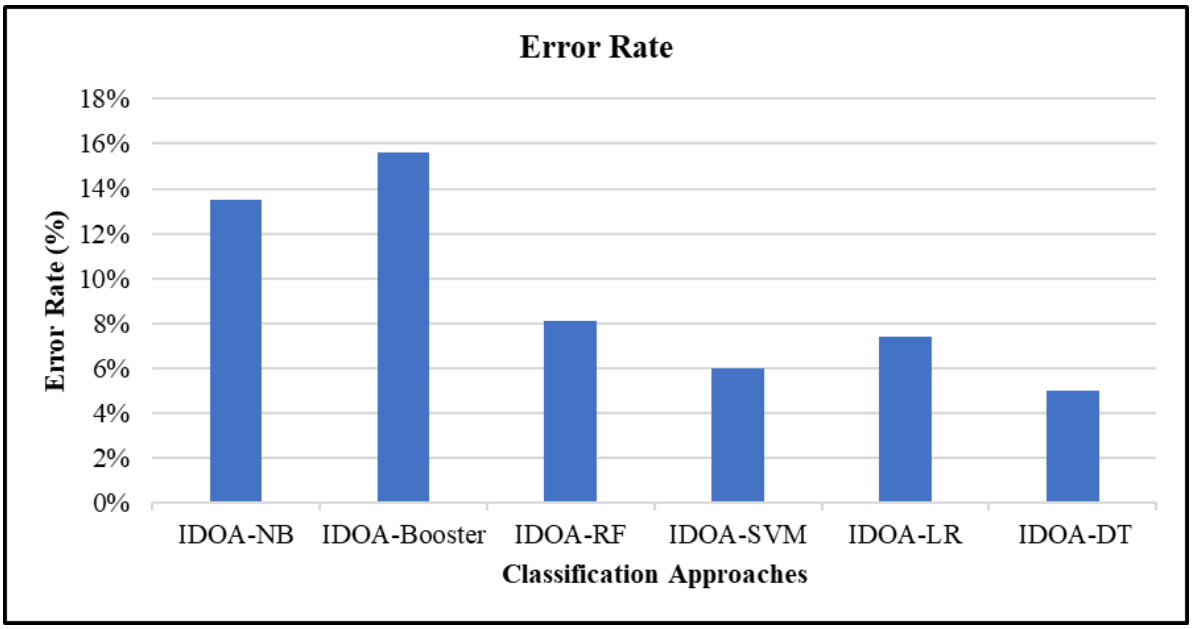

Figure 5.8 Error Rate of the proposed Strategic level framework

Table 5.8 Error Rate values of the proposed Strategic level framework

Approach	Error Rate (%)
IDOA-NB	**13.48**
IDOA-Booster	**15.62**
IDOA-RF	**8.09**
IDOA-SVM	**6.01**
IDOA-LR	**7.39**
IDOA-DT	**4.99**

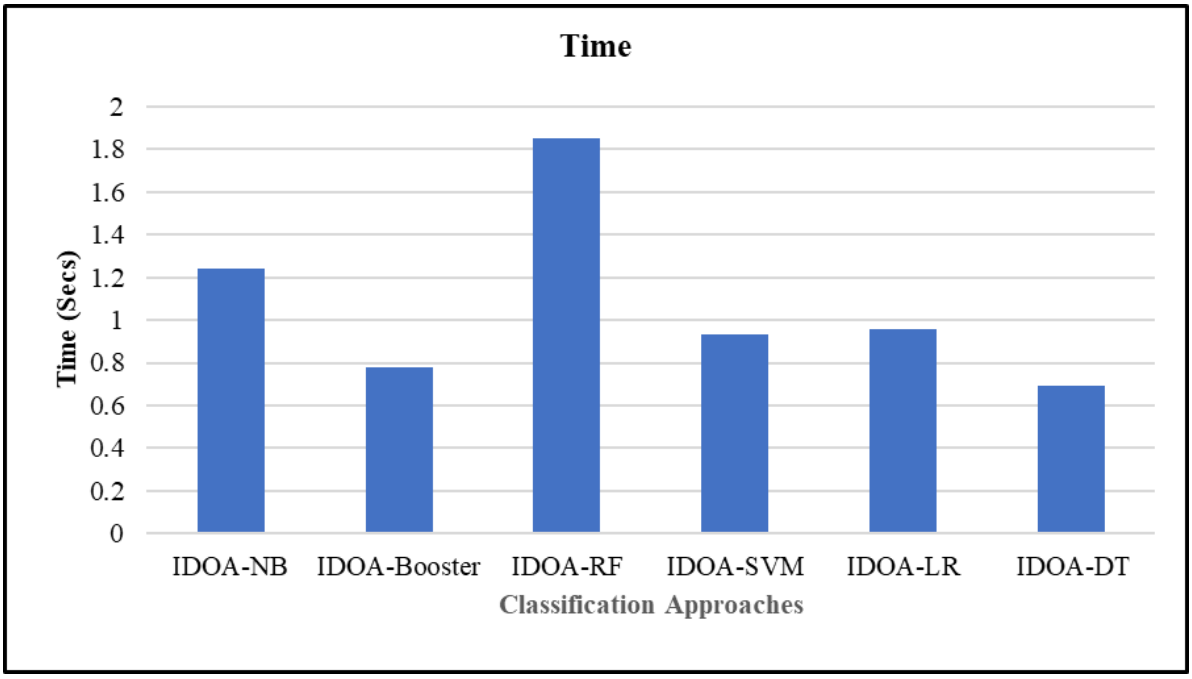

Figure 5.9 Detection Time of the proposed Strategic level framework

Table 5.9 Detection Time values of the proposed Strategic level framework

Approach	Time (s)
IDOA-NB	1.24
IDOA-Booster	0.78
IDOA-RF	1.85
IDOA-SVM	0.93
IDOA-LR	0.96
IDOA-DT	**0.69**

Table 5.10. Proposed framework model's execution and calculations

Approach	Accuracy (%)	Precision (%)	Recall (%)	F-Score (%)	Time (s)	FPR (%)	Error Rate (%)
IDOA-NB	67.00	67.00	71.00	62.00	1.24	14.22	13.48
IDOA-Booster	74.00	76.00	83.00	86.00	0.78	17.28	15.62
IDOA-RF	86.00	77.00	74.00	85.00	1.85	10.06	8.09
IDOA-SVM	92.00	93.00	96.00	97.00	0.93	9.00	6.01
IDOA-LR	83.00	92.00	88.00	93.00	0.96	12.59	7.39
IDOA-DT	**98.89**	**98.00**	**97.00**	**98.00**	**0.69**	**6.71**	**4.99**

Other performance among the metrics are displayed in Figure 5.3-5.9 and its relative values in table 5.3-5.9. Table 5.10 represents the classification metrics for the proposed strategic framework model with different alternatives. The proposed strategic framework integrates Improved Dragonfly Optimization techniques with various machine learning algorithms, demonstrating superior performance. When comparing the Improved Dragonfly Optimization Algorithm with a Decision Tree (IDOA-DT) to other combinations, the IDOA-DT approach yields the high-performance metrics: Precision (98.00%), Recall (97.00%), F-Score (98.00%), Accuracy (98.89%), and execution time (0.69). This highlights the efficacy of the IDOA-DT combination over other methods.

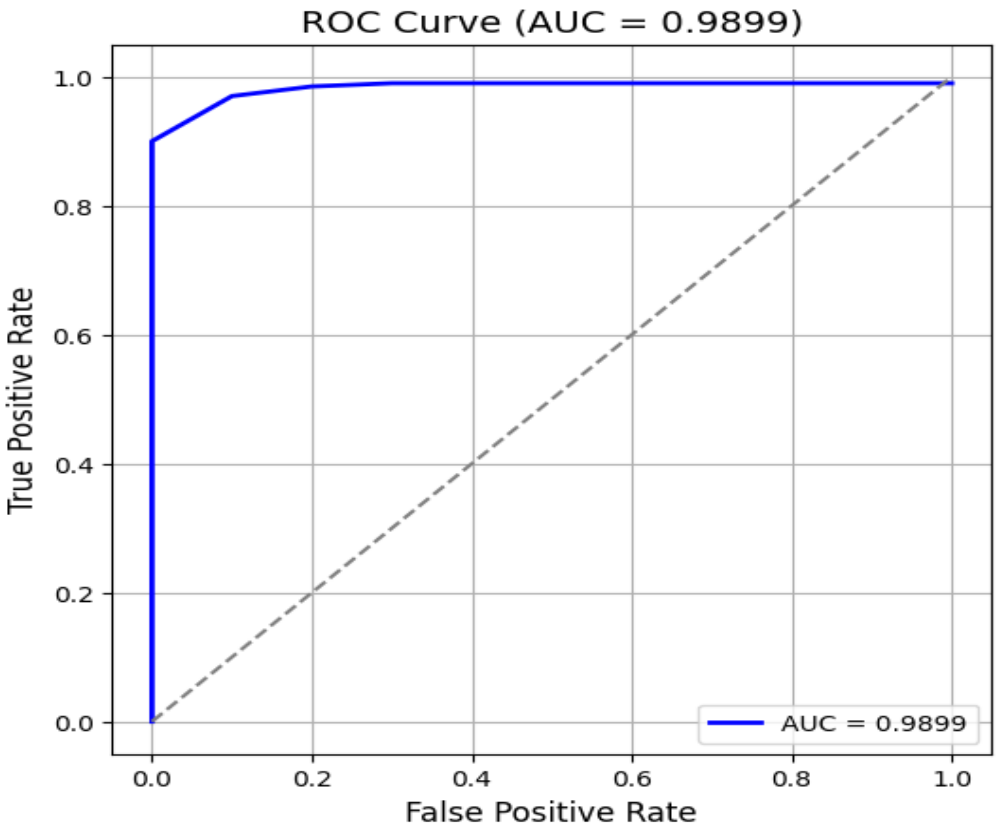

Figure 5.10 ROC Curve of proposed detection framework in Phase II

Receive Performance Curve (ROC) is employed to measure model execution exactly. The ROC bend illustrates the connection between the real and fake class boundaries and the x-hub structure, with the incorrect x-pivot y having a positive value of the various benefits of the various qualities between 0.0 and 1.0 (S. M. Kasongo and Y. Sun, 2019). The proposed visual system's ROC bend is shown in Figure 5.10, with an AUC of 98.8, indicating that the model can accurately detect 98.8% of positive and negative classes. In

the suggested model, favorably predicted values are denoted by TP and TN. FP and FN, on the other hand, serve as examples of the categorization.

The DDoS flooding attack identification model was analyzed and the effectiveness of the findings was confirmed using a 10-fold cross-validation approach which involves dividing the data set into ten parts. The one-fold is utilized to provide test data in each of these rounds, while the other nine folds are used for training. To offer a more thorough evaluation of the model's functionality it will use generic evaluation measures, which will be acquired repeatedly and then averaged. Because the whole dataset is used for both training and validation, this method is more accurate, lowers the possibility of overfitting, and enables validation to be done on unseen data. The distinction between attack and regular traffic, which is accomplished in the detection of the DDoS classification model, can be ensured when used in real-world operations. In Figure 5.11 below, the 10-fold validation procedure is displayed:

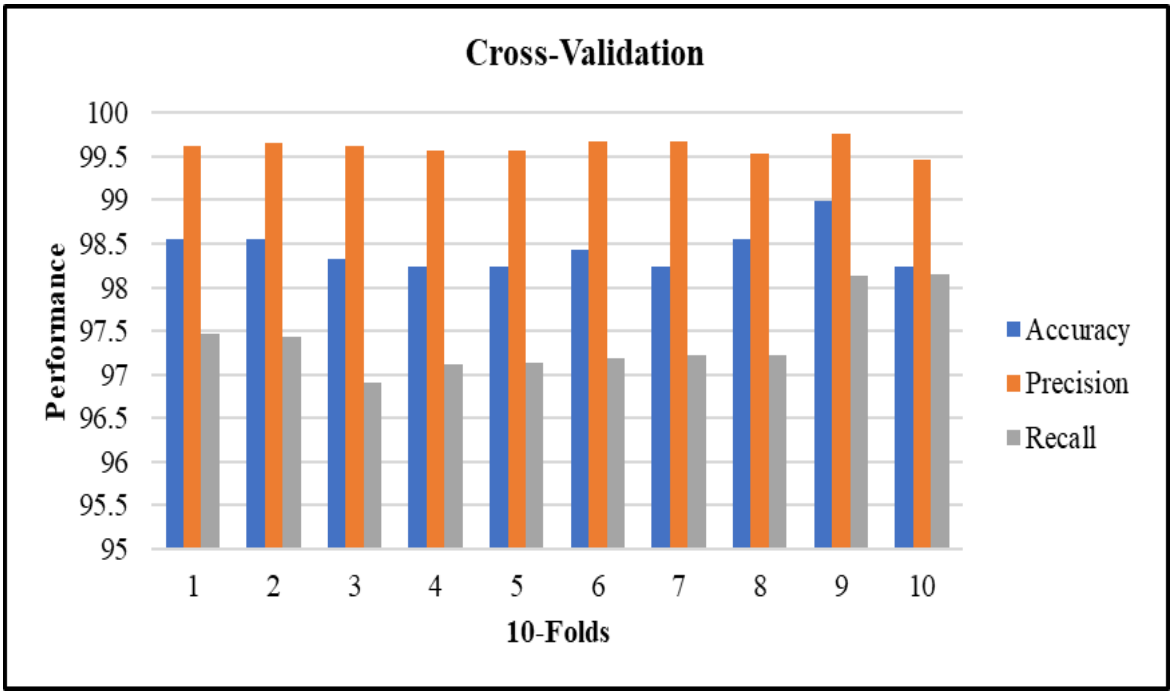

Figure 5.11 The bar model displays the DDoS Flooding attack detection model's ten-step validation procedure

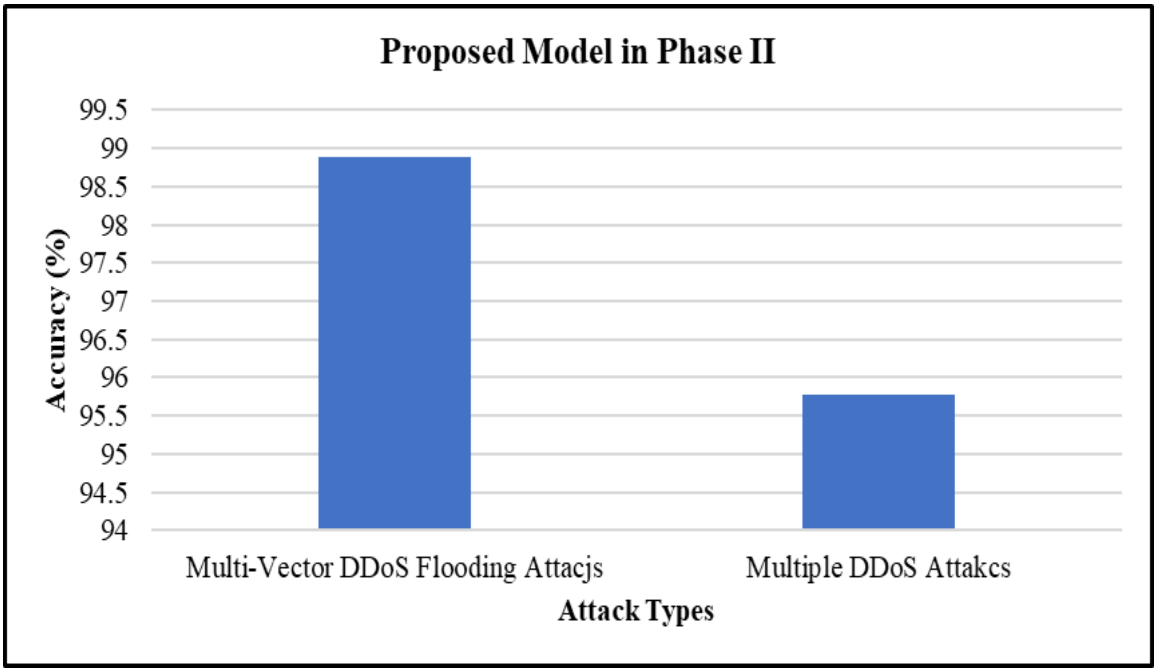

Figure 5.12 The accuracy graph of the proposed model for multiple DDoS attacks and multi-vector flooding attacks

The Figure 5.12 below demonstrates the accuracy obtained from the proposed model in Phase II for generating Multi-Vector DDoS Flooding Attacks and Multiple DDoS Attacks. The accuracy obtained for Multi-Vector DDoS Flooding Attacks is 98.89% whereas for Multiple DDoS Attacks Proposed Model the accuracy achieved is 95.77%. That means the proposed approach does a better job of recognizing Multi-Vector DDoS Flooding Attacks compared to Multiple DDoS Attacks. The percentage improvement in accuracy from Multiple DDoS Attacks to Multi-Vector DDoS Flooding Attacks is approximately 3.26%.

5.6 Chapter Summary

In conclusion, the efficiency of the proposed framework in Phase II, integrates the Improved Dragonfly Optimization Algorithm (IDOA) with Decision Tree (DT) classification, in enhancing DDoS attack detection is clearly demonstrated through notable improvements in performance metrics. The combination of IDOA for feature engineering and selection with DT classification resulted in a 7.49% improvement in accuracy in contrast to other approaches. In particular, the IDOA-DT strategy produced outstanding performance metrics: recall of 97.00%, F-score of 98.00%, and absolute accuracy of 98.89% with the selected approach, and a total accuracy of 98.00% execution time not

exceeding 0.69 seconds. The obtained results prove that the use of the IDOA-DT combination is more effective in comparison with other methods as a DDoS attack detector. Employing feature engineering and selection in this work making use of the logical ID of Improved Dragonfly Optimization Algorithm created several new germane features enhancing the model's precision accordingly.

The involvement of 10-fold cross-validated helped to reduce the level of over attainment and ensure the validity of the findings. The strategy framework used in this kind of activity is based on the hybridization of IDOA and DT does not show high accuracy, but also makes clear evidence that there is a decrease in detection time and less false alarms. The integration of the IDOA and the DT improved the detection performance by increasing the accuracy of detection, decreasing the time needed to detect the event, and minimizing the rate of false alarms. It also flourishes in the situation where the type of attack under consideration is a multi-vector DDoS flooding attack by applying the proposed method that makes it feature selection complexity-aware. They allow IDOA to be adaptive in complexity of the new threats, work with high accuracy even in cases of intricate data cases.

The model provides a platform for identifying the flooding attacks with accuracy stands at 98.89% for the multiple DDoS attacks is just 95.77%. This difference is indicative of a vulnerability, an indication that although the model is extraordinarily reliable with specific form of attack, it may not be as efficient with different form of DDoS attacks. The above disparity shows the optimized IDE for enhancements to deal with a larger range of DDoS attack types for better performance. They go further to highlight the importance of assimilation of both sophisticated features selection techniques and cutting edge machine learning techniques to augment IDSs to counter increasingly complex IT threats. The following chapter described the advanced bio inspired algorithms to manage multiple DDoS attacks.