An Explainable Ensemble Feature Selection Framework for Enhanced IoT Edge Attack Detection

Hoang Trong Minh, Le Thi Trang Linh, Nguyen Minh Hoang, Ngo Thi Thu Trang, and Lam Sinh Cong*

Original scientific article

***Abstract*—Intrusion detection systems (IDS) deployed on edge-network devices are rapidly increasing as the edge computing approach is widely applied in network infrastructure. Despite the advantages that edge computing offers, enhanced network performance in terms of latency, bandwidth consumption, and privacy, there have remained challenges related to the constrained resources of edge devices. In machine learning-based IDS systems, to deal with large, noisy, or strong feature correlations in practical IoT data, dimensionality reduction methods are often applied to generate efficient inputs for machine learning models. However, a major challenge is enhancing both the performance and explainability of the edge IDS system to meet the requirements of the security field. To overcome the challenge, we propose an explainable hybrid feature-selection framework (X-EFS) that combines multiple feature reduction techniques via a multi-expert system module, then uses the MDA metric to select the most important features, ensuring high performance and explainability. Evaluations on the CICIoT2023 dataset indicate that condensing the feature space from 46 to 6 dimensions yields an Accuracy of 98.33% and an F1-score of 98.54% across a 5-fold stratified validation. This extreme dimensionality reduction directly mitigates inference latency and memory overhead, satisfying edge-constrained deployment requirements without compromising detection capacity or intrinsic explainability.**

***Index Terms*—Feature Selection, Ensemble Methods, Edge Computing, Explainable AI, IoT Security, Attack Detection.**

I. Introduction

THE rapid expansion and diversification of Internet of Things (IoT) devices and applications have produced vast and heterogeneous streams of high-dimensional data [1]. Empirical studies in recent years indicate that IoT-related attacks have grown at an annual rate exceeding 30%, resulting in substantial economic losses from service disruptions, data leaks, and infrastructure damage [2]. The growing scale and complexity of these problems put significant pressure on traditional centralized processing architectures, affecting both performance and network security [3].

To tackle this issue, the edge computing approach provides effective network performance solutions by bringing computation closer to the data source, which reduces bandwidth usage and transmission latency. However, edge devices typically operate under limited resource constraints, which significantly challenge the deployment of complex security solutions at the network edge [4]. At the same time, deploying edge-based IDS enables local analysis of traffic, reducing the amount of data sent to centralized servers and thereby lowering privacy exposure while improving security [5].

To face resource-constrained edge devices, several studies used feature selection methods not only to reduce computational costs but also to retain the most important attributes for accurate attack classification [6]. In edge-based IDS systems, feature selection plays a critical role by directly reducing memory usage, complexity, and inference latency, which are key factors that enable deployment on low-cost devices [7]. In this direction, three categories of the feature selection approach include (1) filter-based methods (mutual information) [8], (2) wrapper-based methods (recursive feature elimination), and (3) embedding methods (e.g., Lasso, tree-based importance) [9]. Each method has specific advantages and disadvantages when dealing with the multimodal, noisy, and highly correlated nature of practical IoT data [10]. On the other hand, ensemble feature selection approaches can improve robustness but often reduce explainability, which is especially important in the security context, where analysts require trustworthy, transparent decision-making logic [11]. Explainability plays a crucial role in security systems because it directly affects the reliability of decisions [12]. However, existing IDS frameworks primarily focus on attack-detection performance or on reducing complexity, neglecting the trade-off between system performance and explainability [13].

Hence, computational efficiency and explainability are simultaneous goals in edge IDSs. In this study, we propose a novel framework that utilizes the Mean Decrease Accuracy (MDA) metric to reach these goals. In more detail, our proposed framework combines three kinds of feature selection methods into a multi-expert decision module. Their feature outputs are aggregated and ranked based on their maximum MDA values in an MLP model that optimizes a minimal feature subset while maintaining the attack detection rate. This proposed framework ensures all intrinsic explainability, a high detection rate, and computational efficiency for effective IDS

Manuscript received February 4, 2026; revised March 3, 2026. Date of publication August 17, 2026. Date of current version August 17, 2026. The associate editor prof. Toni Perković has been coordinating the review of this manuscript and approved it for publication.

H. T. Minh and N. T. Thu Trang are with the Faculty of Telecommunications 1, Posts and Telecommunications Institute of Technology, Vietnam.

L. T. Trang Linh is with the Faculty of Information Technology, Electric Power University, Vietnam.

L. S. Cong and N. M. Hoang are with the Faculty of Electronics and Telecommunications, VNU University of Engineering and Technology, Hanoi, Vietnam (e-mail: congls@vnu.edu.vn).

Digital Object Identifier (DOI): 10.24138/jcomss-2026-0051

*Corresponding author

deployed on edge devices.

The main contributions are listed below:

- We propose an explainable multi-expert ensemble framework that combines and selects feature selection methods through a multi-decision module based on the MDA metric to enhance explainability.
- We propose a lightweight MLP-based optimization solution to achieve high discrimination capacity at low computational cost, making it suitable for resource-constrained edge devices.
- We validate the proposed framework on the CICIoT2023 dataset, demonstrating preferable performance (Accuracy of 98.33%, F1-score of 98.54%) with only 6 features, reducing inference time significantly while maintaining high explainability.

The paper is organized as follows. Section II briefly reviews related studies on feature selection and explainability in IDS systems. Section III introduces the proposed X-EFS framework and its main components. Section IV presents the experiments and results using the benchmark CIC IoT 2023 dataset. Section V concludes the paper and discusses our future work.

II. Related Work

In an IDS based on machine learning, the data processing stage transforms data into feature sets that serve as inputs to the machine learning model. Feature selection methods play an important role in reducing data dimensionality, removing noise, and improving model performance. There are three main types of feature selection methods: filters, wrappers, and embedded methods. Filter-based methods (such as mutual information) offer computational efficiency and scalability for high-dimensional datasets by evaluating features based on statistical properties. However, these methods assess features individually, which prevents them from capturing feature interactions that may critically affect predictive performance [14]. Wrapper methods (such as recursive feature elimination) evaluate groups of features using a classifier to achieve better detection accuracy; however, they require longer training times and pose challenges for devices with limited resources [15]. Embedded methods (such as LASSO) perform feature selection during training, balancing computational efficiency and model accuracy [16].

Otherwise, ensemble methods like majority voting [17] and hybrid schemes that combine filter pre-selection with tree-based importance ranking [18] have been proposed to enhance stability purposes. However, the above methods do not provide detailed transparency into feature selection decisions, leading to a lack of clarity regarding intrinsic explainability. To improve explainability, recent explainable AI (XAI) tools, such as Shapley Additive exPlanations (SHAP) and Local Interpretable Model-Agnostic Explanations (LIME), have been adopted in intrusion detection systems. However, this approach is post-explainability. In [19], the authors focused on intrinsic explainability in the design of the intrusion detection system. They use ensemble-based feature selection for intrusion detection and attention-driven web threat identification.

All of these studies clearly demonstrate the importance and tight relationship between feature selection and explainability. This problem is especially important for the security field. In IDS systems deployed at the network edge, a key challenge in the feature selection problem is the trade-off between improving performance to tailor the edge device and maintaining reasonable interpretability [20]. In the case of security for IoT and industrial systems, the authors in [21] showed that feature selection can improve performance for IoT and cyber-physical systems (CPS). The authors in [22], [23] confirmed that feature selection methods significantly influenced the decision-making process in IoT intrusion detection systems.

Regarding explanation concerns, recent studies have focused on embedding interpretability directly into the feature selection stage using various techniques. In [24], the authors developed an explainable machine learning model for detecting web attacks in IIoT environments. In [25], the authors proposed an IoT security framework that combines two feature selection methods with post-hoc explainability on the CIC IoT 2023 dataset [26]. In detail, the validation results from this dataset indicate that the proposed model has achieved a high attack detection rate of 99.7% and is explainable using SHAP and LIME tools. The above studies all confirm the importance of model interpretability alongside performance parameters. Hence, our framework directly tackles this issue to improve attack detection rates and provide intrinsic explainability to the IoT IDS system operating with limited resources.

III. Proposed Framework

A. Feature selection Metric

In an IoT intrusion detection system, feature selection must account for two data characteristics, such as sensor data/environmental noise and feature correlations (eg, flow duration, packet count, and total size). The use of measurement metrics is crucial for feature selection methods. Impurity-based metrics, such as Mean Decrease Impurity (MDI) in decision trees, are known to exhibit bias towards features with wide domain values or high correlations with the output [27]. However, MDI systematically assigns higher importance to noisy features, particularly numerical or categorical features with many categories, leading to a bias in MDI feature selection.

In contrast, the Mean Decrease Accuracy (MDA) metric is a feature selection method that assesses feature importance by measuring the impact of perturbing a feature on a model's predictive performance. This method involves randomly changing the values of a feature and catching how it affects the model's accuracy. A bigger drop in accuracy means that the feature is more important. However, the MDA metric is subject to intrinsic randomness, which can affect the stability and interpretability of the selected features [28]. We detail the MDA below.

Formally, let $\mathcal{M}$ be a trained classifier with accuracy $\text{Acc}(\mathcal{M})$ on a test set $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$. For a feature X_j, its

theoretical MDA is defined as the expected drop in accuracy when X_j is randomly permuted as,

$$\text{MDA}(X_j) = \mathbb{E}[\text{Acc}(\mathcal{M}) - \text{Acc}(\mathcal{M}|\tilde{X}_j)], \quad (1)$$

where $\tilde{X}_j$ denotes a random permutation of the j-th feature column, and the expectation is taken over the space of all possible permutations. In practice, this expectation is approximated by averaging over a finite number R of independent random permutations. The empirical MDA is thus computed as,

$$\text{MDA}(X_j) = \frac{1}{R}\sum_{r=1}^{R}(\text{Acc}(\mathcal{M};\mathcal{D}) - \text{Acc}(\mathcal{M};\mathcal{D}^{(r)}_{\tilde{X}_j})), \quad (2)$$

where the notation is defined as follows:

- $\mathcal{M}$: the trained ensemble classifier (e.g., voting/stacking of RF, LR, SVC)
- $\mathcal{D}$: the original test set
- $\tilde{X}_j^{(r)}$: the r-th random permutation of feature X_j
- $\mathcal{D}^{(r)}_{\tilde{X}_j}$: the test set with the j-th feature column replaced by $\tilde{X}_j^{(r)}$
- $\text{Acc}(\mathcal{M};\mathcal{D})$: model accuracy on the original test set
- $\text{Acc}(\mathcal{M};\mathcal{D}^{(r)}_{\tilde{X}_j})$: accuracy after permuting feature X_j in the r-th trial
- R: number of permutation trials (set to $R = 20$ in our experiments)

In fact, we set $R = 20$ based on empirical stability analysis, where increasing beyond 20 yields only marginal gains in ranking consistency (Spearman's $\rho > 0.95$ for top features), while $R = 10$ exhibiting higher variance. Hence, this parameter balances computational cost and estimation reliability. Moreover, this formulation yields robustness under noise and correlation. As shown in [29], MDA remains asymptotically unbiased under uncertain conditions even when features are noisy or highly correlated in practical conditions of IoT data traffic environments. Empirically, MDA maintains stable importance rankings under non-stationary data streams, whereas MDI can spuriously distribute importance across redundant features. Recent theoretical work further confirms that standard MDA may suffer from inconsistency under feature dependence, but practical variants (e.g., Sobol-MDA) offer consistent estimation [30]. Therefore, MDA provides a theoretically grounded and intrinsically interpretable metric for our ensemble feature selection framework.

B. Multi-expert Feature Selection Module

To handle the heterogeneous, noisy, and strongly correlated IoT data traffic, we integrate five representative feature selection algorithms into a multi-expert system module. This approach aims to capture multiple aspects of feature relevance, including statistical dependence (filter), model-specific predictive utility (wrapper), sparsity-inducing regularization (embedded), and nonlinear information-theoretic relationships. The brief introductions to the selected algorithms in Figure 1 are listed below.

SelectKBest (Filter). Given a feature matrix $X \in \mathbb{R}^{n\times p}$ and label vector $y \in \mathbb{R}^n$, SelectKBest computes a univariate score

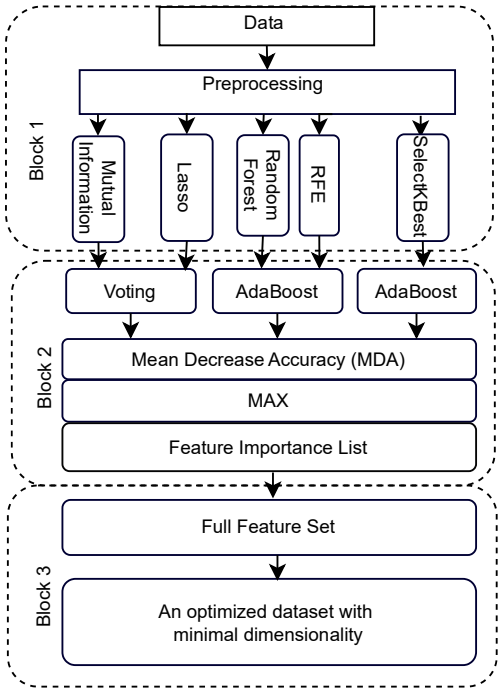


Fig. 1. The proposed explainable feature selection framework

$S_i = f(x_i, y)$ for each feature x_i (e.g., ANOVA F-statistic or mutual information) and selects the top-k features:

$$K = \arg\max_{\substack{S\subseteq\{1,\ldots,p\}\\|S|=k}} \sum_{i\in S} S_i. \quad (3)$$

This method is computationally efficient and robust to model misspecification, making it ideal for initial screening in edge environments.

Recursive Feature Elimination (RFE, Wrapper). RFE iteratively removes the least important feature based on a base estimator's coefficients or sensitivities. Formally, let $\mathcal{M}$ be a predictive model (e.g., logistic regression). At each iteration, feature importance scores r_i (e.g., $r_i = |w_i|$ for linear models) are computed, and the feature with the minimal score r_i is eliminated. The procedure is detailed in Algorithm 1.

Algorithm 1 Recursive Feature Elimination (RFE)

Require: Feature matrix X, labels y, target cardinality k.
Initialize feature set $S \leftarrow \{1, \ldots, p\}$.
while $|S| > k$ **do**
  Train model $\mathcal{M}$ on $X[:, S]$.
  Compute ranking scores r_i for all $i \in S$.
  Remove $i^* = \arg\min_{i\in S} r_i$.
  Update $S \leftarrow S \setminus \{i^*\}$.
end while
Return S.

Since RFE is executed only once during offline model preparation, its higher computational overhead is justified by the ability to capture feature interactions.

Random Forest Importance (Embedded, MDI). Although we avoid MDI for final aggregation due to its bias under correlation, we include it as one expert to leverage its ability

to model nonlinear feature interactions. For a tree T, the impurity-based importance of a feature j is

$$V_j(T) = \sum_{\{t \in T\}} p(t) \cdot \Delta\text{Imp}(t, j) \cdot \mathbb{1}(v_t = j), \tag{4}$$

where $p(t)$ is the proportion of samples reaching node t, $\Delta\text{Imp}(t, j)$ is the impurity reduction from splitting on feature j at node t, and $\mathbb{1}(\cdot)$ is the indicator function. The ensemble importance is averaged over B trees: $V_j = \frac{1}{B}\sum_{b=1}^{B} V_j(T_b)$.

Lasso (Embedded, Sparsity-Inducing). Lasso solves the regularized optimization problem:

$$\min_{\beta} \frac{1}{2n}|y - X\beta|_2^2 + \lambda|\beta|_1, \tag{5}$$

which drives irrelevant coefficients to zero, yielding a sparse solution. The coordinate descent update for feature j is

$$\hat{\beta}_j = \text{sign}(z_j)\max(|z_j| - \lambda, 0), \text{ where } z_j = \frac{1}{n}X_j^{\top}(y - X_{-j}\beta_{-j}). \tag{6}$$

Mutual Information (Filter, Nonlinear). Mutual Information (MI) quantifies the general statistical dependence between a feature X and label Y:

$$I(X;Y) = \sum_x \sum_y p(x,y) \log \frac{p(x,y)}{p(x)p(y)} = H(Y) - H(Y|X), \tag{7}$$

where $H(\cdot)$ denotes entropy. Continuous-valued features are handled using the Kraskov k-NN neighbor mutual information estimator with $k = 3$. This estimator captures nonlinear dependencies and has been reported to be effective at recognizing attack signatures, particularly under bursty traffic conditions.

The proposed multi-expert module combines five complementary feature selection algorithms to construct five sets of candidate features. This process constructs stable sets as the input to the subsequent Feature Importance Evaluation Block.

C. Feature Importance Evaluation Module

The output sets generated by the multi-expert system module are then combined using the Feature Importance Evaluation Block. As shown in Fig. 1, this block incorporates three ensemble learners (soft majority voting, AdaBoost, and stacking) with detailed configurations provided in Table II. Feature relevance is evaluated through the MDA metric. For each ensemble e, the importance of a selected feature X_j is obtained by applying equation (2) with 20 random permutations, resulting in the score $\text{MDA}^{(e)}(X_j)$. To measure feature importance across ensembles, we define the aggregated score as:

$$\text{MDA}_{\text{MAX}}(X_j) = \max_{e \in \mathcal{E}: X_j \in F_e} \text{MDA}^{(e)}(X_j), \tag{8}$$

where F_e represents the feature subset that e has been selected, and $\mathcal{E}$ is the collection of all ensembles e. Features that have a higher predictive impact in at least one modelling context produce a high MDA_{MAX} value and are therefore prioritized.

To validate the effectiveness of the MDA_{MAX} aggregation operator in Eq. (8), we benchmarked it against a standard Mean-MDA baseline. While both configurations yield statistically similar macroscopic classification metrics, their security implications diverge. Averaging feature importance via Mean-MDA mathematically smooths expert disagreement, thereby suppressing isolated deviations caused by minority attack vectors that only specific wrapper or embedded models detect. Conversely, selecting the maximum expected drop, MDA_{MAX}, explicitly preserves these highly specialized protocol signatures, ensuring that nuanced anomalies are retained in the reduced subspace.

Features are ordered by their MDA_{MAX} scores and then put into the pruning stage. The pruning module keeps a compact subset of features to enable efficient deployment on edge devices.

D. Feature Set Optimization Module

To tackle the edge resource constraints condition, a lightweight MLP is used for validation during optimization, as it captures nonlinear feature relationships and yields more stable performance than tree-based models. The MLP also serves as a lightweight classifier because it has a low-complexity structure of only two hidden layers and dropout. By utilizing an iterative backward elimination procedure operated by the MLP, we identify a minimal yet high-performing feature subset. Starting from the full ranked list $\{f^{(1)}, f^{(2)}, \ldots, f^{(m)}\}$, sorted in descending order MDA_{MAX} in the previous phase, the algorithm evaluates the impact of removing the least important feature at each step. Let $F^{(k)}$ be the set of features after k removals, and let $\Phi_{\text{MLP}}(\cdot)$ be a validation model trained on $F^{(k)}$. We start from the baseline scenario with all $m = 46$ features and compute the baseline performance $P_{\text{full}} = \Phi_{\text{MLP}}(F)$. At each iteration k, the lowest-ranked feature is removed and the score $P^{(k)}$ is recomputed on the remaining set via the MLP-based evaluator. The procedure terminates when $P^{(k)} < P_{\text{full}}$, and the subset from the previous step, $F^{(k-1)}$, is returned as the final set to maintain baseline detection capability. The full procedure is provided in algorithm below 2.

Algorithm 2 Iterative Feature-Set Optimization

Require: Full feature set $F = \{f_1, \ldots, f_m\}$, validation model $\Phi_{\text{MLP}}(\cdot)$.
Compute baseline performance $P_{\text{full}} \leftarrow \Phi_{\text{MLP}}(F)$.
Obtain feature ranking $\{f^{(1)}, f^{(2)}, \ldots, f^{(m)}\}$ (descending by MDA_{MAX}).
Initialize $F^{(0)} \leftarrow F$, $k \leftarrow 0$
repeat
    $k \leftarrow k + 1$.
    Remove the lowest-ranked feature, $F^{(k)} \leftarrow F^{(k-1)} \setminus \{f^{(m-k+1)}\}$.
    Evaluate performance, $P^{(k)} \leftarrow \Phi_{\text{MLP}}(F^{(k)})$.
until $P^{(k)} < P_{\text{full}}$
return $F_{\text{selected}} \leftarrow F^{(k-1)}$

The algorithm derives a compact feature subset without relying on fixed thresholds, while preserving detection rate performance. In our practical evaluation, six features are retained (IAT, Tot size, Weight, Number, urg_count, and rst_count), and the MLP is used solely for performance assessment during selection.

For the MDA value estimation process, each feature is permuted 20 times, and the average values are collected. The pruning process is validated using a lightweight MLP with two hidden layers (128 and 64 units), enabling low-cost evaluation in edge settings. Finally, soft voting, AdaBoost, and stacking

are applied to the top-10 features, where we choose a high rank correlation ($\rho > 0.85$) to ensure consistent importance estimates across ensembles.

IV. Experimental Setup and Results

A. Dataset Description and Preprocessing

Our experiments are conducted using the CIC IoT 2023 dataset [26]. This dataset includes 46 flow-level features and represents 33 attack types organized into 7 categories (*DDoS, DoS, reconnaissance, web-based, brute force, spoofing, and Mira*).

The experimental validation utilizes a subset of 466,866 network flow records from the CICIoT2023 dataset. To align with our edge-based threat quarantine objective, all malicious instances are aggregated into a unified class, formulating a strictly binary classification task (Benign vs. Attack). The aggregated data exhibits a heavily skewed class distribution, accurately reflecting operational IoT environments. A preliminary integrity check yields a 0% drop ratio, as the raw CSV subset contains no missing or degenerate features.

During preprocessing, Z-score standardization is strictly fitted on the training split to prevent distributional leakage. To counter class imbalance without incurring the prohibitive memory footprint of synthetic oversampling (e.g., SMOTE) on edge nodes, an inverse-frequency penalty is embedded directly into the optimization loss function. Specifically, the weight for each binary class $c \in \{\text{Benign}, \text{Attack}\}$ is mathematically formulated as $W_c = N/(2 \cdot n_c)$, where N represents the total training samples, and n_c is the cardinality of class c. Finally, to ensure robust evaluation across the imbalanced distribution, we partitioned the data using a 5-fold stratified cross-validation scheme. This protocol preserves the native skewness of the IoT traffic across all training and validation folds, establishing a rigorously unbiased baseline for subsequent performance evaluations. To ensure the reproducibility of our findings, the source code and implementation details of the proposed X-EFS framework have been made available at https://github.com/ICNLABPTIT/Explainable_Ensemble_Feature_Selection_CICIoT2023.git.

B. Multi-Expert Selection and Aggregation

We evaluated 5 feature selection algorithms using the parameters shown in Table I and combined their outputs using 3 ensemble strategies (Table II).

TABLE I
Feature-selection methods and main parameters

Method	*Key Parameters*
SelectKBest (ANOVA F-test)	$k =$ all
Recursive Feature Elimination	Estimator: LogisticRegression, select 6 features
Random Forest Importance	100 trees
L1-based Selection (Lasso)	$\alpha = 0.01$, L1 penalty
Mutual Information	neighbours = 3

For each ensemble, we computed the MDA value of every selected feature using Eq. (2) with $R = 20$ permutations, then aggregated the results using the MDA_{MAX} strategy. Table III shows the top features ranked by maximum observed MDA across ensembles. Using the iterative MLP-based pruning procedure (Algorithm 2), we reduced the original 46 features to a final compact subset of 6 features. (*IAT, Tot size, Weight, Number, urg_count, and rst_count*).

TABLE II
Ensemble strategies and their configurations

Strategy	*Learners*
Majority Voting (soft)	Random Forest, Logistic Regression, SVC
AdaBoost	Random Forest (base learner)
Stacking	Random Forest, AdaBoost, Logistic Regression

TABLE III
Top features ranked by maximum MDA value

Feature	*MDA*
IAT	0.018609
Weight	0.015912
Number	0.015248
urg_count	0.014925
rst_count	0.013194
Tot size	0.013179
flow duration	0.011249
Rate	0.010448
Max	0.010024
Srate	0.009339
AVG	0.008696
Covariance	0.007846

C. Evaluation Metrics and Main Results

We examine our proposed framework over standard performance metrics (Accuracy, Precision, Recall, and F1-score). We summarize the main results comparing the full feature set (46 features) with the optimised 6-feature subset selected by Algorithm 2 below.

The optimized 6-feature subset demonstrates consistent structural stability. Under the 5-fold evaluation protocol, the compact model preserves an Accuracy of $98.33\% \pm 0.07\%$ and an F1-score of $98.54\% \pm 0.05\%$. The tight confidence intervals suggest that truncating the feature space eliminates spurious, non-generalizable correlations, forcing the classifier to rely strictly on the remaining linearly independent protocol attributes for threat detection.

Confusion matrices for models trained with different feature subset sizes (46, 30, 20, 15, 10, 6 features) are shown in Fig. 2.

Figure 3 compares the evaluation metrics between the 46-dimensional and 6-dimensional models. The results indicate that truncating 86.9% of the feature space incurs only a 0.84% degradation in F1-score (from 99.38% to 98.54%). This proportional retention of classification capacity implies that the discarded 40 features predominantly encode redundant

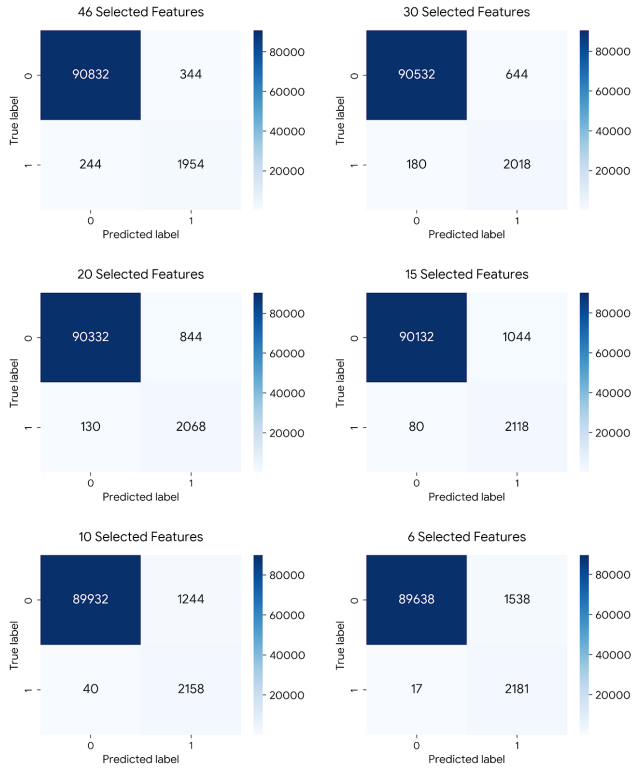


Fig. 2. Confusion matrices for models with different feature set sizes.

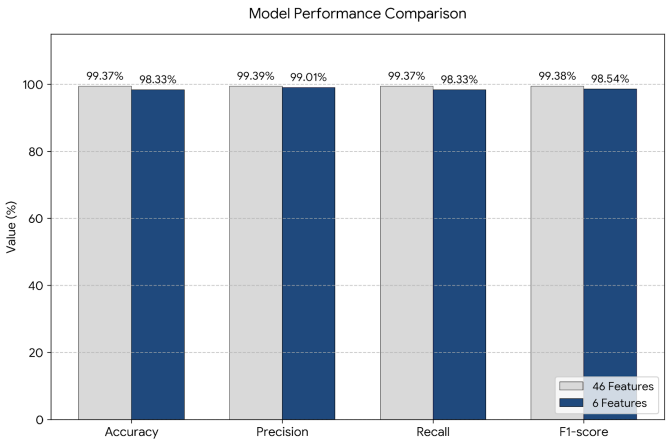


Fig. 3. Comparison of evaluation metrics between models using 46 and 6 features.

flow statistics or localized sensor noise rather than unique cryptographic or topological attack patterns.

The classification objective is intentionally restricted to a binary taxonomy (Benign vs. Attack). In resource-depleted edge ecosystems, minimizing false negatives to trigger immediate quarantine supersedes fine-grained botnet attribution. Resolving structural aliasing among 33 specific attack variants necessitates higher-dimensional hyperspaces, which strictly violates the 168.3 KB memory constraint imposed by IoT micro-architectures.

Regarding misclassification patterns, our error analysis indicates that the compact 6-feature subset primarily increases False Negatives for low-volume, stealthy reconnaissance attacks. These specific attack vectors exhibit packet counts and payload sizes that are statistically similar to the benign baseline traffic. Resolving this classification overlap would require evaluating additional temporal features, which strictly violates the 168.3 KB memory constraint imposed by IoT micro-architectures. Therefore, we explicitly trade off the fine-grained detection of low-volume anomalies to maintain microsecond-level latency and strict memory compliance at the network edge.

D. Robustness and Efficiency Evaluation

To evaluate the practical applicability of X-EFS, we analyze its efficiency and robustness to input noise across simulations. Inference latency is measured in a unified software environment using scikit-learn and a lightweight MLP on a standard workstation (Intel Core i7-1165G7 CPU, 16 GB RAM, Ubuntu 22.04, Python 3.10). The proposed model optimizes to use only 6 features and requires 12.7ms per inference, compared with 48.3ms for the full 46 features, resulting in a 73.7% reduction in latency. This optimization supports deployment on resource-limited edge platforms such as the Raspberry Pi 4.

Structural stability against sensor-level degradation is quantified by injecting continuous Gaussian noise $\mathcal{N}(0, \sigma^2)$ across the standardized test space. As detailed in Table IV, the F1-score fluctuates marginally, peaking at 0.9858 for $\sigma = 0.25$. This dynamic suggests that the additive noise functions as an implicit stochastic regularizer (analogous to Tikhonov regularization), softening the decision boundaries and preventing the Multi-Layer Perceptron from overfitting to transient, non-stationary artifacts in the IoT training distribution.

As illustrated in Fig. 4, we evaluate the system's operational stability during continuous streaming scenarios, exposing the extreme distributional shifts caused by bursty IoT traffic.

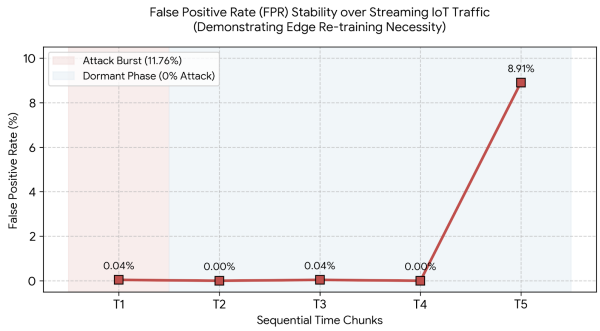


Fig. 4. False Positive Rate (FPR) stability observed through sequential evaluation windows ($T_1 \rightarrow T_5$). The sudden FPR surge in T_5 illustrates the severe concept drift inherent in continuous IoT traffic streams, empirically demonstrating the necessity for periodic edge re-training.

Direct micro-architectural profiling of the 6-feature MLP confirms its viability for edge deployment. Truncating the input vector proportionally compresses the computational graph, yielding an architecture with 11,298 parameters and a memory footprint of 168.3 KB. Sequential inference profiling on an unoptimized CPU translates to an execution latency of 2.48 ms per sample (throughput of 402.1 samples/s).

TABLE IV
ROBUSTNESS OF THE MODEL UNDER GAUSSIAN NOISE PERTURBATION

σ	F1-score	F1-drop
0	0.98541	0
0.05	0.98508	0.00033
0.10	0.98522	0.00019
0.15	0.98535	0.00006
0.20	0.98559	-0.00018
0.25	0.98587	-0.00046
0.30	0.98548	-0.00007

Continuous streaming non-stationarity is simulated via a chronological sliding-window evaluation ($T_1 \rightarrow T_5$). This temporal progression captures the distributional shifts inherent to IoT networks: T_1 isolates a concentrated attack burst (11.76% malicious traffic), whereas subsequent windows ($T_2 \rightarrow T_5$) evaluate prolonged dormant phases devoid of attacks (0% attack ratio).

Standard threshold metrics such as the F1-score mathematically collapse under single-class dormant streams, necessitating the use of the False Positive Rate (FPR) for stability tracking. Although the system sustains near-zero FPR throughout $T_2 \rightarrow T_4$, cumulative deviations in benign protocol behaviors (e.g., shifting baseline inter-arrival times or varying payload weights in standard IoT periodic reporting) induce a concept drift, driving the FPR to 8.91% at T_5. This degradation underscores a fundamental limitation in static edge deployments: monolithic models inevitably decay. The 168.3 KB footprint of X-EFS resolves this by accommodating high-frequency, localized weight updates directly on the edge node, bypassing the latency of cloud-dependent retraining to suppress localized false alarms.

E. Explainability Analysis (SHAP/LIME)

While the MDA_{MAX} metric provides intrinsic interpretability exclusively for the feature selection stage, the non-linear decision boundary generated by the subsequent MLP acts as an opaque process. To prevent overstating the methodological contribution, our framework is formulated as utilizing interpretable feature selection validated via post-hoc explainability methods. Accordingly, the global explainability of the predictive pipeline is validated by applying two post-hoc tools, SHAP and LIME (6 features). LIME explanations for individual attack predictions consistently indicate that *Weight, Number, and rst count* are the dominant contributors...

As shown in Fig. 5, the bar chart illustrates the contribution of each feature to the model's prediction. In which positive values tend to push the model toward an attack, while negative values tend to push it toward normal. Features such as *Weight, Number, and rst count* exhibit strong positive contributions, aligning with known attack signatures (e.g., high packet counts or *TCP RST flags*). Otherwise, SHAP summary plots rank IAT (inter-arrival time) and Number as the most influential features across the entire test set. This aligns with network security intuition: a low *IAT* indicates bursty traffic (common in flooding attacks), while a high *IAT* indicates abnormal packet volume.

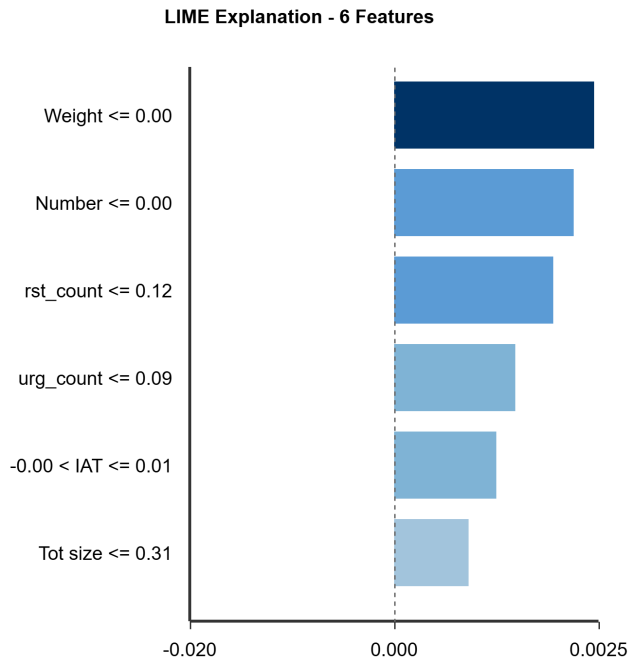


Fig. 5. LIME explanation using 6 selected features

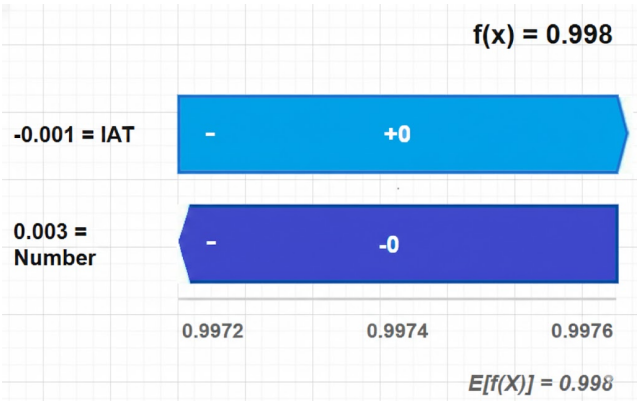


Fig. 6. SHAP explanation using 6 selected features

In Fig. 6, the bar chart shows the contribution of each feature to the model's prediction as the red bars push toward attack (positive SHAP value) and the blue bars toward normal (negative SHAP value). Features such as IAT and Number exhibit strong positive contributions, aligning with known attack signatures (e.g., bursty traffic or high packet volume).

Critically, both SHAP and LIME rankings strongly align with our MDA_{MAX}. The Spearman rank correlation between the MDA_{MAX} ranking and the mean absolute SHAP value is $\rho = 0.87$, indicating strong global alignment between the feature-selection metric and the post-hoc explanations. We also assessed explanation stability by computing Spearman's ρ across feature rankings obtained from five independent training seeds. Stability for the top-10 features exceeds $\rho > 0.85$, confirming that interpretability is consistent across model instantiations.

F. Performance Comparison

Benchmarking X-EFS against contemporary architectures on the CICIoT2023 dataset (Table V) establishes its relative detection boundary. The baseline proposed by Neto et al. (2023) [26] yields an F1-score of 96.52% and a Precision of 96.51%. The X-EFS framework pushes these metrics to an F1-score of 98.54% and a Precision of 99.01%. Preserving high precision confirms that isolating 6 dominant features through MDA_{MAX} successfully filters out sensor noise that typically generates false positives. The architecture balances discriminative capacity with hardware limitations, providing a mathematically justified deployment path for edge IDS configurations.

TABLE V
PERFORMANCE COMPARISON WITH RECENT STUDIES

Model	*Acc (%)*	*Prec (%)*	*Rec (%)*	*F1-score (%)*
Alqahtany et al. (2025)	88.64	91.20	88.64	88.51
Neto et al. (2023)	99.68	96.51	96.53	96.52
Ours (6 features)	**98.33**	**99.01**	**98.33**	**98.54**

Although the experimental results are superior to those of previous studies, this research has some limitations in its experimental implementation. Inference time is performed on a workstation and may therefore vary in real environments. Furthermore, calculating the MDA value via the permutation algorithm will also consume energy.

G. Discussion on Generalizability

The current validation is confined to the CICIoT2023 dataset. However, there are strong grounds to believe that the proposed framework can be extended to a wider range of scenarios. In particular, the multi-expert module is built upon five complementary selectors that operate on generic statistical, model-based, and information-theoretic principles. This design is not tied to any particular feature semantics and therefore remains applicable to other flow-level IoT datasets.

The MDA aggregation inherits the same property, as its permutation-based formulation imposes no assumption on attack taxonomy. Extending the framework to multi-class detection requires only adjusting the classifier output, while the selection pipeline remains unchanged.

In terms of deployment, the 168.3 KB footprint remains well below typical edge memory budgets. This indicates feasibility on platforms beyond Raspberry Pi 4, and even on lower-tier devices such as the ESP32 or ARM Cortex-M.

V. CONCLUSIONS AND FUTURE WORK

This paper proposes an explainable ensemble-based feature selection framework that combines feature techniques through a multi-expert system module, then uses the MDA metric to select the most important features. Aiming for a balance between model performance and explanation ability, our proposed framework has shown significant improvements in experimental evaluation on the publicly available CIC IoT 2023 dataset. Experimental results show that by reducing the selection of critical features in the full dataset (6 features), system performance achieved impressive metrics (Accuracy of 98.33% and F1-score of 98.54%). This demonstrates the feasibility of deploying IDS on resource-constrained edge devices. The consistency between MDA-based rankings and posterior interpretations (SHAP/LIME, Spearman correlation coefficient = 0.87) indicates reliable explanatory power. Our next work will focus on deploying this proposed framework on real-world edge devices with different datasets. In the future, we will focus on deploying this proposed framework on real-world edge devices with different datasets, as well as extending the classification scope to multi-class attack scenarios. Additionally, lightweight federated learning will be studied to adapt to the strict memory and computational constraints of IoT systems.

ACKNOWLEDGEMENTS

This work was funded by the Australia–Vietnam Strategic Technologies Centre.

REFERENCES

[1] E. Dritsas and M. Trigka, "Federated learning for IoT: A survey of techniques, challenges, and applications," *Journal of Sensor and Actuator Networks*, vol. 14, no. 1, p. 9, 2025. DOI: 10.3390/jsan14010009.

[2] A. E. Akhdar, C. Baidada, and A. Kartit, "Study of Cyber Threats in IoT Systems," *International Conference on Data Analytics & Management*, pp. 329–344, 2023. DOI: 10.1007/978-981-99-4622-8-25.

[3] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet of Things Journal*, vol. 3, pp. 1–1, 2016. DOI: 10.1109/JIOT.2016.2579198.

[4] A. Gawade and N. M. Shekokar, "Impact of cyber security threats on IoT applications," *Cyber Security Threats and Challenges Facing Human Life*, pp. 71–80, 2022. DOI: 10.1201/9781003264637-6.

[5] A. Al-Zoubi, K. I. Sundus, and T. Kanan, "A comprehensive survey on intrusion detection systems in IoT: Challenges and future directions," in *2025 12th International Conference on Information Technology (ICIT)*, 2025, pp. 213–218.

[6] M. Fatima, O. Rehman, I. M. H. Rahman, A. Ajmal, and S. J. Park, "Towards ensemble feature selection for lightweight intrusion detection in resource-constrained IoT devices," *Future Internet*, vol. 16, no. 10, p. 368, 2024.

[7] A. G. Ayad, N. A. Sakr, and N. A. Hikal, "A Hybrid Feature Selection Model for Anomaly-Based Intrusion Detection in IoT Networks," *2024 International Telecommunications Conference (ITC-Egypt)*, pp. 1–7, 2024. DOI: 10.1109/ITC-Egypt61148.2024.10639908.

[8] J. Li, K. Cheng, S. Wang, F. Morstatter, R. Trevino, J. Tang, and H. Liu, "Feature selection: A data perspective," *ACM Computing Surveys*, vol. 50, 2016. DOI: 10.1145/3136625.

[9] W. Liu and J. Wang, "Recursive elimination current algorithms and a distributed computing scheme to accelerate wrapper feature selection," *Information Sciences*, vol. 589, pp. 636–654, Apr. 2022. DOI: 10.1016/j.ins.2021.12.086.

[10] G. Rathod, V. Sabnis, and J. K. Jain, "Improving IoT botnet attack detection using machine learning: Comparative analysis of feature selection methods and classifiers in intrusion detection systems," in *IEEE International Conference on Innovations in Converging Technologies (INOCON)*, 2024. DOI: 10.1109/inocon60754.2024.10511883.

[11] A. Arqane, O. Boutkhoum, H. Boukhriss, and A. Moutaouakkil, "Intrusion detection system using ensemble learning approaches: A systematic literature review," *International Journal of Online and Biomedical Engineering (iJOE)*, vol. 18, pp. 160–175, 2022. DOI: 10.3991/ijoe.v18i13.33519.

[12] C. Shand, R. Fong, and U. Butt, "How Explainable Artificial Intelligence (XAI) Models Can Be Used Within Intrusion Detection Systems (IDS) to Enhance an Analyst's Trust and Understanding," *International Conference on Global Security, Safety, and Sustainability*, pp. 321–342, 2023. DOI: 10.1007/978-3-031-40402-3-21.

[13] S. Neupane, J. Ables, W. Anderson, S. Mittal, S. Rahimi, I. Banicescu, and M. Seale, "Explainable intrusion detection systems (x-ids): A survey of current methods, challenges, and opportunities," *IEEE Access*, vol. 10, pp. 112392–112415, 2022. DOI: 10.1109/ACCESS.2022.3216199.
[14] J. Vergara and P. Estevez, "A review of feature selection methods based on mutual information," *Neural Computing and Applications*, vol. 24, 2014. DOI: 10.1007/s00521-013-1368-0.
[15] U. Njoku, B. Bilalli, A. Abelló, and G. Bontempi, "Wrapper methods for multi-objective feature selection," 2023. DOI: 10.48786/edbt.2023.58.
[16] M. Alqahtani, H. Mathkour, and M. M. Ben Ismail, "IoT botnet attack detection based on optimized extreme gradient boosting and feature selection," *Sensors*, vol. 20, p. 6336, 2020. DOI: 10.3390/s20216336.
[17] D. Dua, H. Singh, and M. R. Bhatnagar, "A hybrid ensemble learning model for intrusion detection systems in IoT networks," *IEEE Internet of Things Journal*, vol. 11, no. 3, pp. 2985–2998, 2024. DOI: 10.1109/JIOT.2023.3337445.
[18] B. Pes, "Ensemble feature selection for high-dimensional data: A stability analysis across multiple domains," *Neural Computing and Applications*, vol. 32, p. 5951, 2020. DOI: 10.1007/s00521-019-04093-2.
[19] P. Li et al., "Intrusion detection algorithm based on fusion feature selection in intelligent measurement systems," *Journal of Network Intelligence*, vol. 10, no. 2, pp. 145–158, 2025.
[20] S.-J. Lee and I.-G. Lee, "Lightweight federated learning-based intrusion detection system for industrial Internet of Things," *ICT Express*, vol. 11, 2025. DOI: 10.1016/j.icte.2025.05.002.
[21] S. Akintade, S. Kim, and K. Roy, "Explaining machine learning-based feature selection of IDS for IoT and CPS devices," in *Artificial Intelligence Applications and Innovations. AIAI 2023*, Cham: Springer, 2023. DOI: 10.1007/978-3-031-34107-6-6.
[22] A. F. Çelik et al., "Developing explainable intrusion detection systems for Internet of Things," in *16th Int. Conf. on Info. Security and Cryptology*, 2023, pp. 1–6.
[23] L. A. C. Ahakonye, A. A. Adeniran, and H. Chiroma, "Machine learning explainability for intrusion detection in the industrial Internet of Things," *IEEE Internet of Things Magazine*, vol. 7, no. 2, pp. 72–77, 2024. DOI: 10.1109/IOTM.001.2300171.
[24] O. Chakir, Y. Sadqi, and E. H. Ait Alaoui, "An explainable machine learning-based web attack detection system for industrial IoT web application security," *Info. Security Journal*, vol. 33, no. 2, pp. 121–135, 2024. DOI: 10.1080/19393555.2024.2362813.
[25] S. Saif, M. A. Hossain, and M. S. Islam, "IoT security fortification: Enhancing cyber threat detection through feature selection and advanced machine learning," in *ICIESTR*, Muscat, Oman, 2024, pp. 1–6. DOI: 10.1109/ICIESTR60916.2024.10798181.
[26] E. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu, and A. Ghorbani, "CICIoT2023: A real-time dataset and benchmark for large-scale attacks in IoT environment," *preprints*, 202305.0443.v1, 2023.
[27] A. P. White and W. Z. Liu, "Bias in information-based measures in decision tree induction," *Machine Learning*, vol. 15, no. 3, pp. 321–329, 1994. DOI: 10.1007/BF00993349.
[28] H. Wang, H. Wang, F. Yang, and Z. Luo, "An experimental study of the intrinsic stability of random forest variable importance measures," *BMC Bioinformatics*, vol. 17, no. 1, p. 60, 2016. DOI: 10.1186/S12859-016-0900-5.
[29] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001. DOI: 10.1023/A:1010933404324.
[30] A. Gregorutti, B. Michel, and P. Saint-Pierre, "Correlation and variable importance in random forests," *Statistics and Computing*, vol. 27, no. 3, pp. 659–678, 2017. DOI: 10.1007/s11222-016-9646-1.



Hoang Trong Minh (Senior Member, IEEE) received the bachelor's degree in physics engineering (1994) and electronic and telecommunication engineering (1999) from Hanoi University of Science and Technology, the master's degree in electronic and telecommunication engineering (2003), and the Ph.D.'s degree in telecommunication engineering (2014) from Posts and Telecommunications Institute of Technology. He is currently an associate professor in the Telecommunication Faculty of Posts and Telecommunications Institute of Technology. He is the head of the Telecommunications Network Department and the Intelligent Connected Networks Laboratory. He has been working on issues related to the performance of wireless networks. His research interests include network performance and security issues in edge computing, wireless mobile networks, and 5G/6G networks. He is a member of the IEEE, the IEEE Communications Society, the IEEE Circuits and Systems Society, the IEEE Electron Devices Society, and the IEEE Signal Processing Society.



Le Thi Trang Linh is curently a lecturer in the Faculty of Information Technology, Electric Power University, Viet Nam. She received an Enigeer's degree in information system (2009) from MUCTR (Mendeleev University of Chemical Technology of Russia) and Ph.D. in system analysis in Moscow in 2019 from MIPT (Moscow Institute of Physics and Technology). Her research focus on AI, Neural Networks, Information Security.



Nguyen Minh Hoang received B.E degree in Information and Communication Technology at University of Science and Technology Hanoi (USTH) in 2018 and a M.E in Computer Science from University of Engineering and Technology, Vietnam National University, in 2024. He is currently a research assistant at the Intelligent Connected Networks Laboratory (ICN Lab), Posts and Telecommunications Institute of Technology (PTIT). His research interests focus on Machine learning, deep learning, Intrusion Detection System, EdgeIoT, WSN.



Ngo Thi Thu Trang received her B.E degree of Telecommunications and Electronics Engineering from Vietnam National University, Hanoi (VNUH) in 2002 and M.E degree of Computer and Communication Engineering from Chungbuk National University, Korea in 2005 and PhD in Communication Engineering from Posts and Telecommunications Institute of Technology (PTIT) in 2021. Now, she is a lecturer in Telecommunications Faculty 1 of PTIT. Her research interests include optical communications, signal processing, IoT and sensor data processing in machine learning, and security.



Lam Sinh Cong received the Ph.D. degree in Engineering from University of Technology, Sydney, Australia, in 2018. He has been working with the University of Engineering and Technology, Vietnam National University, Hanoi since 2010 where he was promoted to an Associate Professor in 2024. His research interests focus on modeling, performance analysis and optimization for 5G and B5G, stochastic geometry model for wireless communications.