International Journal of Computers and Informatics

Journal Homepage: https://www.ijci.zu.edu.eg

Int. j. Comp. Info. Vol. 11 (2026) 111–122

Paper Type: Original Article

Machine Learning and Deep Learning Approaches for Cyberattack Detection in IoT Environments

Alaa Hassan [1,*]

[1] Faculty of Computers and Informatics, Zagazig University, Zagazig, Egypt. Email: alaa.hassan@sha.edu.eg.

Received: 30 Jan 2026 **Revised:** 08 Apr 2026 **Accepted:** 28 Jun 2026 **Published:** 30 Jun 2026

Abstract

Intrusion detection has become a crucial field of study due to the quick expansion of Internet of Things (IoT) devices and the growing complexity of cyber threats. With an emphasis on their evaluation settings, advantages, and disadvantages, this paper examines current Machine Learning (ML) and Deep Learning (DL) techniques for cyberattack detection in Internet of Things contexts. Current research frequently relies on a small number of datasets, which limit the capacity to evaluate model generalization and robustness in a variety of contexts. This study uses five benchmark datasets UNSW_NB15, NSL_KDD, Edge-IIoTset, RT-IoT2022, and IoTID20 to compare ML and DL models in order to address this problem. Additionally, the Manta Ray Foraging Optimizer (MRFO) and other metaheuristic optimization techniques are emphasized as a means of enhancing model performance through ensemble learning and hyperparameter adjustment. Two optimized models, ET-MRFO and VC-RXE-MRFO, are analyzed as examples of enhanced detection accuracy and stability. The results indicate that optimized ML ensemble models achieve competitive performance compared to DL methods.

Keywords: Cyberattack detection, Internet of Things (IoT), Machine learning, Deep learning, Manta Ray Foraging, Optimization techniques.

1 | Introduction

1.1 | Background and Importance

The Internet of Things (IoT) has emerged as a key element of contemporary digital systems, facilitating constant communication in a variety of fields, including industrial automation, smart cities, and healthcare [1]. The rapid expansion of Internet of Things (IoT) devices has significantly increased the attack surface of modern networks, making them attractive targets for cybercriminals [2]. Among the most critical threats facing IoT environments are Distributed Denial of Service (DDoS) attacks, which can disrupt network services, degrade system performance, and compromise the availability of critical applications. Traditional security mechanisms often struggle to cope with the massive volume and complexity of IoT-generated traffic [3], [4] because IoT networks are large-scale and dynamic, traditional security measures are frequently inadequate. Intelligent data-driven strategies have therefore drawn more attention in recent years. Specifically, by recognizing patterns in network traffic data, Machine Learning (ML) and Deep Learning (DL) algorithms have demonstrated significant promise in detecting harmful network behavior [5], [6].Despite these developments, there are still a number of obstacles. Concerns about model generalization and robustness in

Corresponding Author: alaa.hassan@sha.edu.eg

real-world scenarios are raised by the fact that many current studies assess their models on a small number of datasets. Furthermore, efficient feature selection and appropriate hyperparameter tuning which are frequently disregarded or inadequately optimized are crucial to the success of ML and DL models[7], [8]. Recent studies have investigated the incorporation of Metaheuristic optimization approaches to improve model performance to overcome these issues[9] Among these methods, the Manta Ray Foraging Optimizer (MRFO) has shown encouraging outcomes in enhancing stability and classification accuracy in high-dimensional IoT datasets[10]

1.2 | Research Objectives

- Conduct a comparative study of recent ML and DL models on five widely used intrusion datasets: UNSW_NB15 (DS1), NSL_KDD (DS2), Edge-IIoTset (DS3), RT-IoT2022 (DS4), and IoTID20 (DS5).
- Propose ET-MRFO, which tunes the Extra Trees classifier via the Manta Ray Foraging Optimizer (MRFO) to reduce under/overfitting by selecting near-optimal tree depth and ensemble size.
- Propose VC-RXE-MRFO, a weighted voting ensemble over Random Forest (RF), Extra Trees (ET), and XGBoost (XGB), using MRFO to optimize both base-model hyperparameters and voting weights.
- Evaluate models including K-fold validation and analyses of class imbalance handling.

1.3 | Research Questions

- How well do ML and DL models detect cyberattacks across various IoT datasets with diverse attributes?
- How much can intrusion detection model performance be enhanced by metaheuristic optimization techniques?
- In IoT intrusion detection tasks, how successful are ensemble-based machine learning models in comparison to individual classifiers and deep learning models?
- What are the primary obstacles to model deployment, stability, and generalization in actual IoT environments?

2 | Literature Review

This section surveys existing research and technologies related to intrusion detection in IoT networks. It critically analyzes previous machine learning and deep learning approaches, their methodologies, datasets, performance, and limitations. By identifying knowledge gaps and unresolved challenges, establishes the scientific basis for the proposed work and clarifies how this thesis advances the current state of the art.

2.1 | Machine Learning Techniques for Cyberattack Detection

To overcome the limitations imposed by the hardware for IoT and WSNs, there have been attempts by various researchers to use classical machine learning algorithms along with feature engineering to minimize computational costs while keeping detection accuracy very high.

In the IoT environment, there are cases where machine learning algorithms have been adopted in developing the IDSs. What makes their solution unique is the implementation of a combined feature selection method that can filter out irrelevant and redundant network features. The use of only the most relevant features when training AI and machine learning classifiers led to substantial reduction in energy usage and processing times, proving that properly optimized classic ML solutions can still be efficient and scalable enough for edge intrusion detection [11] examined the effectiveness of using ensembles in traditional machine learning through the design of a Gradient Boosting Decision Tree (GBDT) architecture for WSN security. Since the network

traffic information is tabulated data, this research utilized the step-by-step correction feature of gradient boosting. The GBDT algorithm designed in this work demonstrated an appropriate trade-off between training efficiency and accuracy of classification, which shows that tree-based machine learning algorithms can recognize anomalies efficiently even if they do not have powerful computing facilities like deep neural networks [12].

One of the other major trends in ensuring security for CPS and IIoT is using ensemble learning frameworks together with efficient feature selection algorithms for improved detection variance and generalization.

Abdullahi et al. [13] undertook an extensive comparative study of different AI and machine learning techniques that could be used in detecting cyberattacks on CPS. Through their study, they were able to point out both the merits and demerits of different conventional classifiers that can be used as benchmarks in designing AI tools [13].

To improve the stability of classification, developed an innovative cascaded ensemble technique that involves the sequential combination of bagging and boosting schemes. This combined scheme is especially designed for intrusion detection in CPS systems where it takes advantage of the variance reduction characteristic of bagging along with the bias reduction ability of boosting, which increases its resistance to complex network attacks. Likewise, developed an ensemble learning technique based on voting that is used to protect IIoT networks. They made use of the voting scheme to combine the outputs of different classifiers to avoid single classifier failures [14], [15].

Feature preprocessing was applied, where they designed an ensemble learning model to enhance threat detection by leveraging feature selection methods. They highlighted that the integration of heterogeneous traditional ML algorithms as a meta-classifier (Stacking architecture) in conjunction with proper feature subset results in highly efficient threat detection rates [16] .

Fourthly, have made further progress in this area through developing an enhanced Intrusion Detection Framework (IDF) that uses an optimal feature selection algorithm to optimize their framework. Their approach has a strong focus on the importance of selecting the most important parameters from the network as the use of the two improves efficiency and reduces computational overhead [17]

2.2 | Deep Learning Techniques for Cyberattack Detection

Deep Learning (DL) approaches have attracted much interest in detecting cyberattacks within the Internet of Things (IoT) as they offer automatic learning of features from raw inputs. In comparison with conventional ML algorithms, the use of DL approaches decreases the need for explicit feature engineering.

Responded to the rising demand for robust networks by developing a method using hybrid deep learning approaches. Using hybrid deep neural network models, their method could detect spatial as well as temporal features in the highly dynamic IoT traffic. The hybrid approach proved to be more effective in detecting complex zero-day attacks [18].

Building upon this further within the area of explainability and privacy, [19] came up with the CPS-IIoT-P2Attention framework. In this model, architecture with the scaled dot product attention scheme is employed, which is tailored for use in CPS and IIoT systems. Through the incorporation of attention layers, the network can assign weights to the most important threat features to improve the classification performance while ensuring privacy of the data.

With reference to the concept of trustworthy AI, CPS-IoT-PPDNN was introduced. This refers to a new, explainable, and privacy preserving Deep Neural Network for detecting anomalies in CPS enabled IoT networks. This model is designed to provide strong cryptographic or perturbation-based privacy protection during the deep training stage while leveraging XAI to make sure that the deep learning network makes its decisions in a transparent manner [20].

Finally, proposed DeepAK-IoT, which was an efficient and customized deep learning approach for quick detection of cyberattacks in IoT networks. The authors made use of customized deep architecture for the optimization of the internal hidden layers and the activation functions to gain quick convergence. It was shown by the authors that custom-made deep architectures can sustain high detection throughput, and hence, are very effective [21].

2.3 | Metaheuristic Optimization in Intrusion Detection

To deal with problems such as those posed by high-dimensional search spaces, slow convergence, and unsuitable hyperparameters in security frameworks, recent literature has extensively investigated the use of nature-inspired metaheuristic optimization algorithms in IDSs.

Provided a thorough study on intelligence, security, and optimization in multi-tier Internet of Things architecture. The key aspect highlighted in their study was the need for security in the edge that cannot be accomplished using classification models only but should also include metaheuristic optimization methods[22].

In the context of core optimization algorithms, the Harris Hawks Optimization (HHO) algorithm was proposed by Heidari et al. (2019). This is a bio-inspired metaheuristic algorithm that is based on the cooperative hunting strategy employed by Harris Hawks. It is evident that the HHO possesses superior capabilities regarding exploration and exploitation and has been extensively used in cyber security to avoid local minima[23] .

Similarly, the Artificial Rabbits Optimization (ARO) algorithm, which is one of the recent metaheuristics based on biological concepts, mimicking the behavior of rabbits in terms of their hunting and hiding behaviors, has been found to be highly efficient in dealing with difficult problems of engineering and continuous optimization[24].

Considering the above ideas in relation to current network security challenges, suggested a smart hybrid model using Fuzzy C-means clustering along with the Sperm Whale Optimization technique for the purpose of detecting cyberattacks in IoT networks. In doing so, they were able to take advantage of the optimization search behavior of sperm whales for the selection of cluster centers and subsequent classification boundaries, thus maximizing attack detection accuracy[25].

Introduced the Manta Ray Foraging Optimizer (MRFO) algorithm, which is a new type of metaheuristic population-based algorithm based on the intelligent foraging behavior of manta rays. In this work, the authors have formulated three different survival mechanisms mathematically, namely chain foraging, cyclone foraging, and somersault foraging. As a result, this algorithm can dynamically shift between the extensive exploration phase and exploitation phase of search space. Based on the benchmark studies, it has been found that the Manta Ray Foraging Optimizer algorithm possesses remarkable converging speed and computational efficiency along with an excellent ability to overcome local optima[10].

Even though there have been successful implementations of metaheuristic algorithms such as HHO and ARO in the field of network security, most of the frameworks that have been used till date face the problems of high computational cost or early convergence when using high dimensional datasets generated by the internet of Things. With this problem in mind, the Manta Ray Foraging Optimizer can be considered as one of the promising solutions due to its unique features of chain, cyclone and somersault foraging movement that results in efficient balance between global and local searches. Based on these specific mathematical strengths, this thesis proposes two new optimization frameworks – namely ET-MRFO and VC-RXE-MRFO.

2.4 | search gaps, limitations, and contributions

Based on the reviewed literature, several research gaps and limitations can be identified in the field of cyberattack detection in IoT environments:

- There is limited integration of uncertainty-handling techniques within existing intrusion detection frameworks, despite the inherently ambiguous nature of IoT network data.
- Most existing studies focus on individual models rather than adopting hybrid or integrated approaches that combine multiple techniques to improve performance and robustness.
- Current models often lack the ability to simultaneously consider multiple evaluation criteria, such as detection performance, computational cost, and practical usability, which are essential for real-world deployment.
- In addition, handling incomplete, imbalanced, and high-dimensional data remains a significant challenge in complex IoT environments.
- Furthermore, advanced models and optimization techniques, although effective, are often associated with high computational cost, limiting their applicability in resource-constrained IoT systems.

Finally, many proposed models are evaluated on a limited number of datasets, which raises concerns regarding their generalization capability across diverse real-world scenarios. The main contributions of this survey are as follows:

- A comprehensive comparative analysis of twelve ML and DL models across five widely used IoT intrusion detection datasets.
- The investigation of optimization-based approaches, particularly the use of the Manta Ray Foraging Optimizer (MRFO), for improving model performance.
- The presentation and analysis of two optimized models, namely ET-MRFO and VC-RXE-MRFO, demonstrate the effectiveness of combining ensemble learning with metaheuristic optimization.
- A detailed evaluation of model performance using multiple metrics, including accuracy, precision, recall, and F1-score, to provide a balanced assessment.
- Identification of key challenges and future research directions, including computational efficiency, model robustness, and real-time deployment in IoT environments.

3 |Methodology

3.1 | Manta Ray Foraging Optimizer (MRFO)

Recently, a new optimization algorithm, known as the Manta Ray Foraging Optimizer (MRFO) and inspired by the chain, cyclone, and somersault foraging mechanisms of manta rays, has been proposed for handling engineering optimization algorithms [10]. It was evaluated and compared to several competitors on mathematical benchmarks and eight engineering problems to show its stability and efficiency. In MRFO, the chain foraging mechanism is replicated by moving all individuals except the first one towards not only the one in front of it but also the best-so-far solution, while the first individual is moved towards the best-so-far solution, as stated in the following mathematical model:

$$\vec{x}_i^{t+1} = \begin{cases} \vec{x}_i^t + \vec{r} \cdot (\vec{x}^* - \vec{x}_i^t) + \boldsymbol{\alpha} \cdot (\vec{x}^* - \vec{x}_i^t), & \boldsymbol{if\ i = 1} \\ \vec{x}_i^t + \vec{r} \cdot (\vec{x}_{i-1}^t - \vec{x}_i^t) + \boldsymbol{\alpha} \cdot (\vec{x}^* - \vec{x}_i^t), & \boldsymbol{otherwise} \end{cases} \quad (1)$$

$$\boldsymbol{\alpha} = \mathbf{2} \cdot \vec{r} \cdot \sqrt{|\log \vec{r}|} \quad (2)$$

where $\vec{r}$ is a vector including random values in $[\mathbf{0}, \mathbf{1}]$, $\vec{x}^*$ represents the best-so-far solution, $\vec{x}_i^{t+1}$ represents

The new position of the $\boldsymbol{ith}$ solution, $\cdot$ stands for element-wise multiplication operator, and $\vec{x}_i^t$ represents the current position of the $\boldsymbol{ith}$ solution. The cyclone foraging mechanism in MRFO is divided into two folds.

The first fold is responsible for exploiting the regions around the best-so-far solution for accelerating the convergence rate towards the best-so-far solution and is mathematically defined as follows:

$$\vec{x}_i^{t+1} = \begin{cases} \vec{x}^* + \vec{r} \cdot (\vec{x}^* - \vec{x}_i^t) + \boldsymbol{\beta} \cdot (\vec{x}^* - \vec{x}_i^t), & \boldsymbol{if\ i = 1} \\ \vec{x}^* + \vec{r} \cdot (\vec{x}_{i-1}^t - \vec{x}_i^t) + \boldsymbol{\beta} \cdot (\vec{x}^* - \vec{x}_i^t), & \boldsymbol{otherwise} \end{cases} \quad (3)$$

$$\boldsymbol{\beta} = \mathbf{2} \cdot \boldsymbol{e}^{r_1 \frac{T_{max}-t+1}{T_{max}}} \cdot \boldsymbol{sin}(\mathbf{2}\pi \boldsymbol{r_1}) \quad (4)$$

where $\boldsymbol{r_1}$ is a random number in $[\mathbf{0}, \mathbf{1}]$, $\boldsymbol{t}$ represents the current function evaluation, and $\boldsymbol{T_{max}}$ is the maximum function evaluation. On the other hand, the second fold is responsible for exploring the search space as much as possible to avoid stagnation in local optima, significantly improving the final results. This fold is mathematically defined as follows:

$$\vec{x}_i^{t+1} = \begin{cases} \vec{x}_a^t + \vec{r} \cdot (\vec{x}_a^t - \vec{x}_i^t) + \boldsymbol{\beta} \cdot (\vec{x}_a^t - \vec{x}_i^t), & \boldsymbol{if\ i = 1} \\ \vec{x}_a^t + \vec{r} \cdot (\vec{x}_{i-1}^t - \vec{x}_i^t) + \boldsymbol{\beta} \cdot (\vec{x}_a^t - \vec{x}_i^t), & \boldsymbol{otherwise} \end{cases} \quad (5)$$

$$\vec{x}_a^t = \vec{L} + \vec{r} \cdot (\vec{U} - \vec{L}) \quad (6)$$

where $\vec{L}$ is a vector including the lower limits of all dimensions in an optimization problem, and $\vec{U}$ includes the upper limits. According to[10], the preference for both folds during the optimization process is related to the current function evaluation, with the second fold applied at the start and the first fold applied as the optimization process progresses, as defined in the following equation:

$$\vec{x}_i^{t+1} = \begin{cases} (5), & if\ r_1 > \frac{t}{T_{max}} \\ (3), & otherwise \end{cases} \quad (7)$$

Finally, the somersault foraging mechanism always updates the current individual around the best-so-far solution to improve the exploitation operator and expedite convergence to the global optimum. The mathematical model for this mechanism is shown below:

$$\vec{x}_i^{t+1} = \vec{x}_i^t + S \cdot (r_2 \cdot \vec{x}^* - r_3 \cdot \vec{x}_i^t) \quad (8)$$

where S stands for the somersault factor and is recommended to be set to 2, and r_2 and r_3 are random values in $[0, 1]$.

3.2 | Ensemble and Boosting Base Learners

As an ensemble learning method, this classifier, namely the extra trees (ET) classifier, was proposed for tackling classification and regression problems. Like the random forest (RF) classifier, it employs a set of decision trees (DTs) to learn the characteristics of the input data, and the final classification is based on the majority voting of all DTs. Despite that, the ET classifier is different from the RF classifier in the way the decision trees are built [26, 27]. The ET classifier trains each DT on the entire dataset, whereas the RF employs bootstrapping aggregating to train each DT independently on random subsets. In addition, the ET classifier selects the split point at random, while the RF classifier selects the best split. Finally, all decision trees in the ET classifier present their classification labels, and the class with the majority of votes represents the final classification.

The boosting algorithms combine weak classifiers to create a strong learner capable of better understanding input data and achieving higher classification accuracy, according to[28] . In these algorithms, a weak learner is applied to the training dataset, and any observations misclassified by this learner are corrected by another learner, and so on until the maximum number of iterations is reached or all training datasets are correctly classified. There are several boosting algorithms in literature, such as Adaboost [29], gradient boosting [30], CatBoost [31], histogram-based gradient boosting (HGB), and Extreme gradient boosting (XGBoost). AdaBoost (Adaptive Boosting) as an ensemble learning approach combines several DTs to improve classification accuracy. First, this method gives all training samples the same amount of weight. The weights are then changed over time to give misclassified observations more weight so that the next model can focus on them. Gradient Boosting uses gradient descent to build models sequentially, with each weak learner

minimizing the residual error of the previous learner. Rather than modifying weights as AdaBoost does, Gradient Boosting optimizes a loss function to directly minimize error. Both XGBoost and HGB are improved variants of gradient boosting, which could improve computational cost and final accuracy. In brief, the main advantages of XGBoost are described as follows [32]:

- Unlike the gradient boosting, it minimizes the computational cost and improves final accuracy.
- Like the RF, it can create DTs in a parallel way.
- Can evaluate large and complex datasets using distributed computing methods.
- Employs cache optimization to get a higher degree of optimal resource use.

3.3 | Proposed Models

3.3.1 | ET classifier tuned by MRFO

The ET and RF classifiers have various hyperparameters, such as max_depth and the number of DTs (n_estimators), which can significantly affect overall classification accuracy. Therefore, they must be accurately tuned to enhance their performance in identifying network cyberattacks. This motivates us to propose a new cyberattack detection method, called ET-MRFO, which combines the ET classifier with the MRFO algorithm to better optimize those hyperparameters and achieve more precise classification results. In brief, the main steps of ET-MRFO are outlined as follows:

- **Initialization step:** The MRFO starts by creating a population of N solutions, each with d dimensions, which are then randomly initialized within the lower and upper bounds of various parameters. In this study, we use this algorithm to estimate both n_estimators and max_depth; therefore, each solution contains two dimensions and is randomly initialized according to eq (1). These initial solutions, one at a time, are used to build a new ET classifier, which is trained using the training dataset before being tested on the testing dataset. The objective function that the MRFO algorithm seeks to optimize to find the near-optimal hyperparameters measures the error rate between the classification accuracy achieved with the current hyperparameters and the maximum possible accuracy of 1, as shown in eq (2). This function must be minimized to improve overall classification accuracy. After evaluating all solutions, they are compared, and the one with the lowest error is considered the best so far.

$$\vec{x}_i^t = \vec{L} + \vec{r} \cdot (\vec{U} - \vec{L}), \forall i \in 1{:}N \quad (1)$$

$$f(\vec{x}_i^t) = 1 - \alpha(\vec{x}_i^t) \quad (2)$$

where 1 represents the maximum classification accuracy; $\alpha(\vec{x}_i^t)$ indicates the classification accuracy achieved with the hyperparameters in the ith solution $(\vec{x}_i^t)$; $\vec{L}$ represents the lower bounds for max_depth and n_estimators, set at 10 and 2, respectively; and $\vec{U}$ indicates the upper bounds, set at 500 and 2000. It's worth noting that using different values for these bounds could yield better results; in this study, we chose them heuristically.

- **Optimization process:** The MRFO begins its search to update current solutions and find new ones that can boost classification accuracy. It uses chain, cyclone, and somersault foraging mechanisms, which have been extensively presented before, to update these solutions. In each iteration, new solutions are evaluated as described earlier, and the best-so-far solution is updated with the new one that can achieve higher classification accuracy. This process repeats until the maximum allowed number of function evaluations is reached.

- **Create the final ET classifier**: The best-so-far solution returned after completing the optimization process is used to create the final ET classifier, known as ET-MRFO.

- **Training and testing process:** Using K-fold cross-validation, the dataset is divided into 3 folds. Then, the ET classifier is trained independently 3 times. Each time, it is trained with 2 distinct folds, and the third fold is used to test its generalization.

3.3.2 | Weighted voting-based classifier tuned by MRFO

Furthermore, this survey suggests a different approach for detecting cyberattacks by combining a novel weighted voting classifier with MRFO. The weighted voting classifier is based on integrating three high-performing ML models—RF, XGBoost (XGB), and ET—to leverage their strengths and build a more accurate classifier for cyberattack detection. This classifier assigns a weight to each model to determine how much it influences the final decision. For example, suppose there are three classifiers, ET, RF, and XGB, with weights of 0.2 for ET, 0.7 for RF, and 0.1 for XGB, reflecting each model's contribution to final accuracy. When applied to a two-class problem, A and B, the models generate predicted probabilities for each class. After training, for each test sample, ET might produce a probability of 0.6 for class A and 0.3 for class B; RF might produce 0.1 for class A and 0.9 for class B; and XGB might produce 0.7 for class A and 0.3 for class B. The final classification label is determined by multiplying each model's predicted probabilities by its weight and summing these values for each class. The multiplied probabilities for Class A are summed to decide the final class label, as defined below:

$$\textbf{Class A}: 0.2 \times 0.6 + 0.7 \times 0.1 + 0.1 \times 0.7 = 0.26$$

Likewise, the classification probability to class B is computed, as defined in the following formula:

$$\textbf{Class B}: 0.2 \times 0.3 + 0.7 \times 0.9 + 0.1 \times 0.3 = 0.72$$

The classification probabilities for different classes show that Class B is the best match for the sample because it has the highest probability. However, changing the weights assigned to the models can influence the classification results; for example, if the weights are adjusted to 0.2, 0.2, and 0.6, respectively, the probabilities to the various classes are updated to:

$$\textbf{Class A}: 0.2 \times 0.6 + 0.2 \times 0.1 + 0.6 \times 0.7 = 0.56$$

$$\textbf{Class B}: 0.2 \times 0.3 + 0.2 \times 0.9 + 0.6 \times 0.3 = 0.42$$

The updated classification probabilities indicate that Class A is the most probable label, making it the best choice. This confirms that the weights assigned to the ET, RF, and XGB models in the voting classifier (VC) influence the final label. Therefore, these weights must be carefully optimized to achieve near-optimal values that can enhance classification accuracy. To address this, we treat the problem as an optimization task and use the MRFO method to find near-optimal weights for all three models, aiming to enhance network intrusion detection accuracy. The main steps of VC-RXE-MRFO are summarized as follows:

- **Initialization step:** The MRFO begins by creating a population of N solutions, each with d dimensions. These solutions are randomly assigned within their lower and upper bounds. In VC-RXE-MRFO, this process involves estimating parameters such as n_estimators and max_depth for both RF and ET, along with the weights for ET, RF, and XGB in the VC model. Consequently, each solution has seven dimensions and is initialized according to eq (1). The lower and upper bounds for various dimensions were defined as [10, 500] for max_depth of both RF and ET, [2, 2000] for n_estimators of both RF and ET, and [0.01, 2] for the three weights of ET, RF, and XGB. It is important to note that choosing different values for these bounds could produce better results; however, in this study, we heuristically selected them. The initialized solutions are individually used to build a new VC model, which is trained on the training data and tested on the testing data. The performance of each solution is evaluated using eq (2), and the solution with the lowest error is regarded as the best-so-far solution, $\vec{x}^*$.

- **Optimization process**: The optimization process of the MRFO begins by updating existing solutions and seeking new ones that could improve classification accuracy. The new solutions in each iteration are evaluated as described in the previous phase, and the best-so-far solution is replaced by a new one with higher classification accuracy. This process continues until T_{max} is achieved.
- **Create the final weighted VC model:** The final weighted VC model, called VC-RXE-MRFO, is constructed using $\vec{x}^*$ returned at the end of the optimization phase.
- **Training and testing process:** Like the ET-MRFO model, K-fold cross-validation is employed to split the dataset into 3 folds. Then, the weighted VC model is trained independently 3 times, each time using 2 folds for training and the remaining fold for testing

4 | Results and Discussion

4.1 | Datasets and Experimental Setup

All the models were compared against each other on five popular IoT network intrusion datasets. UNSW_NB15 consists of 257,673 records that belong to nine categories of attacks and have 42 features. The second dataset is NSL_KDD consisting of 148,515 samples across 23 types of attacks with 40 features. Edge-IIoTset has around 1,900,000 records belonging to 15 types of attacks with 13 features. IoTID20 dataset consists of 625,783 records having 85 features and belonging to five types of attacks. Lastly, RT-IoT2022 has 123,117 samples, which belong to 12 types of attacks with 84 features.
These datasets were used because of their popularity in literature, and different sizes, numbers of features, as well as variety in types of attacks. The preprocessing process included removing uninformative columns, removing duplicated records, one-hot encoding of categorical variables, and standardization of numerical variables. Due to class imbalance, which is a common problem with these types of datasets, four different methods of handling this problem were used: NearMiss, SMOTE, SMOTETomek, and SMOTEENN. SMOTEENN performed the best and thus was used for all further analysis.

4.2 | Comparative Performance Analysis

However, the results obtained in all five datasets indicated that ML algorithms perform better than DL algorithms in identifying cyberattacks within IoT environment. In the case of UNSW_NB15 dataset, the best accuracy obtained is for VC-RXE-MRFO with 0.995780, followed by ET-MRFO which has the accuracy rate of 0.995217 while the worst performing algorithm among the three was LSTM with an accuracy of 0.935119. The second dataset, NSL_KDD dataset, had the best accuracy rate for VC-RXE-MRFO, which has an accuracy of 0.999936 followed by ET-MRFO with an accuracy rate of 0.999880. On Edge-IIoTset dataset, VC-RXE-MRFO is once again the best performing model with an accuracy rate of 0.993838 and followed by ET-MRFO with an accuracy rate of 0.993723. The two last datasets RT-IOT2022 and IOTID20 the best accuracy rate for using MRFO optimizer SC-REXL-MRFO 0.999047 and 0.999604 followed by VC-RXE-MRFO with an accuracy 0.998834 and 0.999526. Overall, our experiments confirm that ML models significantly outperformed DL models in detection cyberattacks in IOT environments.

4.3 | ROC-AUC Analysis

An evaluation by the ROC-AUC method on UNSW_NB15 and Edge-IIoTset datasets revealed the better discriminatory capability of the proposed model. VC-RXE-MRFO attained an AUC score of 1.0 in eight out of ten classes for UNSW_NB15 and in eleven out of fifteen classes for Edge-IIoTset, with the rest of the classes scoring more than 0.9995. The results of ET-MRFO were similarly good, where the minimum ROC-AUC score was more than 0.9946 in all classes.

4.4 | Statistical Significance and Stability Analysis

The use of statistical verification with the Wilcoxon signed-rank test showed that the improvement obtained from VC-RXE-MRFO compared with all considered ML models is statistically significant (p-value < 0.05) except for ET-MRFO in some performance indicators. The stability test of models with respect to 15 repetitions revealed that ET-MRFO was the most stable method with the standard deviation of 0.000678, then came ET with 0.000682 and VC-RXE-MRFO with 0.000736.

The computational cost of models about Edge-IIoTset dataset revealed that VC-RXE-MRFO needed approximately11 seconds for training, which takes around 3 seconds longer than ET-MRFO because three base classifiers were used at once. However, VC-RXE-MRFO outperformed ET-MRFO in terms of inference time since the combination of various models decreased the number of estimators and their depth.

5 | Conclusion

This survey conducted a thorough assessment of ML and DL approaches towards detecting cyberattacks in IoT environments. This work discussed the pros and cons of different methods used and paid particular attention to ensemble learning algorithms in machine learning and optimization approaches. The results showed that an ensemble learning approach, particularly using ML algorithms in combination with metaheuristics, like the Manta Ray Foraging Optimizer (MRFO), could provide highly accurate predictions and better stability of the models on different data samples. On the other hand, despite the high capacity of deep learning models in recognizing complex patterns, applying these approaches can be difficult due to expensive computing costs and the need for large datasets. Moreover, this paper also pointed out the necessity of assessing models on multiple data sets to ensure their generalizability and robustness within the IoT environment. However, there are still some challenges associated with computational efficiency, high dimensionality, and imbalance in the data, as well as real-time implementation that hamper the performance of current IDSs. In future research, the focus will be on employing other types of metaheuristic algorithms, including the marine predator algorithm and spider wasp optimizer. Moreover, lightweight ensemble learning models will be developed to minimize computational cost, while hybrid deep learning (DL) and machine learning (ML) models with convolutional neural networks for feature extraction and the suggested ensemble classifiers will be studied.

Funding

This research was not supported by any funding agency or institute.

Data Availability

All available data are present in the manuscript. All data and models generated or used during the study appear in the sub-mitted article.

Conflicts of Interest

The authors declare no conflict of interest.

References

[1] S. Aktar and A. Yasin Nur, "Towards DDoS attack detection using deep learning approach," *Comput. Secur.*, vol. 129, p. 103251, 2023, doi: https://doi.org/10.1016/j.cose.2023.103251.

[2] A. A. Diro and N. Chilamkurti, "Distributed attack detection scheme using deep learning approach for Internet of Things," *Future Generation Computer Systems*, vol. 82, pp. 761–768, May 2018, doi: 10.1016/j.future.2017.08.043.

[3] R. Manivannan and S. Senthilkumar, "Intrusion Detection System for Network Security Using Novel Adaptive Recurrent Neural Network-Based Fox Optimizer Concept," *International Journal of Computational Intelligence Systems*, vol. 18, no. 1, Dec. 2025, doi: 10.1007/s44196-025-00767-x.

[4] R. Aboalela *et al.*, "Harnessing feature pruning with optimal deep learning-based distributed denial of service cyberattack detection on IoT environment," *Alexandria Engineering Journal*, vol. 120, pp. 584–597, May 2025, doi: 10.1016/j.aej.2025.02.070.

[5] F. R. A. M. K. I. A. S. M. M. ALANOUD AL MAZROA, "Boosting Cyberattack Detection Using Binary Metaheuristics with Deep Learning on Cyber- Physical System Environment," *IEEE Access*, 2025.

[6] S. A. Bakhsh, M. A. Khan, F. Ahmed, M. S. Alshehri, H. Ali, and J. Ahmad, "Enhancing IoT network security through deep learning-powered Intrusion Detection System," *Internet of Things (Netherlands)*, vol. 24, Dec. 2023, doi: 10.1016/j.iot.2023.100936.

[7] T. Vaiyapuri *et al.*, "Automated cyberattack detection using optimal ensemble deep learning model," *Transactions on Emerging Telecommunications Technologies*, vol. 35, no. 4, p. e4899, Apr. 2024, doi: https://doi.org/10.1002/ett.4899.

[8] O. A. Alzubi, I. Qiqieh, and J. A. Alzubi, "Fusion of deep learning based cyberattack detection and classification model for intelligent systems," *Cluster Comput.*, vol. 26, no. 2, pp. 1363–1374, 2023, doi: 10.1007/s10586-022-03686-0.

[9] M. Maazalahi and S. Hosseini, "Machine learning and metaheuristic optimization algorithms for feature selection and botnet attack detection," *Knowl. Inf. Syst.*, vol. 67, no. 4, pp. 3549–3597, Jan. 2025, doi: 10.1007/s10115-024-02322-0.

[10] W. Zhao, Z. Zhang, and L. Wang, "Manta ray foraging optimization: An effective bio-inspired optimizer for engineering applications," *Eng. Appl. Artif. Intell.*, vol. 87, p. 103300, 2020, doi: https://doi.org/10.1016/j.engappai.2019.103300.

[11] M. H. Behiry and M. Aly, "Cyberattack detection in wireless sensor networks using a hybrid feature reduction technique with AI and machine learning methods," *J. Big Data*, vol. 11, no. 1, Dec. 2024, doi: 10.1186/s40537-023-00870-w.

[12] R. Mohandas, P. A, V. e S, A. Prabhakaran a, S. Gupta, and N. Malik, *Enhanced Cyberattack Detection in Wireless Sensor Networks Using Gradient Boosting Decision Tree*. International Conference on Mobile Networks and Wireless Communications (ICMNWC), 2024. doi: 10.1109/ICMNWC63764.2024.10872135.

[13] M. Abdullahi *et al.*, "Comparison and Investigation of AI-Based Approaches for Cyberattack Detection in Cyber-Physical Systems," *IEEE Access*, vol. PP, p. 1, Jan. 2024, doi: 10.1109/ACCESS.2024.3370436.

[14] R. Ji, A. Selwal, N. Kumar, and D. Padha, "Cascading Bagging and Boosting Ensemble Methods for Intrusion Detection in Cyber-Physical Systems," *SECURITY AND PRIVACY*, vol. 8, no. 1, p. e497, Jan. 2025, doi: https://doi.org/10.1002/spy2.497.

[15] R. Golchha, A. Joshi, and G. P. Gupta, "Voting-based Ensemble Learning approach for Cyber Attacks Detection in Industrial Internet of Things," *Procedia Comput. Sci.*, vol. 218, pp. 1752–1759, 2023, doi: https://doi.org/10.1016/j.procs.2023.01.153.

[16] W. F. Urmi *et al.*, "A stacked ensemble approach to detect cyber attacks based on feature selection techniques," *International Journal of Cognitive Computing in Engineering*, vol. 5, pp. 316–331, Jan. 2024, doi: 10.1016/j.ijcce.2024.07.005.

[17] R. Ji, N. Kumar, and D. Padha, "Hybrid Enhanced Intrusion Detection Frameworks for Cyber-Physical Systems via Optimal Features Selection," *Indian J. Sci. Technol.*, vol. 17, no. 30, pp. 3069–3079, Aug. 2024, doi: 10.17485/IJST/v17i30.1794.

[18] M. Maaz, G. Ahmed, A. Al-Shamayleh, A. Akhunzada, S. Akhtar, and A. Al-Ghushami, "Empowering IoT Resilience: Hybrid Deep Learning Techniques for Enhanced Security," *IEEE Access*, vol. PP, p. 1, Jan. 2024, doi: 10.1109/ACCESS.2024.3482005.

[19] Y. Saheed and J. Chukwuere, "CPS-IIoT-P2Attention: Explainable Privacy-Preserving with Scaled Dot-Product Attention in Cyber Physical System-Industrial IoT Network," *IEEE Access*, vol. PP, p. 1, Jan. 2025, doi: 10.1109/ACCESS.2025.3566980.

[20] Y. K. Saheed and S. Misra, "CPS-IoT-PPDNN: A new explainable privacy preserving DNN for resilient anomaly detection in Cyber-Physical Systems-enabled IoT networks," *Chaos Solitons Fractals*, vol. 191, p. 115939, 2025, doi: https://doi.org/10.1016/j.chaos.2024.115939.

[21] W. Ding, M. Abdel-Basset, and R. Mohamed, "DeepAK-IoT: An effective deep learning model for cyberattack detection in IoT networks," *Inf. Sci.*, vol. 634, no. C, pp. 157–171, Jul. 2023, doi: 10.1016/j.ins.2023.03.052.

[22] F. Al-Turjman and J. P. Lemayian, "Intelligence, security, and optimization in multi-tier IoT networks: A survey," *Computers and Electrical Engineering*, vol. 87, 2020.

[23] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: Algorithm and applications," *Future Generation Computer Systems*, vol. 97, pp. 849–872, 2019, doi: https://doi.org/10.1016/j.future.2019.02.028.

[24] L. Wang, Q. Cao, Z. Zhang, S. Mirjalili, and W. Zhao, "Artificial rabbits optimization: A new bio-inspired meta-heuristic algorithm for solving engineering optimization problems," *Eng. Appl. Artif. Intell.*, vol. 114, p. 105082, 2022, doi: https://doi.org/10.1016/j.engappai.2022.105082.

[25] E. I. Elsedimy and S. M. M. AboHashish, "An intelligent hybrid approach combining fuzzy C-means and the sperm whale algorithm for cyber attack detection in IoT networks," *Sci. Rep.*, vol. 15, no. 1, p. 1005, 2025, doi: 10.1038/s41598-024-79230-4.

[26] W. Zhao, Z. Zhang, and L. Wang, "Manta ray foraging optimization: An effective bio-inspired optimizer for engineering applications," *Eng. Appl. Artif. Intell.*, vol. 87, p. 103300, Jan. 2020, doi: 10.1016/j.engappai.2019.103300.

[27] M. W. Ahmad, J. Reynolds, and Y. Rezgui, "Predictive modelling for solar thermal energy systems: A comparison of support vector regression, random forest, extra trees and regression trees," *J. Clean. Prod.*, vol. 203, pp. 810–821, 2018, doi: https://doi.org/10.1016/j.jclepro.2018.08.207.

[28] S. Jafari and Y.-C. Byun, "Efficient state of charge estimation in electric vehicles batteries based on the extra tree regressor: A data-driven approach," *Heliyon*, vol. 10, no. 4, p. e25949, 2024, doi: https://doi.org/10.1016/j.heliyon.2024.e25949.

[29] Andreas Lianos and Yanyan Yang, "Classifying Unstructured Text Using Structured Training Instances and an Ensemble of Classifiers," *Journal of Intelligent Learning Systems and Applications,* vol. Vol.7 No.2, May 2015.

[30] R. Schapire, "Explaining AdaBoost," 2013, pp. 37–52. doi: 10.1007/978-3-642-41136-6_5.

[31] A. Natekin and A. Knoll, "Gradient boosting machines, a tutorial," *Front. Neurorobot.*, vol. Volume 7-2013, 2013, doi: 10.3389/fnbot.2013.00021.

[32] J. T. Hancock and T. M. Khoshgoftaar, "CatBoost for big data: an interdisciplinary review," *J. Big Data*, vol. 7, no. 1, p. 94, 2020, doi: 10.1186/s40537-020-00369-8.

[33] A. Dezhkam and M. T. Manzuri, "Forecasting stock market for an efficient portfolio by combining XGBoost and Hilbert–Huang transform," *Eng. Appl. Artif. Intell.*, vol. 118, p. 105626, 2023, doi: https://doi.org/10.1016/j.engappai.2022.105626.