International Journal of Electrical and Computer Engineering (IJECE)
Vol. 15, No. 3, June 2025, pp. 3494~3505
ISSN: 2088-8708, DOI: 10.11591/ijece.v15i3.pp3494-3505

Butterfly optimization-based ensemble learning strategy for advanced intrusion detection in internet of things networks

Mouad Choukhairi, Sara Tahiri, Ouail Choukhairi, Youssef Fakhri, Mohamed Amnai
LARI, Department of Computer Science, Faculty of Sciences, Ibn Tofail University, Kenitra, Morocco

Article Info

Article history:

Received Aug 20, 2024
Revised Mar 3, 2025
Accepted Mar 20, 2025

Keywords:

Butterfly optimization algorithm
Cybersecurity
Ensemble learning
Gradient-boosting
Internet of things
Intrusion detection system
Machine learning

ABSTRACT

The massive growth in internet of things (IoT) devices has led to enhanced functionalities through their interconnections with other devices, smart infrastructures, and networks. However, increased connectivity also increases the risk of cyberattacks. To protect IoT systems from these threats, intrusion detection systems (IDS) employing machine learning (ML) techniques have been developed to identify cybersecurity threats. This paper introduces a novel ensemble IDS framework called butterfly optimization-based ensemble learning (BOEL). This framework integrates the butterfly optimization algorithm (BOA) with ensemble learning techniques to improve IDS detection performance in IoT networks. BOEL is designed to accurately detect various types of attacks in IoT networks by dynamically optimizing the weights of base learners, which are the four sophisticated ML gradient-boosting algorithms (GBM, CatBoost, XGBoost, and LightGBM) for each attack category, and identifying the best weight combination for ensemble models. Experiments conducted on two public IoT security datasets, CICIDS2017 and Bot-IoT, demonstrate the robustness of the proposed BOEL in intrusion detection across diverse IoT environments, achieving 99.795% accuracy on CICIDS2017 and 99.966% accuracy on Bot-IoT. These results outline the successful application of diverse learning approaches and highlight the framework's potential to enhance IDS in addressing IoT cyber threats.

Corresponding Author:

Mouad Choukhairi
Department of Computer Science, Faculty of Sciences, Ibn Tofail University
B.P 133, University Campus, Kenitra, Morocco
Email: mouad.choukhairi@uit.ac.ma

1. INTRODUCTION

The exponential growth of internet of things (IoT) devices and networks has ushered in a new era of connectivity, but it has also led to an alarming increase in cyber threats and sophisticated attacks [1]. IoT networks present unique challenges for security implementations due to their heterogeneous nature, resource constraints, and large-scale deployments, which collectively create a vast attack surface for malicious actors to exploit [2]. Traditional approaches to intrusion detection systems (IDS) designed for conventional computer networks are often insufficient to meet the distinct challenges posed by IoT ecosystems [3]. These challenges include the need for accurate detection, handling of diverse data types, and adaptation to rapidly evolving threat landscapes while operating within the constraints of limited computational resources typical of IoT devices [4]. Consequently, advanced, adaptive, and efficient IDS are crucial security precautions for this increase in cyber threats in IoT environments [5]. Extensive research has focused on enhancing intrusion detection capabilities in IoT networks, with machine learning (ML) approaches gaining significant traction. These techniques leverage their ability to analyze large datasets and identify patterns, significantly improving

IoT network defenses. Thus, they improve IDS by reducing false positives and increasing accuracy. Churcher *et al.* [6] conducted a comparison of seven machine learning algorithms for intrusion detection in IoT networks using the *Bot-IoT* dataset. They found that random forest (RF) performed best in binary classification, while k-nearest neighbors (KNN) excelled in multi-class classification with 99% accuracy. Choukhairi *et al.* [7] proposed a tree-structured ML-based IDS for enhanced security in IoT networks, effectively detecting diverse cyberattacks with high accuracy and low computational costs on the UNSW-NB15 dataset. Zhang *et al.* [8] developed a two-stage intrusion detection model for IoT networks, using a light gradient-boosting machine (LightGBM) for initial traffic classification and a convolutional neural network (CNN) for detailed attack identification. Their model, tested on the *CSE-CIC-IDS2018* dataset, showed superior performance in handling imbalanced, large-scale network data compared to existing systems. Zhao *et al.* [9] developed a lightweight network intrusion detection method for IoT using principal component analysis (PCA) for dimensionality reduction and a custom neural network architecture. Their approach achieved high classification performance with low computational complexity, enabling it to be used on resource-limited IoT devices. Shitharth *et al.* [10] developed a novel clustering-based classification method for network IDS using NSL-KDD, CICIDS2017, and Bot-IoT datasets. They combined anticipated distance-based clustering (ADC) with density-based spatial clustering of applications with noise (DBScan) for data grouping, optimized parameters using perpetual pigeon galvanized optimization (PPGO), and employed likelihood naïve Bayes (LNB) for final classification. Their ADC-DBScan-LNB model outperformed other techniques in performance evaluations. However, ensemble learning methods, which combine multiple classifiers to leverage their collective strengths, have shown superior performance in handling complex and diverse data types and attack patterns often encountered in IoT environments [11], [12]. Additionally, recent studies on ensemble learning methods for IDS in IoT networks have focused on enhancing anomaly detection capabilities. By utilizing ensemble techniques like extreme gradient-boosting (XGBoost), LightGBM, and super learner, these studies aim to improve the accuracy and efficiency of anomaly detection [12]–[14]. Soni *et al.* [13] employed ensemble learning techniques, particularly XGBoost and LightGBM, to improve binary classification in IDS for IoT networks. These methods enhance anomaly detection accuracy and improve the distribution of detection capabilities across IoT devices. Balega *et al.* [14] indicated that XGBoost is a superior model for anomaly detection in IoT networks. It outperformed support vector machine (SVM) and deep convolutional neural networks (DCNN), achieving up to 99.98% accuracy. Additionally, XGBoost demonstrated significantly faster training times, being 717.75 times quicker than SVM.

Despite these crucial advances, existing approaches still face notable challenges and encounter important limitations, including high computational requirements, difficulty in detecting zero-day attacks, limited adaptability and scalability to evolving threats, and complexity in managing heterogeneous IoT data. These obstacles impede the development of robust and scalable security solutions. Addressing these gaps requires focused improvements in two key areas: i) refining detection performance to effectively handle diverse IoT situations and emerging threats, including zero-day attacks, and ii) enhancing scalability and resource efficiency to meet computing demands and adapt to the rapid expansion of the IoT landscape.

To address these issues and overcome the security limitations of IoT networks, this paper proposes a novel butterfly optimization-based ensemble learning (BOEL) framework for IDS, inspired by the natural foraging behavior of butterflies, specifically their use of sensory modalities and fragrances to locate food sources and communicate with each other. The BOEL approach leverages the butterfly optimization algorithm (BOA) to dynamically optimize the selection and weighting of base learners in the ensemble, and to achieve optimal efficiency in identifying various attack types by finding the most effective weight combinations for ensemble models, which are four sophisticated gradient-boosting ML techniques, such as XGBoost, categorical boosting (CatBoost), LightGBM, and gradient-boosting machines (GBM) for each attack type or specific class, thereby enhancing the ensemble's adaptability, performance, and computational efficiency. This paper primarily delivers the following contributions: i) a novel ensemble learning framework, termed BOEL, designed for effective intrusion detection in IoT environments by leveraging the BOA for dynamic weight optimization of base learners alongside gradient-boosting ML approaches; ii) the proposed BOEL framework is rigorously evaluated using two widely recognized public real-world IoT security datasets, Bot-IoT and CICIDS2017; and iii) the performance of BOEL is benchmarked against state-of-the-art IDS techniques.

The rest of the paper is structured as follows: section 2 provides the chosen methodology and the proposed BOEL framework design in detail. Section 3 addresses the experimental results, discussion, analysis, and evaluation method. Finally, section 4 outlines and concludes the study.

2. PROPOSED METHOD

This section describes the development of an advanced BOEL IDS framework that integrates several ML models and optimizes their combination using BOA, ensuring that the system is both effective and efficient in identifying various cyber threats in IoT networks, which implies a systematic approach to overcoming various cybersecurity challenges by boosting detection capabilities in IoT networks. As illustrated in Figure 1, the system operates in three phases: data preprocessing phase, training phase, and testing/prediction phase. It begins with comprehensive data collection from established IoT security datasets, and then, via a series of successive steps, we obtain the final predicted class to perform attack-based detection.

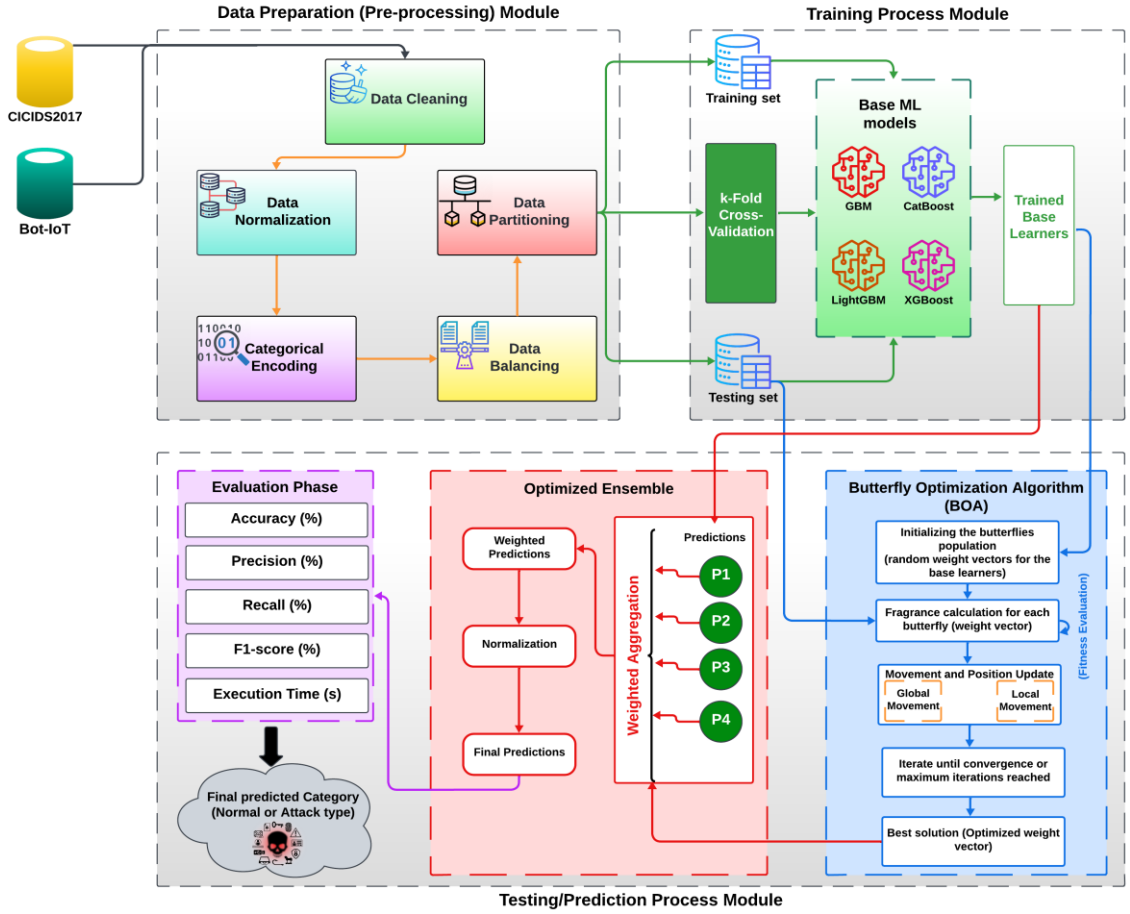

Figure 1. The framework of the proposed BOEL-based IDS

2.1. Datasets collection

The CICIDS2017 dataset [15], developed by the Canadian Institute for Cybersecurity, is a comprehensive dataset widely used for evaluating intrusion detection systems in IoT environments. It contains approximately 2.8 million network flows captured over five days, representing both benign traffic and 6 types of modern attacks (i.e., denial of service (DoS), brute force, botnet, web attacks, infiltration attacks, and sniffing) relevant to IoT networks. It meets 11 crucial benchmarks for IDS datasets and encompasses over 225,745 packages, in which each flow is described by more than 80 features. The dataset's popularity stems from its realistic nature, proportionate representation of normal and attack traffic, and inclusion of diverse attack scenarios. These characteristics make CICIDS2017 highly suitable for IoT-based research and an ideal choice for developing and evaluating ML-based IDS for IoT environments.

The Bot-IoT dataset, created at UNSW Canberra's Cyber Range Lab, is a significant resource specifically designed to address the challenges of intrusion detection in IoT environments [16]. The dataset is available in two formats: a comprehensive version with over 72 million records and a condensed 5% sample containing approximately 3 million entries of both normal and malicious traffic captured from a realistic IoT

network setup. The dataset includes various IoT-specific attacks such as distributed denial of service (DDoS), OS and service scan, DoS, keylogging, and data exfiltration, making it highly relevant for IoT security research. The Bot-IoT dataset's popularity in IoT security experiments lies in its comprehensive representation of IoT-specific threats, high-quality ground truth labels, and the inclusion of both IoT and industrial internet of things (IIoT) traffic. It addresses the limitations of previous datasets by incorporating a diverse range of IoT protocols and devices, making it more representative of real-world IoT ecosystems.

2.2. Data preparation (preprocessing)

The data preprocessing stage is dedicated to preparing traffic data from the CICIDS2017 and Bot-IoT datasets. It transforms raw data into a structured table of predefined features optimized for input into the ML models in the training process stage. This component operates through a series of consecutive steps, each designed to refine and format the data for optimal analysis and model performance, which include the following processes:

2.2.1. Data cleaning

In our study, we implemented a comprehensive data-cleaning procedure to eliminate inconsistencies and inaccuracies. This process involved a thorough examination of the data to gain deeper insights and rectify any misinterpretations. For both CICIDS2017 and Bot-IoT datasets, missing values were handled by imputing with median values. The median imputation was selected for its resilience against outlier effects compared to mean-based methods. Additionally, duplicate and corrupted records, which could skew the analysis, were identified by filtering techniques for missing value columns and duplicate rows and were removed to ensure the uniqueness of each data point. Finally, we rectified attribute labels, a common issue in comma-separated values (CSV) formatted data. To preserve data integrity, we established a standardized data representation, ensuring each attribute maintained a singular, unambiguous value per entry. This comprehensive refinement process was instrumental in creating a more accurate and reliable dataset, essential for developing a robust IDS.

2.2.2. Data normalization

Data normalization is essential to ensure that the features in the dataset are on a similar scale, which helps improve the performance of many machine learning algorithms. We employed z-score normalization, also known as standardization, which transforms the data to have a mean of 0 and a standard deviation of 1, to ensure all features are on a similar scale. This technique is particularly effective in handling datasets with attributes that have vastly different ranges, ensuring that each feature contributes equally to the model's learning process. For instance, in the context of rainfall data classification, z-score normalization has been shown to improve accuracy, sensitivity, and specificity rates, outperforming unnormalized data and other normalization techniques like min-max normalization [17]. The formula for z-score normalization is:

$$Z = \frac{(X-\mu)}{\sigma} \quad (1)$$

where X represents the original data point, μ denotes the mean of the feature, and σ signifies the standard deviation of the feature.

2.2.3. Categorical encoding

For categorical features, we employed label encoding, which is a straightforward technique for converting categorical variables into numerical values by assigning a unique integer to each category to transform categorical features present in our datasets into numerical representations. This method was chosen due to its simplicity and effectiveness, particularly for ordinal features where the order of categories is meaningful. We ensured that the encoding method aligned well with the model requirements.

2.2.4. Data balancing

To address the issue of class imbalance in IoT datasets, we applied the synthetic minority oversampling technique (SMOTE) [18]. This technique generates synthetic samples for the minority classes by interpolating between existing minority samples, thereby balancing the dataset. By applying SMOTE, we balanced the datasets more effectively, ensuring that the IDS model could learn equally from all classes and thus perform better in detecting rare attack types.

2.2.5. Data partitioning

The data partitioning phase plays a critical role in ML workflow. Each pre-processed data was randomly partitioned into training and testing sets using an 80/20 split, which is a common practice in ML

for model evaluation. In addition, a 5-fold cross-validation is integrated to evaluate the performance and generalizability of a model by partitioning the dataset into five subsets, or folds. Each fold is used as a test set once, while the remaining four folds are used for training, ensuring that every data point is used for both training and validation. This method helps mitigate issues related to overfitting and provides a more reliable estimate of model performance than a single train-test split.

2.3. Base machine learning models and training process

Decision trees (DT) are a supervised ML technique commonly applied to both classification and regression problems, renowned for their simplicity and interpretability. DTs function by iteratively splitting the dataset into smaller subgroups based on the values of the input features. This recursive partitioning constructs a hierarchical, tree-like model of decision rules. Each internal node in the tree structure represents a ”test” conducted on an attribute, the branches emanating from the node correspond to the outcomes of these tests, and the leaf nodes denote the final class labels or a continuous value (in the case of regression) assigned to the data instances [19]. DTs are fundamental components of gradient-boosting decision trees (GBDT), a powerful ensemble learning technique that combines the predictions of multiple DTs to improve accuracy and robustness. GBDT constructs a model in a stage-wise fashion by sequentially adding DTs, where each new tree corrects the errors made by the previous ones. This is achieved by fitting the new tree to the residual errors of the combined ensemble of trees, effectively using gradient descent to minimize the loss function [20]. GBDTs have evolved significantly, resulting in several state-of-the-art algorithms that are widely used in ML. The primary types of gradient-boosting algorithms include the original GBM, XGBoost, LightGBM, and CatBoost.

GBM is the foundational algorithm that introduced the concept of boosting weak learners to form a strong predictive model [21]. It improves accuracy by sequentially combining DTs, where each subsequent tree focuses on correcting the errors of the previous one. GBM’s utility extends to natural disaster prediction, such as landslide detection, where it demonstrates high predictive precision and efficiency in processing large datasets, outperforming other models like LightGBM in certain scenarios [22].

XGBoost is a scalable and efficient GBDT that has become a popular choice for its reliability and performance in ML competitions [23]. It optimizes both the loss function and a regularization term, incorporating L1 and L2 regularization to prevent overfitting. XGBoost features built-in handling of missing values, tree pruning based on *’max_depth’* and loss reduction, feature importance scoring, and support for parallel and distributed computing. Its computational complexity is low and it is $O(d||x||Tlog(n))$, where d is the maximum tree height, x is the number of non-zero samples, T is the number of trees, and n is the data length. XGBoost can be executed serially on a single thread, in parallel using multi-threading on a single machine or distributed across multiple machines using frameworks like Spark. These characteristics, along with its speed and performance, make XGBoost particularly effective for structured data problems.

LightGBM is an efficient and fast gradient-boosting framework that uses tree-based learning algorithms [24]. It employs a novel technique called gradient-based one-side sampling (GOSS) to filter out data instances with small gradients and exclusive feature bundling (EFB) to reduce the number of features. These strategies allow LightGBM to achieve faster training speed and higher efficiency with lower memory usage. The algorithm grows trees leaf-wise (best-first) rather than level-wise, which can lead to better accuracy. LightGBM supports parallel and graphics processing unit (GPU) learning and handles large-scale data effectively. By implementing GOSS, the algorithm reduces the effective sample size to N_r, while EFB condenses the feature space to F_b. Consequently, LightGBM achieves a streamlined space and time complexity of $O(N_r * F_b)$, representing a substantial improvement over traditional approaches and maintaining essential information. It is particularly well-suited for large datasets and high-dimensional feature spaces.

CatBoost is a powerful ensemble model designed for gradient boosting on DTs [25]. It is built to handle many categorical variables, such as categorical, textual, and numerical features, efficiently without extensive preprocessing with the help of a native feature support technique. CatBoost employs a novel technique called ordered boosting, which avoids overfitting and reduces prediction shifts by using a permutation-driven alternative to the classic gradient boosting scheme. It also uses a symmetric tree structure and implements the oblivious DT algorithm, which can lead to faster inference times and minimize overfitting. CatBoost’s weak computational complexity is generally $O(PN_{DT})$; in this context, P represents the number of subset permutations, while N_{DT} denotes the total count of DT models employed in the ensemble. The algorithm supports GPU acceleration and can be executed in parallel or distributed modes.

The training process for base learners in the BOEL framework involves independently training multiple ML models on a training dataset. This approach aims to capture diverse patterns and insights from the data, leveraging the unique strengths of various ML algorithms. By training models such as GBM, CatBoost, LightGBM, and XGBoost, the framework can leverage its capabilities in handling different aspects of the data, including high-dimensional features, non-linear relationships, and imbalanced classes. Each base

learner is trained to predict the target variable by minimizing its loss function. Subsequently, the predictions of these models are combined using optimized weights to form a robust ensemble model.

2.4. Testing/prediction process

The testing/prediction phase of the BOEL framework involves leveraging the optimized weight coefficients derived from BOA to effectively combine the predictions of the individual base learners. The goal is to maximize the overall predictive performance of the ensemble by harnessing the distinctive strengths of the constituent models. In this process, the predictions made by each base learner on the test data are weighted according to the optimized coefficients and aggregated to form a unified ensemble prediction. The aggregated prediction is then normalized to ensure it represents a valid probability distribution. The final classification decision is made based on the class with the highest probability. The optimized ensemble's performance is assessed using key metrics such as precision, F1-score, accuracy, and recall, providing a comprehensive assessment of its effectiveness in detecting various types of cyberattacks in the IoT network environment.

2.4.1. Butterfly optimization algorithm

The BOA is a nature-inspired meta-heuristic approach that models the mating and foraging actions/behaviors of butterflies [26]. The movement behavior of butterflies can be expressed as an optimization procedure, where butterflies indicate seeking entities and generated fragrances correspond to fitness values. In the BOA, butterflies (i.e., seeking entities) can produce fragrance (i.e., fitness) values with distinguishing quality over other fragrances, which is represented as (2):

$$f_k = cI_k^a \tag{2}$$

where f_k is the fragrance concentration of the k^{th} butterfly, c is a constant scaling factor (i.e., the sensory modality), I_k is the stimulus intensity, and is the fitness value of the k^{th} butterfly, and a is an exponential factor that determines the shape of the fragrance distribution depending on modality.

This behavior can assist other search entities in updating their positions within the search space. When the butterfly that locates the optimum nectar source in the search area releases a fragrance, all neighboring butterflies will fly to that butterfly's position. This update process is referred to as global search in BOA. Conversely, butterflies will randomly navigate the search space when different butterflies' fragrances are discovered, which is known as local search in BOA. The position of each butterfly individual is represented by a vector of parameter values corresponding to the problem being optimized. This position can be updated when seeking to find a more optimal position within the search space using the following mathematical formula:

$$X_k^{t+1} = X_k^t + F_k^{t+1} \tag{3}$$

where X_k^{t+1} and X_k^t represent the actual position of the k^{th} butterfly at iterations $t + 1$ and t, respectively, and F_k is the fragrance used by X_k^t for position update throughout the iterations.

As stated earlier, the update process in BOA is governed by two key mechanisms: local and global search. In the global mode, butterflies are attracted towards the top-performing butterfly b^*, which is modeled by (4):

$$F_k^{t+1} = f_k \times (r^2 \times b^* - X_k^t) \tag{4}$$

with r a numerical random factor and $r \in [0,1]$. The local mode refines the research by exploiting the nearby promising areas and is modeled by (5):

$$F_k^{t+1} = f_k \times \left(r^2 \times X_i^t - X_j^t\right) \tag{5}$$

where X_i^t and X_j^t stand for i^{th} and j^{th} butterflies' positions in the search space.

In the context of the proposed framework, the initialization phase of BOA involves generating a diverse population of butterflies, where each butterfly represents a potential weight vector for the ensemble's base learners. The fitness of each weight vector is evaluated based on the ensemble model's F1-score when using those weights. The F1-score metric was selected due to its comprehensive performance evaluation and efficacy in handling imbalanced datasets. The fragrance calculation then translates these fitness scores into signals that guide the search process, drawing butterflies toward more promising solutions. BOA's movement and position update steps employ global and local search strategies, striking a balance between exploring new

solutions and refining known good ones. This iterative process ensures a thorough search for the optimal weight combinations. The iteration phase repeats these steps until convergence criteria are met, progressively improving the solutions. The final optimized weight vector represents the best combination found, which is then used to combine the basic learners' predictions.

2.4.2. Optimized ensemble model

The ensemble prediction within the BOEL framework combines the outputs of the multiple base learners with each base learner's prediction P_j, weighted by its corresponding optimized weight w_j^*, determined through BOA. The combined ensemble prediction P^*, is the weighted sum of base learner predictions:

$$P^* = \sum_{j=1}^{n} w_j^* \cdot P_j \tag{6}$$

where n is the number of base learners. This aggregated prediction is then normalized to form a valid probability distribution:

$$P^* = \frac{P^*}{\sum_{k=1}^{C} P_k^*} \tag{7}$$

where C is the number of classes. The final class label is decided based on the selection of the class having the highest probability in the normalized prediction:

$$\hat{y} = \arg\max_k P_k^* \tag{8}$$

This process leverages the strengths of each base learner, as determined by the optimized weights, resulting in a highly accurate and robust prediction model.

2.4.3. Evaluation stage

A comprehensive evaluation is crucial to understanding the efficacy of the ensemble model, ensuring its reliable detection of diverse cyberattack types with robust performance metrics. The evaluation of the BOEL framework assesses the optimized ensemble model's performance using various assessment measures. Key metrics, such as accuracy, precision, recall, and F1-score, comprehensively evaluate the model's effectiveness in detecting cyberattacks. These primary evaluation metrics are detailed in Table 1. where TP represents true positives, TN represents true negatives, FP represents false positives, and FN represents false negatives.

Table 1. Evaluation metrics for BOEL framework

Metric	Formula	Description
Accuracy	$\frac{TP + TN}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} * 100\%$	measured as the proportion of correctly classified samples to the total number of samples
Precision	$\frac{TP}{\mathrm{TP} + \mathrm{FP}} * 100\%$	reflects the model's ability to accurately identify positive instances
Recall	$\frac{TP}{\mathrm{TP} + \mathrm{FN}} * 100\%$	also referred to as sensitivity, represents the model's capacity to detect all positive instances
F1-score	$2 \cdot \frac{\mathrm{Precision} \cdot \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}} * 100\%$	is the harmonic mean of precision and recall and provides a balanced assessment of the model's overall performance.

3. RESULTS AND DISCUSSION

This section presents the results from experiments conducted on CICIDS2017 and Bot-IoT datasets using the proposed framework. To assess the system's effectiveness, we compared the performance of the BOEL framework against four widely adopted ML gradient-boosting algorithms as base learners and other commonly used state-of-the-art ML algorithms. Due to the inherent imbalance in IoT network traffic data, where attack samples often constitute a small fraction of the overall data, the evaluation of the proposed model's performance utilizes four key metrics: accuracy, precision, recall, and F1-score. Furthermore, the experimental outcomes were validated by comparing them with findings from relevant recent studies. The results are presented in a tabular and figurative format and evaluated using the metrics discussed earlier.

3.1. Experimental setup

The experiments were conducted using the 5th iteration of the Google Collaboratory environment, which provides free access to robust computational resources, including high-performance GPUs and tensor processing units (TPUs), essential for the study. This platform offers memory capacities of up to 13 GB of RAM and Intel Xeon CPU processors with 2 virtual central processing units, making it representative of an IoT machine in terms of processing capabilities. To develop and evaluate the framework, widely used Python libraries for ML, such as LightGBM, XGBoost, CatBoost, and Scikit-learn, were employed. These libraries offered the necessary tools for efficient model training, testing, and comprehensive evaluation. The integration with Google Drive facilitated the management of datasets and results, enabling a smooth and productive research workflow.

3.2. Comparative analysis and evaluation

The experimental results presented in Figures 2 and 3 compare the performance of four base learners GBM, CatBoost, XGBoost, and LightGBM and the proposed BOEL framework on Bot-IoT and CICIDS2017 datasets. The F1-score plots in Figures 2 and 3 illustrate the varying detection capabilities of the individual base models for different types of attacks across the two datasets. These results highlight the importance of ensemble optimization, showing how combining multiple models can lead to more robust and accurate intrusion detection in IoT environments.

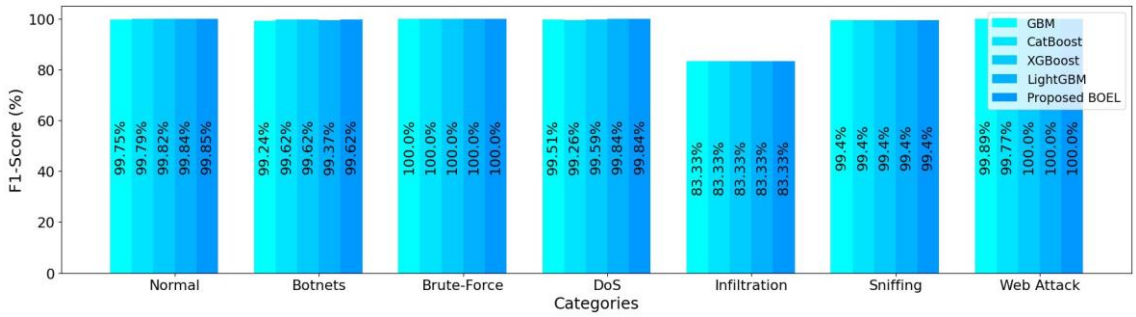

Figure 2. Models performance comparison by category on CICIDS2017 dataset

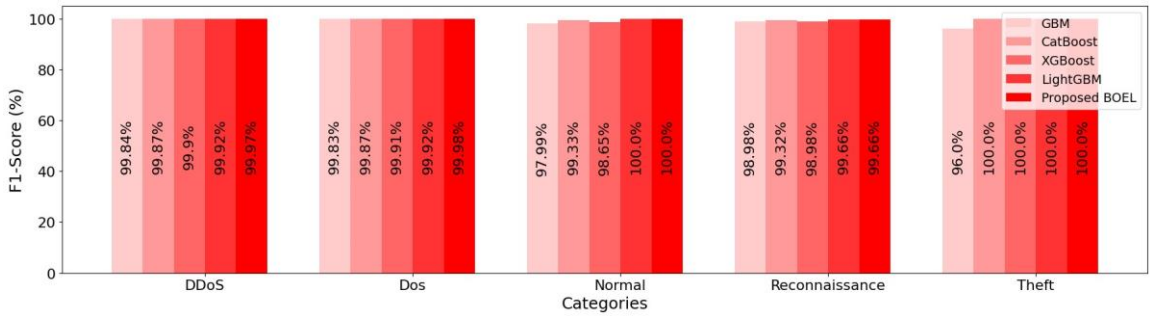

Figure 3. Models performance comparison by category on Bot-IoT dataset

The experimental results demonstrate the varying performance of the individual base learner models on the CICIDS2017 and Bot-IoT datasets. On the CICIDS2017 dataset, LightGBM consistently achieved the highest F1-score across several attack categories, including normal traffic, brute-force attacks, DoS, infiltration, sniffing, and web attacks. However, XGBoost and CatBoost were more effective than LightGBM in detecting botnet attacks. Similarly, on the Bot-IoT dataset, LightGBM outperformed the other base learners, particularly in the DDoS and DoS attack classes, where it attained the highest F1-score of 99.92% for both. XGBoost also exhibited strong performance, with F1-scores of 99.90% and 99.91% in these two attack categories. While CatBoost performed well with normal samples and reconnaissance attacks, surpassing XGBoost, it generally achieved slightly lower scores than LightGBM and XGBoost in the other attack classes, except for the theft category, where it attained a perfect F1-score of 100%, matching the top-performing models. Overall, LightGBM and XGBoost emerged as the leading models in both experiments,

demonstrating superior performance across the majority of the evaluated attack classes compared to CatBoost and GBM. As shown in Figures 2 and 3, the proposed BOEL model achieved the highest F1-score for each category, demonstrating its superior performance in intrusion detection across the evaluated IoT network environments.

Table 2 demonstrates a clear progression in performance across the evaluated ML models on the CICIDS2017. The state-of-the-art models ranged from traditional ML techniques like DT and AdaBoost to more advanced methods such as deep belief networks (DBN) and recurrent neural networks (RNN), as well as cutting-edge gradient-boosting algorithms like GBM, CatBoost, XGBoost, and LightGBM. Among the models, RNN exhibited strong performance, achieving an accuracy of 98% and an F1-score of 96%, outperforming DT and AdaBoost, which had lower accuracy and precision. Notably, AdaBoost had an accuracy of only 81.83% but compensated with a perfect recall of 100%. DBN also delivered robust performance, attaining an accuracy of 98.95%, although its F1-score was slightly lower than that of RNN, indicating a trade-off between precision and recall. The proposed framework has enhanced performance metrics, attaining 99.795% in accuracy and 99.792% in F1-score, making it the top-performing approach in the experiment. Specifically, the ensemble model outperformed RNN by 1.795% in accuracy and 3.792% in F1-score, and DBN by 0.845% in accuracy and 3.982% in F1-score. Moreover, the ensemble model slightly surpassed the advanced LightGBM method, achieving a 0.019% improvement in both accuracy and F1-score. Additionally, while XGBoost and CatBoost also exhibited excellent performance, with accuracies of 99.757% and 99.683%, respectively, the proposed ensemble model still outperformed them by 0.038% and 0.112% in accuracy and F1-score.

The experimental evaluation, as presented in Table 3, on the Bot-IoT dataset, shows a distinct hierarchy in the performance of various ML models, going from traditional methods like SVM and RF to more advanced approaches such as LS-DRNN, GBM, CatBoost, XGBoost, and LightGBM. Among the traditional methods, SVM achieved the lowest score with an accuracy of 89.35% and an F1-score of 89.34%. The RF model, while strong in certain aspects, particularly in precision, lagged with an accuracy of 98% and an F1-score of 98%. The LS-DRNN model, designed specifically for deep learning on time series data, delivered impressive results with an accuracy of 99.93% and an F1-score of 98.22%, although its precision was lower compared to the other advanced models. When compared to the gradient boosting models, LightGBM stood out with an accuracy of 99.916% and an F1-score of 99.916%, surpassing GBM, CatBoost, and XGBoost, which achieved accuracies of 99.782%, 99.849%, and 99.866%, respectively. However, the proposed BOEL framework further elevated performance, achieving an accuracy of 99.966% and an F1-score of 99.966%. This represents an improvement of 0.05% over LightGBM, 0.10% over XGBoost, 0.117% over CatBoost, and 0.184% over GBM. Compared to the LS-DRNN model, BOEL outperformed it by 0.036% in accuracy and a significant 1.746% in the F1-score, demonstrating the robustness and effectiveness of the ensemble approach. The results highlight the superiority of BOEL in achieving near-perfect classification accuracy and balanced precision and recall, making it a powerful technique for IoT intrusion detection, especially when compared to traditional ML models and more advanced gradient boosting techniques.

Table 2. Evaluation of different models' performance using CICIDS2017 dataset

Approach	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
DT [27]	96.67	97.5	85	90
Adaboost [28]	81.83	81.83	100	90.01
RNN [29]	98	96	97	96
DBN [30]	98.95	95.82	95.81	95.81
GBM	99.664	99.665	99.664	99.661
CatBoost	99.683	99.684	99.683	99.68
XGBoost	99.757	99.758	99.757	99.755
LightGBM	99.776	99.777	99.776	99.773
Proposed BOEL	**99.795**	**99.796**	**99.795**	**99.792**

Table 3. Evaluation of different models' performance using Bot-IoT dataset

Approach	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
SVM [31]	89.35	89.6	89.35	89.34
RF [32]	98	96	96	98
LS-DRNN [33]	99.93	96.87	99.75	98.22
GBM	99.782	99.782	99.78	99.782
CatBoost	99.849	99.849	99.84	99.849
XGBoost	99.866	99.866	99.86	99.866
LightGBM	99.916	99.916	99.91	99.916
Proposed BOEL	**99.966**	**99.966**	**99.96**	**99.966**

4. CONCLUSION

To enhance IoT network security, this paper introduces a novel ensemble learning approach called the BOEL framework for IDS, which provides a robust and adaptive solution to detect various types of cyberattacks. The proposed model combines ML-based techniques, particularly gradient-boosting algorithms, including GBM, CatBoost, LightGBM, and XGBoost, with the BOA method to form an optimized and efficient ensemble model. Experiments conducted on CICIDS2017 and Bot-IoT datasets demonstrated BOEL's superior capability, achieving remarkable accuracies of 99.795% and 99.966%, respectively, surpassing individual models and state-of-the-art techniques. Notably, BOEL achieved the highest performance on the Bot-IoT dataset, with a precision of 99.966%, ensuring minimal false positives; a recall exceeding 99.9%, indicating high sensitivity to subtle attack patterns; and an F1-score of 99.966%, highlighting balanced and consistent detection capabilities across diverse attack categories and underscoring its exceptional ability to handle IoT-specific threats. Furthermore, the framework excelled in identifying low-frequency attack types, such as infiltration and sniffing, which are often challenging for the existing models. BOEL's dynamic weighting mechanism further enhanced its adaptability to varying attack complexities, ensuring consistent high performance across diverse scenarios. These findings underscore BOEL's potential as a reliable, efficient, and scalable solution for protecting IoT networks against an ever-evolving landscape of cyber threats.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Mouad Choukhairi	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	
Sara Tahiri	✓	✓			✓	✓		✓		✓				
Ouail Choukhairi	✓	✓			✓	✓		✓		✓				
Youssef Fakhri		✓		✓	✓		✓			✓		✓	✓	
Mohamed Amnai				✓	✓					✓		✓	✓	

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

INFORMED CONSENT

This study did not involve human participants, and informed consent was therefore not required.

ETHICAL APPROVAL

This research did not involve human or animal subjects and did not require ethical approval.

DATA AVAILABILITY

The data that support the findings of this study are openly available:

- The CICIDS2017 dataset is available at: https://www.unb.ca/cic/datasets/ids-2017.html
- The Bot-IoT dataset is available at: https://research.unsw.edu.au/projects/bot-iot-dataset

REFERENCES

[1] I. Stellios, P. Kotzanikolaou, M. Psarakis, C. Alcaraz, and J. Lopez, "A survey of iot-enabled cyberattacks: Assessing attack paths to critical infrastructures and services," *IEEE Communications Surveys and Tutorials*, vol. 20, no. 4, pp. 3453–3495, 2018, doi:

10.1109/COMST.2018.2855563.
[2] S. Bagchi *et al.*, "New frontiers in IoT: Networking, systems, reliability, and security challenges," *IEEE Internet of Things Journal*, vol. 7, no. 12, pp. 11330–11346, 2020, doi: 10.1109/JIOT.2020.3007690.
[3] A. A. Malikopoulos, "A note for CPS data-driven approaches developed in the IDS Lab," *arXiv:2406.15496*, Jun. 2024.
[4] M. A. Khan and K. Salah, "IoT security: Review, blockchain solutions, and open challenges," *Future Generation Computer Systems*, vol. 82, pp. 395–411, 2018, doi: 10.1016/j.future.2017.11.022.
[5] R. Mills, A. K. Marnerides, M. Broadbent, and N. Race, "Practical intrusion detection of emerging threats," *IEEE Transactions on Network and Service Management*, vol. 19, no. 1, pp. 582–600, 2022, doi: 10.1109/TNSM.2021.3091517.
[6] A. Churcher *et al.*, "An experimental analysis of attack classification using machine learning in IoT networks," *Sensors*, vol. 21, no. 2, p. 446, Jan. 2021, doi: 10.3390/s21020446.
[7] M. Choukhairi, M. M'Haouach, Y. Fakhri, and M. Amnai, "Tree-driven intelligent security system for intrusion detection in IoT environment," in *2023 10th International Conference on Wireless Networks and Mobile Communications (WINCOM)*, Oct. 2023, pp. 1–7. doi: 10.1109/WINCOM59760.2023.10322954.
[8] H. Zhang, B. Zhang, L. Huang, Z. Zhang, and H. Huang, "An efficient two-stage network intrusion detection system in the internet of things," *Information*, vol. 14, no. 2, p. 77, Jan. 2023, doi: 10.3390/info14020077.
[9] R. Zhao *et al.*, "A novel intrusion detection method based on lightweight neural network for internet of things," *IEEE Internet of Things Journal*, vol. 9, no. 12, pp. 9960–9972, 2022, doi: 10.1109/JIOT.2021.3119055.
[10] S. Shitharth, P. R. Kshirsagar, P. K. Balachandran, K. H. Alyoubi, and A. O. Khadidos, "An innovative perceptual pigeon galvanized optimization (PPGO) based likelihood naïve Bayes (LNB) classification approach for network intrusion detection system," *IEEE Access*, vol. 10, pp. 46424–46441, 2022, doi: 10.1109/ACCESS.2022.3171660.
[11] Q. Abu Al-Haija and A. Al-Badawi, "Attack-aware IoT network traffic routing leveraging ensemble learning," *Sensors*, vol. 22, no. 1, p. 241, Dec. 2021, doi: 10.3390/s22010241.
[12] T. Lai, F. Farid, A. Bello, and F. Sabrina, "Ensemble learning based anomaly detection for IoT cybersecurity via Bayesian hyperparameters sensitivity analysis," *Cybersecurity*, vol. 7, no. 1, p. 44, 2024, doi: 10.1186/s42400-024-00238-4.
[13] Soni, M. A. Remli, K. M. Daud, and J. Al Amien, "Ensemble learning approach to enhancing binary classification in intrusion detection system for internet of things," *International Journal of Electronics and Telecommunications*, vol. 70, no. 2, pp. 465–472, 2024, doi: 10.24425/ijet.2024.149567.
[14] M. Balega, W. Farag, X.-W. Wu, S. Ezekiel, and Z. Good, "Enhancing IoT security: Optimizing anomaly detection through machine learning," *Electronics*, vol. 13, no. 11, p. 2148, May 2024, doi: 10.3390/electronics13112148.
[15] I. Sharafaldin, A. Habibi Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proceedings of the 4th International Conference on Information Systems Security and Privacy*, 2018, pp. 108–116. doi: 10.5220/0006639801080116.
[16] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset," *Future Generation Computer Systems*, vol. 100, pp. 779–796, 2019, doi: 10.1016/j.future.2019.05.041.
[17] R. R. Laska and A. M. Yolanda, "A comparative study of Z-score and min-max normalization for rainfall classification in Pekanbaru," *Journal of Data Science*, vol. 2024, no. 1, pp. 1–8, 2024, doi: 10.61453/jods.v2024no04.
[18] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002, doi: 10.1613/jair.953.
[19] A. A. Dehghani, N. Movahedi, K. Ghorbani, and S. Eslamian, "Decision tree algorithms," in *Handbook of Hydroinformatics*, Elsevier, 2023, pp. 171–187. doi: 10.1016/B978-0-12-821285-1.00004-X.
[20] Y. Wen, X. He, D. Lv, and F. Li, "Hybrid algorithm of gradient boosted decision tree and multiple linear regression and its application on decision prediction," in *2023 IEEE 3rd International Conference on Electronic Communications, Internet of Things and Big Data (ICEIB)*, Apr. 2023, pp. 253–256. doi: 10.1109/ICEIB57887.2023.10170440.
[21] T. Hastie, R. Tibshirani, and J. Friedman, "Boosting and additive trees," in *The Elements of Statistical Learning*, Springer New York, 2008, pp. 337–387. doi: 10.1007/978-0-387-84858-7_10.
[22] D. Hindarto, "Case study: Gradient boosting machine vs Light GBM in potential landslide detection," *Journal of Computer Networks, Architecture and High Performance Computing*, vol. 6, no. 1, pp. 169–178, 2024, doi: 10.47709/cnahpc.v6i1.3374.
[23] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug. 2016, pp. 785–794. doi: 10.1145/2939672.2939785.
[24] G. Ke *et al.*, "LightGBM: A highly efficient gradient boosting decision tree," *Advances in Neural Information Processing Systems*, vol. 2017-December, pp. 3147–3155, 2017, doi: 10.5555/3294996.3295074.
[25] P. Liudmila, G. Gleb, V. Aleksandr, D. Anna Veronika, and G. Andrey, "Catboost: unbiased boosting with categorical features," *Advances in Neural Information Processing Systems*, pp. 6638–6648, 2018.
[26] S. Arora and S. Singh, "Butterfly optimization algorithm: a novel approach for global optimization," *Soft Computing*, vol. 23, no. 3, pp. 715–734, Mar. 2018, doi: 10.1007/s00500-018-3102-4.
[27] J. Fuhr, F. Wang, and Y. Tang, "MOCA: A network intrusion monitoring and classification system," *Journal of Cybersecurity and Privacy*, vol. 2, no. 3, pp. 629–639, 2022, doi: 10.3390/jcp2030032.
[28] A. Yulianto, P. Sukarno, and N. A. Suwastika, "Improving AdaBoost-based intrusion detection system (IDS) performance on CIC IDS 2017 dataset," in *Journal of Physics: Conference Series*, 2019, vol. 1192, no. 1, p. 12018. doi: 10.1088/1742-6596/1192/1/012018.
[29] V. Ravi, R. Chaganti, and M. Alazab, "Recurrent deep learning-based feature fusion ensemble meta-classifier approach for intelligent network intrusion detection system," *Computers and Electrical Engineering*, vol. 102, p. 108156, Sep. 2022, doi: 10.1016/j.compeleceng.2022.108156.
[30] W. Elmasry, A. Akbulut, and A. H. Zaim, "Evolving deep learning architectures for network intrusion detection using a double PSO metaheuristic," *Computer Networks*, vol. 168, p. 107042, 2020, doi: 10.1016/j.comnet.2019.107042.
[31] J. G. Almaraz-Rivera, J. A. Perez-Diaz, and J. A. Cantoral-Ceballos, "Transport and application layer DDoS attacks detection to IoT devices by using machine learning and deep learning models," *Sensors*, vol. 22, no. 9, p. 3367, 2022, doi: 10.3390/s22093367.
[32] K. Ibrahimi and H. Benaddi, "Improving the IDS for BoT-IoT dataset-based machine learning classifiers," in *Proceedings - 2022 5th International Conference on Advanced Communication Technologies and Networking, CommNet 2022*, 2022, pp. 1–6. doi: 10.1109/CommNet56067.2022.9993869.
[33] S. I. Popoola, B. Adebisi, R. Ande, M. Hammoudeh, and A. A. Atayero, "Memory-efficient deep learning for botnet attack detection in IoT networks," *Electronics*, vol. 10, no. 9, p. 1104, May 2021, doi: 10.3390/electronics10091104.

BIOGRAPHIES OF AUTHORS

Mouad Choukhairi received his bachelor's degree (B.Sc.) in 2018 in mathematics and computer science and his master's degree in big data and cloud computing in 2020 from the Faculty of Sciences, Ibn Tofail University, Kenitra, Morocco. He is currently a Ph.D. student in the Computer Science Research Laboratory (LaRI) in cybersecurity, IDS, big data, ML, DL, and IoT. He can be contacted at email: mouad.choukhairi@uit.ac.ma.

Sara Tahiri obtained her bachelor's degree (B.Sc.) in Mathematics and Computer Science in 2018 and her master's degree in big data and cloud computing in 2020, both from the Faculty of Sciences in Kenitra, Morocco. Currently, she is pursuing a Ph.D. at the Computer Science Research Laboratory (LaRI), focusing on big data, ML, DL, and IoT. She can be contacted at email: sara.tahiri@uit.ac.ma.

Ouail Choukhairi received his bachelor's degree (B.Sc.) in 2020 in mathematics and computer science and his master's degree in big data and cloud computing in 2022 from Faculty of Sciences, Ibn Tofail University, Kenitra, Morocco. He is currently a Ph.D. student in the Computer Science Research Laboratory (LaRI) in the field of artificial intelligence, semantic segmentation, ML, DL. He can be contacted at email: ouail.choukhairi@uit.ac.ma.

Youssef Fakhri obtained a bachelor of science degree in Electronics and a Diploma of Advanced Scientific Studies (DESA) in computer science and telecommunications from the Faculty of Sciences of Rabat (Mohammed V University - Agdal, Rabat, Morocco), in 2001 and 2003 respectively. He obtained a Ph.D. thesis on November 17, 2007, from the University Mohammed V - Agdal, Rabat, Morocco in collaboration with the Polytechnic University of Catalonia (UPC), Spain.His research topics are information theory, signal processing and wireless telecommunications (WSN), and routing protocols. He is now a Professor of Higher Education in Computer Science at the Faculty of Sciences of Kenitra. The author is a head of (LaRI) Research Laboratory in Computer Science. Further info on his homepage: https://sites.google.com/site/proffakhri/. He can be contacted at email: FAKHRI@uit.ac.ma.

Mohamed Amnai obtained his bachelor's degree in 2000 in IEEA (Computer Science, Electronics, Electricity and Automation) at the University Moulay Ismail in the city of Errachidia. Then he obtained his master's degree in 2007 and his Ph.D. in 2011 in Telecommunications and Computer Science at the University Ibn Tofail of Kenitra, Morocco. His current research interests include QoS in wireless communication, ad hoc networks, and telecommunication traffic. He is a member of (LaRI) Research Laboratory in computer science, and he is currently working as a Computer Science Professor at the Ibn Tofail University of Kenitra, Morocco. He can be contacted at email: mohamed.amnai@uit.ac.ma.