Enhancing Classification Accuracy through Cluster-Based Ensemble Learning and Adaptive Weighting

Mustafa Radif[1], Zainab Fahad alnaseri[1], Salam saad alkafagi[2], Ali Hakem Al-saeedi[1], Riyadh Rahef Nuiaa Alogaili[3,*], Mazin Abed Mohammed[4,5]

[1]College of Computer Science and Information Technology, University of Al-Qadisiyah, Diwaniyah, Iraq

[2]Babylon Education Directorate, Ministry of Education, Babil, Iraq

[3]College of Computer Science and Information Technology, Wasit University, Al-Kut, Wasit, Iraq

[4]Department of Artificial Intelligence, College of Computer Science and Information Technology, University of Anbar, Anbar, Iraq

[5]College of science, Al-Farabi University, Baghdad, Iraq

Emails: mustafa.radif@qu.edu.iq; zainab.alnaseri@qu.edu.iq; salam.s.alkafagi@gmail.com; ali.alsaeedi@qu.edu.iq; riyadh@uowasit.edu.iq; mazinalshujeary@uoanbar.edu.iq

Abstract

As digital devices continue to process ever-increasing volumes of complex data, ensuring accurate and efficient machine learning performance has become a significant challenge. Traditional ensemble learning methods often attempt to address these issues through data sampling or partitioning; however, such approaches can introduce biases and fail to fully capture the underlying structure of the data. To address these limitations, this paper proposes a novel classification framework that integrates clustering with adaptive weighting strategies. The process begins by dividing the training data into clusters, each representing a specific subset of the overall data distribution. Separate machine learning models are then trained on these clusters, allowing each model to specialize in different areas of the data. When analyzing a test instance, its relationship to the individual clusters is evaluated using two key measures: the correlation coefficient, which assesses feature similarity, and the Mahalanobis distance, which calculates the statistical proximity to the cluster center. These values are subsequently used to generate optimized weights that determine the influence each model should have in the final ensemble prediction. By aligning model contributions with the structural similarities between the test and training data, the proposed approach enhances both the reliability and precision of classification. Experimental results demonstrate that this cluster-aware ensemble consistently outperforms both baseline and advanced classifiers on benchmark datasets.

Keywords: Ensemble learning; Correlation-based model; Similarity; Data mining; Machine learning

1. Introduction

The rapid development of data science has led to its widespread use in several fields of analysis, detection, or class faction [1]. For example, healthcare, education, finance, cybersecurity, and remote sensing. In many of these areas, large amounts of complex and high-dimensional data are generated and transmitted via various digital media and network infrastructures[2, 3]. This data is often considered big data because it has the three Vs: high volume, variety, and velocity. Moreover, it usually includes heterogeneous formats such as images, videos, sensor readings, and textual information that can traverse multiple layers of computing environments, from edge devices to fog nodes to centralized cloud servers [4-6].

Efficient handling, processing, and analysis of such complex data is essential to ensure timely decision-making (correct) and maintain the quality and integrity of the information as it travels through the various communication channels [6, 7]. Ensemble methods have become a powerful approach for enhancing classification performance

DOI: https://doi.org/10.54216/FPA.210109
Received: February 25, 2025 Revised: June 05, 2025 Accepted: July 10, 2025

[7]. They enhance classification performance by aggregating the predictions of multiple base classifiers. As a result, ensemble models often achieve higher accuracy, improved robustness, and better generalizability compared to individual classifiers[8]. Figure 1. Illustrates the basic scenario of the ensemble techniques:

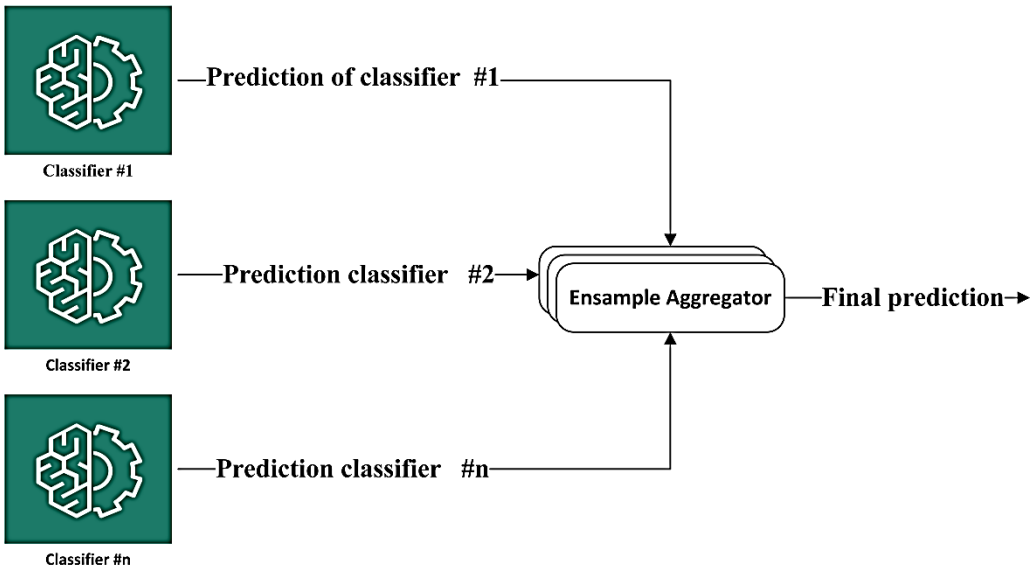

Figure 1. Scenario of ensemble techniques

Table 1 presents various aggregation strategies employed in ensemble learning and illustrative examples for each method.

Table 1: The possible strategies of ensemble techniques

Strategy	**Description**	**Example**
Majority Vote	Picks the label predicted by most classifiers	["A", "A", "B"] → "A"
Weighted Vote	Each classifier's prediction is weighted by confidence/performance.	"A" (0.8), "B" (0.6) → "A"
Averaging	For numeric values (regression).	[0.7, 0.8, 0.9] → 0.8
Stacking	Feeds all predictions to a final model (meta-learner).	Meta-model decides the final class

Despite the advantages of ensemble methods, they frequently face challenges when confronted with imbalanced or heterogeneous data distributions [9-11]. A common strategy to address such issues is the application of sampling techniques, such as oversampling, under sampling, or stratified sampling. These techniques aim to balance class distributions or create more representative training subsets, which can mitigate bias and improve performance [5, 12]. However, sampling can also introduce artificial data artefacts, alter the original data distribution, and sometimes result in losing important information. This raises questions about the reliability and scalability of sampling-based ensemble models, especially when applied to real-world datasets with complex structures.

The main reason for this research is that standard ensemble methods, which usually shuffle data by random sampling, do not always give the best results. Instead, this work utilizes the existing natural groups in the data. By looking at how data points are naturally grouped, it becomes easier to recognize which samples belong where. This helps the model make smarter decisions by giving more importance to the clusters that a test sample most closely resembles, which can lead to predictions that are more accurate. This paper introduces a new approach to ensemble classification that moves away from traditional sampling methods. Instead, it uses cluster-based optimization weights, which are uniquely calculated for each test sample. The process works by assessing how closely a sample aligns with different cluster centroids—in terms of both feature patterns (using correlation

DOI: https://doi.org/10.54216/FPA.210109
Received: February 25, 2025 Revised: June 05, 2025 Accepted: July 10, 2025

coefficients) and statistical distance (through Mahalanobis distance). By combining these two factors, the model can better determine how well a sample fits into different clusters. These insights then help create optimization weights that enhance the classification process, leading to more precise and informed decisions. The main contributions of this research can be summarized as follows:

1. Development of a new ensemble classification framework that employs cluster-based optimization weights derived from correlation and Mahalanobis distance, eliminating the need for data sampling techniques.
2. A dual-weighting mechanism was introduced that quantifies the pattern similarity and statistical proximity of test samples to clusters, providing a richer basis for classifier voting within the ensemble.
3. Comprehensive experimental validation on benchmark datasets, demonstrating improved classification performance and robustness compared to conventional sampling-based ensemble methods.
4. Analysis of the framework's applicability to various domains, with a discussion of strengths, limitations, and potential areas for further research.

The structure of this thesis is as follows. Section 2 reviews related work. Section 3 details materials and techniques. Section 4 reports experimental results and comparative analysis. Section 5 concludes and future works.

2. Related Work

Ensemble learning and cluster-based weighting have been widely investigated to improve classification performance, particularly for complex or imbalanced datasets. Several studies have used this technique to enhance performance model decisions. However, these models still suffer from weaknesses highlighted in this section.

Kanthimathi et al. (2023) [13] proposed a new framework of a self-attention-enabled ensemble system that integrates CNN with XGBoost, LSTM, and Random Forest. The system improves DDoS detection by combining diverse model predictions and capturing temporal patterns. Despite achieving high accuracy, its complex architecture increases computational load, making a real-time application challenging.

Farahmandnia and Özekes (2022) [14] introduced a hybrid intrusion detection system based on a stacked ensemble framework combining KNN, SVM, decision trees, and random forest. This approach enhances classification performance with reduced training time. However, its performance in real-time network environments remains underexplored.

Han et al. (2020) [15] developed a system that leverages a biologically inspired particle swarm optimization algorithm, where each consortium member operates a private chain while anomalous data is stored on a public chain to ensure tamper-proof monitoring. Smart contracts enhanced with fuzzy neural network algorithms distinguish DDoS data and generate anomaly chains on each node, enabling rapid exception chain formation and effective collaborative detection while safeguarding data privacy. Evaluated on the CICDDoS2019 (DDoS DNS) dataset, the system achieved a detection accuracy of 89.8%, demonstrating its efficacy in protecting user data.

Beulah and Manickam (2022) [16] developed an ensemble detection method combining support vector machines, logistic regression, and a voting mechanism. Their work focuses on improving the robustness of detection systems against DDoS attacks. Yet, it lacks evaluation on unseen or zero-day attacks, limiting its generalization.

Thorat et al. (2021) [17] introduced TaxoDaCML, a taxonomy-based divide-and-conquer approach using ML for detecting prominent DDoS attacks. The approach considers four levels to classify one of 11 attack types, with level 4LL focusing on DNS-based DrDoS attacks. Utilizing the CICDDoS2019 dataset initially comprising 88 features, data cleaning retained 67 features and 1 label, upon which statistical feature selection techniques such as analysis of variance and mutual information were applied to identify 20 key DNS-related features for level 4LL. This approach achieved a detection accuracy of 69.8%.

Usha et al. (2021) [18] proposed an efficient DDoS detection system that leverages various machine learning algorithms (including extreme gradient boosting, K-nearest neighbor, stochastic gradient descent, and Naïve Bayes) alongside a deep learning architecture (convolutional neural network) for intrusion detection and attack classification. The system classifies attacks using 34 selected network traffic features and evaluates model performance on the test dataset based on F1 scores, accuracy, and recall.

Akgun et al. (2022) [19] developed a deep learning-based system for high-accuracy detection of DDoS attack types by evaluating Dense Neural Networks (DNN), Convolutional Neural Networks (CNN), and Long Short-Term Memory (LSTM) architectures on the widely used CICDDoS2019 dataset. Balanced sub-datasets were generated by randomly partitioning the original dataset, with repetitive samples removed during feature selection

DOI: https://doi.org/10.54216/FPA.210109
Received: February 25, 2025 Revised: June 05, 2025 Accepted: July 10, 2025

using an InfoGain attribute evaluation algorithm to reduce dataset size and enhance performance. Based on these feature selection operations, a refined dataset comprising 40 features was produced, enabling the identification of the most effective deep learning model in terms of accuracy and inference performance.

Han et al. (2020) [7] developed a blockchain-integrated DDoS detection model using particle swarm optimization and fuzzy neural networks. It leverages private and public chains for anomaly storage and privacy preservation. The system achieved 89.8% accuracy on the CICDDoS2019 dataset, though its complexity may hinder large-scale implementation.

3. Materials and Methods

3.1. Dataset

This work employs the CICDDoS2019 dataset, as shown in Table 1

Table 2: Summarization of CICDDoS2019-DrDoS_DNS dataset [18]

Attribute	**Details**
Number of Features	84
Number of Samples	+1 million samples
Number of Classes	2 (Benign, DNS DDoS)

3.2. Ensemble Learning Theory

Ensemble learning is a machine learning paradigm where multiple individual models-often referred to as base learners-are trained to solve the same task and combined to produce a more robust and accurate prediction[10, 20]. Mathematically, if each model $h_i(x)$ produces a prediction for input x, Then the ensemble's output is typically a function. (as shown in equation 1)

$$H(x) = f(h_1(x), h_2(x), \dots, h_n(x)) \quad (1)$$

Where: $h_i(x)$ is the prediction from the $i$the base model, f is an aggregation function (e.g., majority voting, averaging),and $H(x)$ is the final ensemble output.

Equation 2 uses for averaging (used in regression or probabilistic classification)[21]:

$$H(x) = \frac{1}{n}\sum_{i=1}^{n} h_i(x) \quad (2)$$

This formula reduces variance by balancing fluctuations from individual models.

3.3. Unified Ensemble Classification

Ensemble methods aggregate a multi-classifier decision into one block [22]. It uses a weighted sum decision rule in the final decision. For example, an input x lets each classifier Li produce a score or probability pi(x,y)for class y. Equation 3 will give the final decision:

$$y^* = \arg\max_{y} \sum_{i=1}^{K} w_i\, p_i\,(x, y) \quad (3)$$

Where wi are the weighting factors, this unified structure avoids duplication and simplifies the integration of multiple decision strategies.

The method is inherently robust because each model within the ensemble contributes a unique perspective[23]. This diversity minimizes the impact of individual model errors. It creates a model less sensitive to outliers or anomalies in the data and mimics a more balanced one. Moreover, it introduces a decision like a human decision process. The main disadvantages include increased system complexity and higher computational costs, as managing multiple classifiers and calibrating their weights requires additional resources and careful design [24]. There is also a risk of overfitting if the base classifiers are too similar, and the ongoing maintenance and tuning required can make it challenging to use in dynamic or resource-constrained environments [25].

DOI: https://doi.org/10.54216/FPA.210109
Received: February 25, 2025 Revised: June 05, 2025 Accepted: July 10, 2025

3.4. Proposed model

Figure 2 shows the primary architecture of the proposed solution.

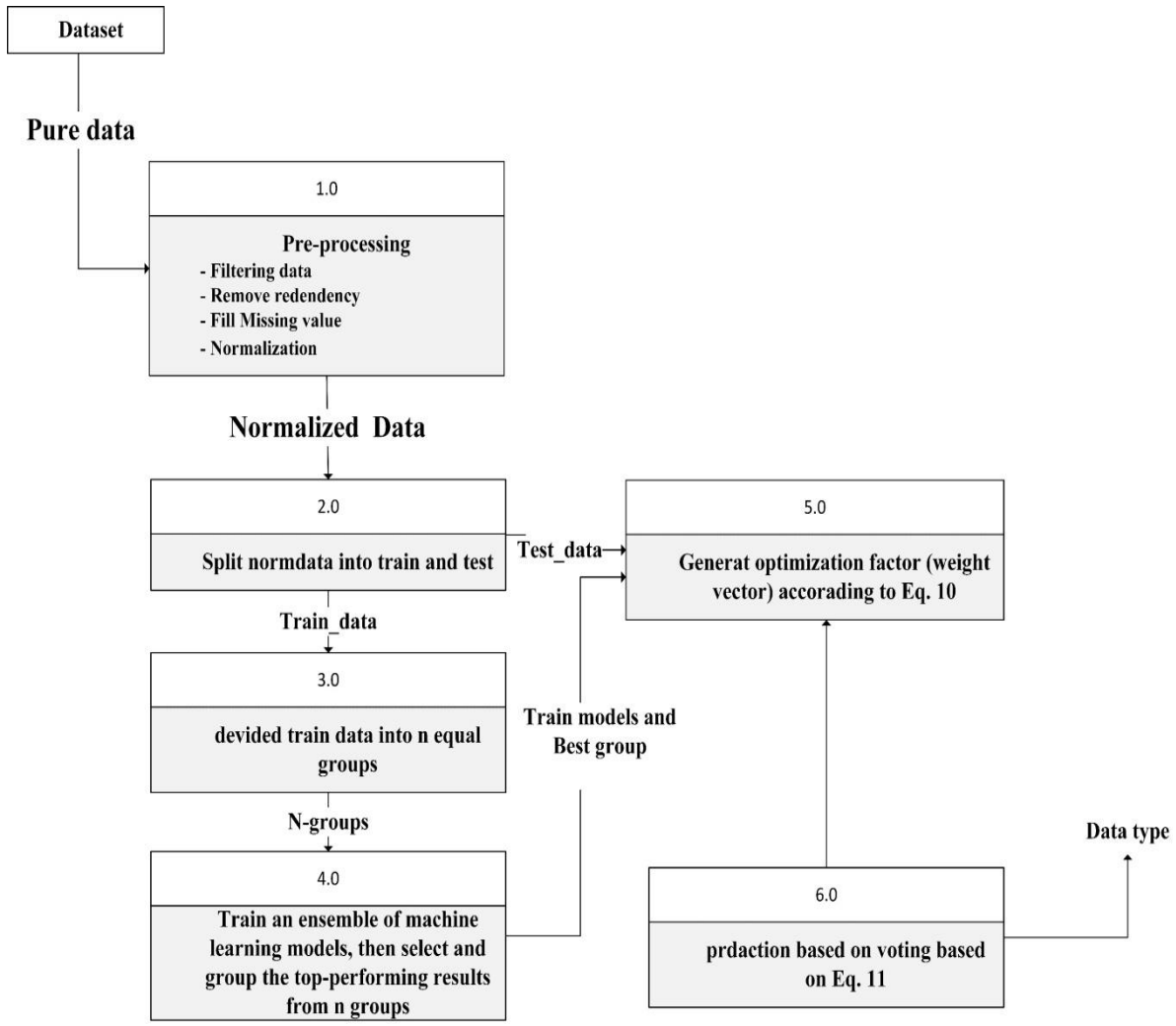

Figure 2. Proposed Model Architecture

3.3.1 Pre-processing

The pre-processing step corresponds to process 1.0 in Figure 2. It consists of four steps:

1. **Filter data:**

   If the entire dataset contains broader data, extract only those records corresponding to the model objective.

2. **Removing Redundancy:**

   The element duplicates or repeats records that do not add new information. This process removes identical data points in the dataset.

3. **Handling Missing Values:**

The mean imputation of missing data in the sample (xj) is defined by Eq. (3).

$$x_j' = \begin{cases} x_j, & if\ x_j \neq missing \\ \frac{1}{N}\sum_{i=1}^{N} x_i, & if\ x_j = missing \end{cases} \quad (3)$$

Where: N diminution of the sample (x_j).

4. **Normalization**

DOI: https://doi.org/10.54216/FPA.210109
Received: February 25, 2025 Revised: June 05, 2025 Accepted: July 10, 2025

The z-score standardization normalizes is used to normalize data.

3.3.2 Splitting Normalized Data into Train and Test

In this process, the proposed model splits data into 70% for training and 30% for testing and evaluates the proposed model.

3.3.3 Dividing Train Data into n Equal Groups

Once the training set is determined, the proposed model divides it into n groups. This division approach potentially helps the proposed classification model to increase diversity among models, which is often helpful in ensemble methods. Moreover, it will parallelize model training and evaluate how each subset's model (or ensemble) performs.

The proposed scenario of dividing the train data (X_{train}) into n groups is defined as follows:

Equation 4 calculates the number of training samples ($|X_{train}|$) after splitting the dataset into train and test sets.

$$|X_{train}| = \alpha M \tag{4}$$

Where: M is the total number of samples in the full (pre-processed) dataset. α is the proportion of the dataset designated for training (e.g., 0.7 for 70% training).

The size of each group (g) is calculated by equation (5)

$$g = \frac{\alpha M}{n} \tag{5}$$

The groups of train data are defined in Equation (6).

$$X_{train} = \bigcup_{k=1}^{n} X_k, y_{train} = \bigcup_{k=1}^{n} y_k \tag{6}$$

Where: k is the index of a particular group, X_{train} is the entire training feature set (consisting of αM samples), and y_{train} corresponding training labels (Attack/Normal), matched to X_{train}.

3.3.4 Training an Ensemble of Machine Learning Models

The proposed model trains based on several ML models (Random Forest, SVM, and XGBoost) on each n group of train data (X_{train} , y_{train}). Denote the set of models trained in the group k by $\{h_{k,1}, h_{k,2}, \ldots, h_{k,p}\}$. This stage consists of two processes:

1. Ensemble Construction:

Each model outputs a prediction $\hat{y}_{k,j} = h_{k,j}(x)$ for a sample *x*.

2. Selection of Top-Performing Models:

After training, evaluate each model's performance (e.g., accuracy, F1-score, etc.) on a validation set or via cross-validation. Pick the top models (or average them) to form a more substantial ensemble from that group k.

From each group, the proposed model keeps only a single best-performing model $h_{k'}^{*}$, or you might keep a sub-ensemble of each group's top models. This leads to a final ensemble, as shown in Eq. (7) :

$$H = \{h_{1,}^{*} h_{2,}^{*}, \ldots, h_{n}^{*}\} \tag{7}$$

3.3.5 Generating an Optimization Factor

In this step, the proposed model calculates weight factors for each test point $X_{test,\,j}$ based on how well it "fits" (i.e., belongs) to each cluster in Group *H*. We quantify this "fitness" or membership using two key measures:

1. A Correlation Coefficient $\rho(X_{test,\,j}, C_i)$ between the test point and cluster C_i.
2. A Mahalanobis Distance $d_M(X_{test,\,j}, C_i)$ between the test point and cluster C_i.

Correlation Coefficient

A (linear) correlation coefficient measures how similar the test point is to the "pattern" of the cluster. One way is to treat $x_{test\,,j}$ and the cluster centroid μ_i as vectors and compute Pearson's *r*. As shown in Equation. (8) :

DOI: https://doi.org/10.54216/FPA.210109
Received: February 25, 2025 Revised: June 05, 2025 Accepted: July 10, 2025

$$\rho(\mathbf{x}_{\text{test},j}, C_i) = \frac{\sum_{f=1}^{F}(x_{\text{test},j,f}-\bar{x}_j)(\mu_{i,f}-\bar{\mu}_i)}{\sqrt{\sum_{f=1}^{F}(x_{\text{test},j,f}-\bar{x}_j)^2}\sqrt{\sum_{f=1}^{F}(\mu_{i,f}-\bar{\mu}_i)^2}} \quad (8)$$

where: F is the number of features, $x_{\text{test},j,f}$ is the f feature of the test sample j, $\mu_{i,f}$ is the f-th component of the cluster i's centroid, $\bar{x}_j$ is the average of the features of the j-th test sample.

Mahalanobis Distance d_M

The Mahalanobis Distance between a test point $\mathbf{x}_{\text{test},j}$ and cluster C_i is calculated in Equation. (9) :

$$d_M(\mathbf{x}_{\text{test},j}, C_i) = \sqrt{(\mathbf{x}_{\text{test},j} - \boldsymbol{\mu}_i)^{\mathrm{T}} \boldsymbol{\Sigma}_i^{-1} (\mathbf{x}_{\text{test},j} - \boldsymbol{\mu}_i)} \quad (9)$$

Where: $\mathbf{x}_{\text{test},j}$: The feature vector of the j-th test point, $\boldsymbol{\mu}_i$: The mean (centroid) vector of the i-th cluster.$\boldsymbol{\Sigma}_i$: The covariance matrix of the i-th cluster.$\boldsymbol{\Sigma}_i^{-1}$: The inverse of the covariance matrix for cluster i.and $(\cdot)^{\mathrm{T}}$: The transpose of a vector.

The $\delta_{j,i}$ is calculated based on Equation. (10):

$$\delta_{j,i} = \frac{1}{2}\left[\rho(\mathbf{x}_{\text{test},j}, C_i) + d_M(\mathbf{x}_{\text{test},j}, C_i)\right] \quad (10)$$

Finally, the $\delta_{j,i}$ normalize across all clusters C_i in group H to represent set of weights $\{w_{j,i}\}$ for each test sample j as shown in the Equation. (11):

$$w_{j,i} = \frac{\mu_{j,i}}{\sum_{k \in H} \delta_{j,k}}, \; i \in H \quad (11)$$

Algorithm 1 describes the scenario of split data based on correlation matric.

Algorithm1: Dividing Data into n Equal Groups

Input: X ← train dataset, n ← number of groups, g← size of each group.

Output: n- groups

1. **Compute the Pearson correlation coefficient between all pairs of samples.**
2. **Construct a correlation matrix C.**
3. **Apply a clustering algorithm that ensures high correlation within each group.**
4. **If the clustering results in imbalanced groups, Move the least correlated sample from a larger group to a smaller group.**
5. **Repeat step (4) until all groups have exact members.**

Return: n- groups

3.5. Experiment Result

In this section, the proposed model is tested and evaluated using the CICDDoS2019 dataset, as described in Section II-A. The performance of the proposed model is compared with that of standard machine learning algorithms—namely, Decision Tree, Random Forest, and XGBoost—as well as with recent models that detect DrDoS attacks using the CICDDoS2019 dataset. The CICDDoS2019 has more than one million records; therefore, this paper used 10% of the original size of the dataset.

Table 2 presents a comparative performance evaluation of proposed MCO and ML models based on four key metrics: Accuracy, Precision, Recall, and F1-score.

DOI: https://doi.org/10.54216/FPA.210109
Received: February 25, 2025 Revised: June 05, 2025 Accepted: July 10, 2025

Table 3: The result of the proposed model and standard ML

Model	**Accuracy**	**precision**	**recall**	**F1-score**
Decision Tree	90.53	94.84	92.63	90.53
Random Forest	94.10	91.35	97.27	94.22
XGBoost	92.12	97.39	95.62	94.95
Proposed MCO	98.5	97.88	99.8	98.93

As shown in Table 1, the Decision Tree demonstrates moderate performance, with 90.53% accuracy and F1-score, alongside high precision (94.84%) and recall (92.63%), suggesting reasonable classification capability but limited generalizability. Random Forest outperforms the Decision Tree in accuracy (94.10%) and recall (97.27%), though its precision (91.35%) is comparatively lower, reflecting a trade-off between minimizing false positives and maximizing true positives. XGBoost achieves the highest precision (97.39%), and a balanced F1-score (94.95%), indicating strong discriminatory power, yet its accuracy (92.12%) trails Random Forest. Notably, the proposed MCO model exhibits superior performance across all metrics: it achieves the highest accuracy (98.5%), near-perfect recall (99.8%), and exceptional F1-score (98.93%), signifying an optimal balance between precision (97.88%) and sensitivity.

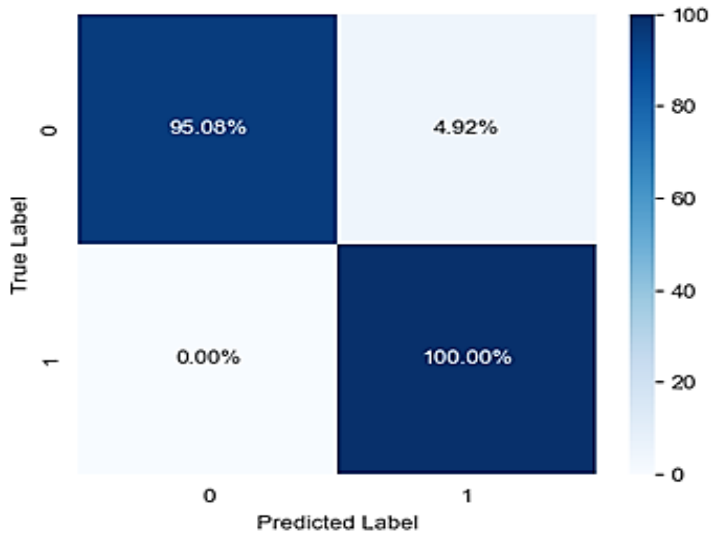

Figure 1. Confusion matrices of the proposed MCO model

Figure 3 presents the confusion matrix for the proposed MCO model, displaying its performance in classifying instances between two classes: 0 (Benin) and 1 (DNS-DRDoS). The matrix reveals that the model achieves a high level of accuracy in its predictions. For class 0, the true negative rate is 95.08%, while only 4.92% of class 0 instances are misclassified as class 1. For class 1, the model demonstrates exceptional performance with a 100% true positive rate, indicating no misclassifications.

Figure 4 illustrates the Receiver Operating Characteristic (ROC) curve for the proposed MCO model, demonstrating its ability to distinguish between classes effectively. The ROC curve is plotted with the True Positive Rate (TPR) against the False Positive Rate (FPR), and the model achieves an Area Under the Curve (AUC) of 99.9, signifying near-perfect classification performance.

DOI: https://doi.org/10.54216/FPA.210109
Received: February 25, 2025 Revised: June 05, 2025 Accepted: July 10, 2025

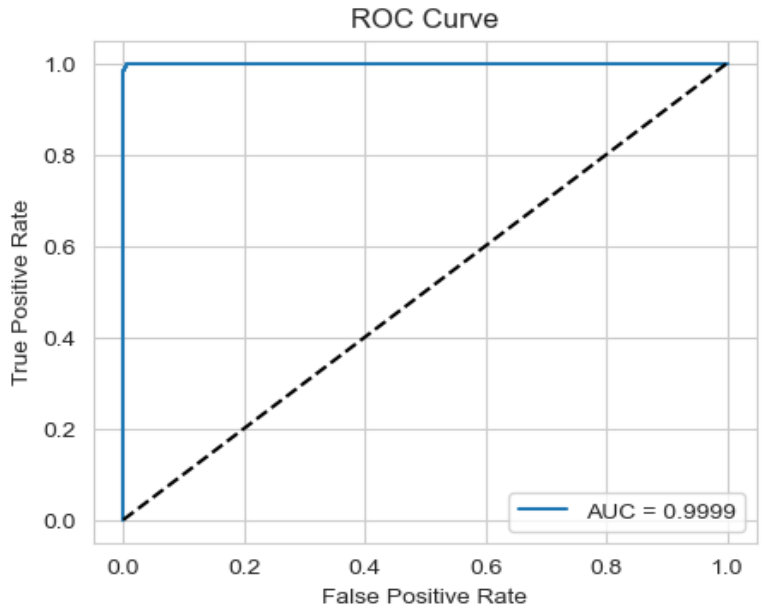

Figure 2. ROC curve of the proposed MCO model

The steep slope of the curve towards the upper left corner of the graph reflects the model's high sensitivity and low false positive rate. The AUC value close to 1 confirms the superior discriminative ability of the proposed MCO model and makes it highly reliable for detection tasks where minimizing misclassification is crucial. This also underpins the excellent results observed in the confusion matrix and performance metrics

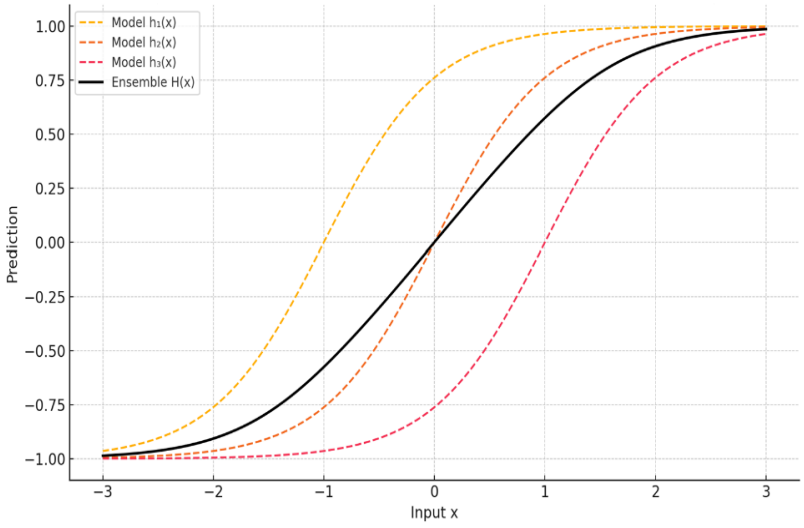

Figure 5. The train of each clusters passed on precession /recall plot

Table3 provides a comparative analysis of the proposed MCO model against several state-of-the-art models based on four key performance metrics: Accuracy, Precision, Recall, and F1-score.

DOI: https://doi.org/10.54216/FPA.210109
Received: February 25, 2025 Revised: June 05, 2025 Accepted: July 10, 2025

Table 4: The result of the proposed model and stat or the art models

Model	**Accuracy**	**precision**	**recall**	**F1-score**
[14]	89.8	100	87.56	93.37
[15]	69.8	80.9	58.4	67.8
[16]	89.29	None	89.29	87.7
[17]	99.30	98.43	98.36	98.40
[18]	94.57	80.49	95.15	87.21
[19]	95.44	99.75	91.11	95.23
Proposed MCO	98.5	97.88	99.8	98.93

The proposed MCO model achieves an accuracy of 98.5%, which is marginally lower than [17], with 99.30%, but outperforms most other models, such as [14] (89.8%), [15] (69.8%), and [18] (94.57%). In terms of precision, the proposed model (97.88%) shows significant improvements compared to [18] (80.49%) and [15] (80.9%), though it is slightly lower than [19], which achieves the highest precision at 99.75%.

For recall, the proposed MCO model demonstrates exceptional performance with 99.8%, outperforming all other models, including [17] (98.36%) and [19] (91.11%). This indicates the model's superior ability to identify positive instances, reducing false negatives effectively and correctly. The F1-score of the proposed model, 98.93%, is also among the highest, showing a well-balanced trade-off between precision and recall. It slightly surpasses [17] (98.40%) and significantly outperforms other models, such as [18] (87.21%) and [15] (67.8%)

4. Conclusion

Since it factors both correlation coefficients and Mahalanobis distances in its cluster processing, the new MCO ensemble model constitutes a welcome step forward, most of all for classification accuracy, by dispensing with traditional sampling and exploiting the data's intrinsic structure, this model lends itself to much more reliable and context-sensitive decision-making. The proposed model initially measures the similarity between data instances and redistributes them so those with similar behavioral patterns are grouped into the same cluster. This clustering process enhances the classification accuracy by allowing each machine-learning algorithm to be trained on more homogeneous subsets of the data. Subsequently, the model selects the most suitable classifier for each group, improving prediction performance. One of the key strengths of the proposed approach lies in its flexibility, as it allows for the integration of multiple machine learning algorithms tailored to the characteristics of each cluster.

Future work could explore ways to reduce computational overhead, such as dimensionality reduction or approximate distance computations. Additionally, extending the model to support incremental or online learning scenarios could make it more applicable to real-time systems. Incorporating adaptive mechanisms to re-cluster data on-the-fly or refine ensemble weights as new data arrives may further boost its utility in evolving, real-world applications.

References

[1] S. M. Hadi, A. H. Alsaeedi, D. Al-Shammary, Z. A. A. Alyasseri, M. A. Mohammed, K. H. Abdulkareem, R. R. Nuiaa, and M. M. Jaber, "Trigonometric words ranking model for spam message classification," *IET Networks*, 2022.

[2] F. S. Alrayes, M. Maray, A. Alshuhail, K. M. Almustafa, A. A. Darem, A. M. Al-Sharafi, and S. D. Alotaibi, "Privacy-preserving approach for IoT networks using statistical learning with optimization algorithm on high-dimensional big data environment," *Scientific Reports*, vol. 15, no. 1, p. 3338, 2025.

[3] A. J. Thompson, B. R. Smith, and C. L. Davis, "A review of machine learning techniques for cybersecurity," *Journal of Computer Virology and Hacking Techniques*, vol. 17, no. 3, pp. 153-167, 2021.

[4] H. Kuchuk and E. Malokhvii, "Integration of IoT with cloud, fog, and edge computing: a review," *Advanced Information Systems*, vol. 8, no. 2, pp. 65-78, 2024.

[5] R. R. Nuiaa, S. Manickam, and A. S. D. Alfoudi, "Dynamic Evolving Cauchy Possibilistic Clustering Based on the Self-Similarity Principle (DECS) for Enhancing Intrusion Detection System," 2022.

DOI: https://doi.org/10.54216/FPA.210109
Received: February 25, 2025 Revised: June 05, 2025 Accepted: July 10, 2025

[6] M. T. Nguyen and H. L. Tran, "An ensemble learning approach for detecting network intrusions," *Journal of Network and Computer Applications*, vol. 200, p. 103278, 2023.

[7] D. Sargiotis, "Data Quality Management: Ensuring Accuracy and Reliability," in *Data Governance: A Guide*, Springer, 2024, pp. 197-216.

[8] Z. Sultana, M. Foysal, S. Islam, and A. Foysal, "Lung cancer detection and classification from chest CT images using an ensemble deep learning approach," in *2024 6th International Conference on Electrical Engineering and Information & Communication Technology (ICEEICT)*, 2024: IEEE, pp. 364-369.

[9] W. Chen, K. Yang, Z. Yu, Y. Shi, and C. L. P. Chen, "A survey on imbalanced learning: latest research, applications and future directions," *Artificial Intelligence Review*, vol. 57, no. 6, p. 137, 2024.

[10] M. Sakib, S. Mustajab, and M. Alam, "Ensemble deep learning techniques for time series analysis: a comprehensive review, applications, open issues, challenges, and future directions," *Cluster Computing*, vol. 28, no. 1, pp. 1-44, 2025.

[11] B. H. Aubaidan, R. A. Kadir, M. T. Lajb, M. Anwar, K. N. Qureshi, B. A. Taha, and K. Ghafoor, "A review of intelligent data analysis: Machine learning approaches for addressing class imbalance in healthcare-challenges and perspectives," *Intelligent Data Analysis*, p. 1088467X241305509, 2024.

[12] M. O. Lawrence, R. G. Jimoh, and W. B. Yahya, "An efficient feature selection and classification system for microarray cancer data using genetic algorithm and deep belief networks," *Multimedia Tools and Applications*, vol. 84, no. 8, pp. 4393-4434, 2025.

[13] S. Kanthimathi, S. Venkatraman, K. S. Jayasankar, T. P. Jiljith, and R. Jashwanth, "A Novel self-attention-enabled weighted ensemble-based convolutional neural network framework for distributed denial of service attack classification," *IEEE Access*, 2024.

[14] P. Mamatha, S. Balaji, and S. S. Anuraghav, "Development of Hybrid Intrusion Detection System Leveraging Ensemble Stacked Feature Selectors and Learning Classifiers to Mitigate the DoS Attacks," *International Journal of Computational Intelligence Systems*, vol. 18, no. 1, p. 20, 2025.

[15] X. Han, R. Zhang, X. Liu, and F. Jiang, "Biologically inspired smart contract: A blockchain-based DDoS detection system," in *2020 IEEE International Conference on Networking, Sensing and Control (ICNSC)*, 2020: IEEE, pp. 1-6.

[16] M. Beulah and B. P. Manickam, "Detection of DDoS Attack Using Ensemble Machine Learning Techniques," in *Soft Computing for Security Applications*, Singapore, G. Ranganathan, X. Fernando, F. Shi, and Y. El Allioui, Eds., 2022: Springer Singapore, pp. 889-903.

[17] O. Thorat, N. Parekh, and R. Mangrulkar, "TaxoDaCML: Taxonomy based Divide and Conquer using machine learning approach for DDoS attack classification," *International Journal of Information Management Data Insights*, vol. 1, no. 2, p. 100048, 2021.

[18] G. Usha, M. Narang, and A. Kumar, "Detection and classification of distributed DoS attacks using machine learning," in *Computer Networks and Inventive Communication Technologies: Proceedings of Third ICCNCT 2020*, 2021: Springer, pp. 985-1000.

[19] D. Akgun, S. Hizal, and U. Cavusoglu, "A new DDoS attacks intrusion detection model based on deep learning for cybersecurity," *Computers & Security*, vol. 118, p. 102748, 2022.

[20] I. D. Mienye and Y. Sun, "A survey of ensemble learning: Concepts, algorithms, applications, and prospects," *IEEE Access*, vol. 10, pp. 99129-99149, 2022.

[21] S. Chen and N. M. Luc, "RRMSE Voting Regressor: A weighting function based improvement to ensemble regression," *arXiv preprint arXiv: 2207.04837*, 2022.

[22] S. R. Quasar, R. Sharma, A. Mittal, M. Sharma, D. Agarwal, and I. de La Torre Díez, "Ensemble methods for computed tomography scan images to improve lung cancer detection and classification," *Multimedia Tools and Applications*, vol. 83, no. 17, pp. 52867-52897, 2024.

[23] R. Bakır, C. Orak, and A. Yüksel, "Optimizing hydrogen evolution prediction: A unified approach using random forests, lightGBM, and Bagging Regressor ensemble model," *International Journal of Hydrogen Energy*, vol. 67, pp. 101-110, 2024.

[24] A. H. Alsaeedi, A. M. Al-juboori, R. R. Nuiaa, Z. A. A. Alyasseri, H. J. Mohammed, N. Sani, M. I. Esa, and B. A. Musawi, "A Hybrid Cracked Tiers Detection System Based on Adaptive Correlation Features Selection and Deep Belief Neural Networks," *Symmetry*, 2023.

[25] E. Yaghoubi, E. Yaghoubi, A. Khamees, and A. H. Vakili, "A systematic review and meta-analysis of artificial neural network, machine learning, deep learning, and ensemble learning approaches in field of geotechnical engineering," *Neural Computing and Applications*, vol. 36, no. 21, pp. 12655-12699, 2024.

DOI: https://doi.org/10.54216/FPA.210109
Received: February 25, 2025 Revised: June 05, 2025 Accepted: July 10, 2025