

Stability-Enhanced Feature Selection and Evolutionary Ensemble Learning: Addressing Adversarial Attacks in IoT Intrusion Detection

Waleed Almughalles*, Jianming Yong, Xiaohui Tao, Yuefeng Li, Jun Shen

Abstract The rapid growth of Internet of Things (IoT) networks has introduced significant challenges for machine learning-based intrusion detection systems (ML-based IDSs), particularly the threat of malicious feature injection through extensible data serialization formats. Such attacks can corrupt training data, compromise feature selection processes, and undermine system reliability. To address this issue, we propose a secure framework that applies recursive feature elimination with cross-validation (RFECV) across multiple models, followed by stability analysis to ensure robust feature selection. Additionally, we enhance the classification accuracy of ML-based IDSs through a weighted ensemble learning approach optimized using the Covariance Matrix Adaptation Evolution Strategy (CMA-ES). Extensive experiments on two benchmark IoT intrusion detection datasets, namely UNSW-NB15 and TON-IoT, demonstrate the effectiveness of the proposed framework in mitigating adversarial feature injection. The proposed approach outperforms state-of-the-art methods across multiple performance metrics, significantly improving the robustness and accuracy of ML-based IDSs.

Keywords Machine Learning (ML), Internet of Things (IoT), Intrusion Detection Systems (IDSs), Adversarial attacks, Feature selection, Ensemble learning

1 Introduction

The rapid proliferation of Internet of Things (IoT) networks has revolutionized industries such as healthcare, smart cities, and industrial automation by enabling real-time decision-making through interconnected devices. These networks gen-

erate vast amounts of data that are transmitted and processed to identify patterns and anomalies, thereby supporting operational efficiency and security. Machine learning-based intrusion detection systems (ML-based IDSs) have become critical components in safeguarding IoT systems against cyber threats. Data exchanged within these systems is commonly represented using serialization formats such as JSON, Protocol Buffers, and XML, which facilitate seamless data exchange between heterogeneous devices and support scalability and adaptability in dynamic IoT environments [1].

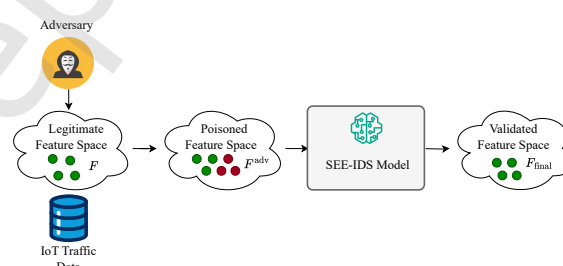


Fig. 1 Graphical abstract of the framework.

Although the extensibility of these serialization formats is essential for supporting the evolving requirements of IoT systems, it also introduces significant security vulnerabilities, particularly when secure implementation practices are lacking. Attackers can exploit this extensibility to inject malicious or adversarial features into serialized data payloads [2], [3], [4]. Once incorporated into the data streams processed by ML-based IDSs, these injected features can disrupt both training and inference processes, leading to data poisoning, false positives, false negatives, or even backdoor attacks. Such vulnerabilities not only compromise the reliability of IoT systems but also disrupt critical services, potentially resulting in financial losses, operational downtime, and diminished trust in IoT technologies.

Researchers have extensively studied adversarial attacks and feature-selection methods to improve the performance of ML-based IDSs [5], [6], [7]. Feature-selection techniques are widely employed to reduce dimensionality and enhance model efficiency by identifying relevant features and eliminating redundant or irrelevant ones [8]. Existing defense mechanisms

- Waleed Almughalles and Jianming Yong are with the School of Business, Law, Humanities and Pathways, University of Southern Queensland, Springfield Central, 4300, Australia. E-mail: {Waleed.Al-mughalles, Jianming.Yong}@unisq.edu.au
- Xiaohui Tao is with the School of Science, Engineering and Digital Technologies, University of Southern Queensland, Springfield Central, 4300, Australia. E-mail: Xiaohui.Tao@unisq.edu.au
- Yuefeng Li is with the School of Computer Science, Queensland University of Technology, Brisbane, 4000, Australia. E-mail: y2.li@qut.edu.au
- Jun Shen is with the School of Computing and Information Technology, University of Wollongong, Wollongong, 2522, Australia. E-mail: jshen@uow.edu.au

* To whom correspondence should be addressed.

Manuscript received: 2025-02-02; revised: 2026-01-10; accepted: 2026-06-01

against adversarial attacks, such as adversarial training and ensemble learning, often focus on scenarios involving data poisoning [9]. However, limited attention has been paid to feature injection attacks (FIAs) and their effects on feature-selection processes. This oversight leaves ML-based IDSs vulnerable to attacks that introduce adversarial features into input data, potentially rendering feature-selection methods ineffective or even counterproductive.

The persistence of this problem is compounded by the unique characteristics of IoT networks, in which devices are often resource-constrained and data collection is decentralized, making the implementation of robust data-validation mechanisms challenging. These factors create an environment in which malicious features can easily contaminate data pipelines and bypass traditional defenses. Furthermore, feature-selection methods typically assume that input data is clean and trustworthy, leaving them ill-equipped to filter out injected adversarial features that may ultimately be included in the final feature set. These challenges highlight the urgent need for research that bridges the gap between adversarial defenses and robust feature-selection methods, particularly in addressing malicious feature injection.

Accordingly, this study investigates the following research questions: (Q1) How can the robustness of feature selection under FIAs be improved? (Q2) How can the detection performance and reliability of ML-based IDSs be enhanced in adversarial IoT environments?

To address this research gap, this paper introduces Stability-Enhanced Ensemble-IDS (SEE-IDS), a secure framework that strengthens feature selection against FIAs while improving intrusion detection performance. The framework integrates robust feature selection with optimized ensemble learning to mitigate adversarial features and enhance classification reliability. An overview of SEE-IDS is presented in Fig. 1. The main contributions of this work are summarized as follows:

- (1) We propose a stability-enhanced feature-selection approach that combines recursive feature elimination with cross-validation (RFECV) and stability analysis to mitigate the risks posed by FIAs by identifying and excluding features that are not consistently selected across models and cross-validation folds.
- (2) We propose a weighted ensemble-learning approach to improve the resilience and accuracy of ML-based IDSs. This approach leverages the strengths of multiple base models while optimizing their contribution weights using an evolutionary optimization method.
- (3) We conduct extensive experiments using two benchmark IoT intrusion detection datasets, namely UNSW-

NB15 and TON-IoT. The proposed approach is evaluated against existing state-of-the-art methods using multiple performance metrics, demonstrating its effectiveness in mitigating FIAs and enhancing the reliability of ML-based IDSs.

The remainder of this paper is organized as follows: Section 2 reviews related studies and identifies the research gaps. Section 3 defines the research problem, and Section 4 presents the proposed framework. Section 5 describes the experimental setup, datasets, and evaluation metrics. Section 6 presents and discusses the results. Finally, Section 7 concludes the paper and outlines future research directions.

2 Related Work

2.1 Adversarial Attacks

Adversarial attacks threaten the integrity of ML-based IDSs, particularly in distributed IoT environments, by injecting malicious data to compromise classification accuracy and reduce detection rates. Several approaches have been developed to defend against label-flipping attacks, which involve changing the labels of training samples. Jiang et al. introduced MCDL, which detects malicious clients by evaluating data quality in a latent feature space while preserving privacy and computational efficiency [10]. Wang et al. proposed FENDER, a resilient distributed classification-learning scheme that integrates loss-resilient mechanisms and ensures algorithmic convergence without assuming symmetric label-flipping rates [11]. Zang et al. enhanced attack-detection sensitivity in federated learning (FL) by combining Kullback–Leibler divergence with a generative adversarial network (GAN)-based framework to preserve privacy [12]. Hallaji et al. employed adversarial training with noisy-label classifiers to mitigate both label-flipping and backdoor attacks [13], whereas Yang et al. developed a lightweight detection mechanism based on clustering the losses of local models [14]. Akram et al. proposed UAVGuard, a framework that protects graph neural networks (GNNs) against label-flipping attacks through community-preserving self-supervision [15].

Various studies have sought to counter backdoor attacks, which manipulate a model so that it misclassifies inputs containing a specific trigger while behaving normally on other inputs. For instance, Hou et al. proposed a federated defense against backdoor attacks in the Industrial Internet of Things (IIoT) that uses explainable artificial intelligence models to detect backdoor inputs through blur-label-flipping filters [16]. Alharbi et al. introduced collusive backdoor attacks (CBAs), which evade federated-learning defenses by manipulating model updates through perturbation estimation and the

Gram–Schmidt process [17]. Wang et al. proposed PerVK, a personalized federated-learning framework that combines virtual distillation and knowledge distillation to mitigate backdoor attacks in IoT environments [18]. Wang et al. also presented MITDBA, a robust algorithm that detects dynamic backdoor attacks using Gramian information and spectral signatures [19]. Zhang et al. developed a class activation mapping-deep neural network (CAM-DNN)-based backdoor attack against convolutional neural network (CNN)-based malware detection systems, using class activation mapping to generate triggers [20]. Yao et al. proposed Reverse Backdoor Distillation (RBD), an online method that uses a shadow model to detect backdoor attacks against deep neural networks [21].

Numerous studies have examined feature manipulation and data injection, which are attacks that modify existing feature values or introduce malicious data points into a training set. Gunawardena et al. demonstrated that altering as little as 6.2% of sensor data can significantly degrade system accuracy and proposed a defense that combines anomaly detection with adversarial training [22]. Alhussien et al. introduced domain-specific constraints for generating valid adversarial examples against IDSs and improved model robustness through adversarial training [23]. Ali et al. enhanced power-transformer fault diagnosis by introducing new gas-concentration features, thereby improving classification accuracy on IoT platforms [24]. Kong et al. introduced a dynamic coding scheme to detect stealthy false data injection (FDI) attacks in train–ground communication systems [25], whereas Ahmadi et al. proposed a scalable, data-driven mechanism for detecting FDI attacks in power systems with low computational cost [26]. Zhang et al. investigated adversarial attacks against deep learning-based network intrusion detection systems (NIDSs), focusing on feature manipulation under domain-specific constraints for generating adversarial samples. They evaluated five state-of-the-art attack strategies and proposed three robust defense mechanisms, including model ensembling and adversarial training, which improved detection rates and reduced attack success rates [27]. Rafiq et al. proposed a scalable ensemble-based model to counter adversarial attacks against IIoT malware-detection systems, in which fabricated feature manipulations cause misclassification. By training classifiers on distinct feature subsets, their approach achieved an accuracy of 91%, even under adversarial conditions involving modifications to as many as 14 features [28].

Existing adversarial defenses for ML-based IDSs effectively address label-flipping, backdoor, and feature-manipulation attacks through adversarial training and ensemble-based strate-

gies. However, these methods generally assume that the feature space is fixed and trustworthy. Consequently, attacks that introduce deceptive features by expanding the feature space remain insufficiently explored.

2.2 Feature Selection

Feature selection is crucial for enhancing the performance of ML-based IDSs in IoT environments by reducing dimensionality and improving detection accuracy. Eljaily et al. developed an IDS for software-defined networks (SDNs) in IoT environments, emphasizing the importance of feature selection for improving detection accuracy through multiple techniques [29]. Lefoane et al. proposed pattern-based selection methods that use dominant patterns, maximal frequent itemset mining, and Gini impurity to identify key features automatically [30]. Han et al. developed a binary particle swarm optimization framework named BPSWO, which integrates feature selection with model training to enhance the accuracy of power-system IDSs and uses chaotic transformation to prevent convergence issues [31]. Wang et al. employed spatiotemporal feature fusion for IoT attack detection by combining a CNN, a BiLSTM network, and an attention mechanism for effective feature learning [32]. Ghubaish et al. proposed LEMDA, a lightweight feature-engineering method designed to reduce dimensionality and computational complexity in IDSs [33]. He et al. developed BR-HIDE, which leverages multigranularity feature extraction and anti-sparse theory to manage high-dimensional system-call traces [34]. Sood et al. designed a two-stage anomaly-detection system for 5G networks that integrates dimensionality reduction with DNN classifiers [35]. Thakkar et al. applied statistical importance measures to DNN-based IDSs, ranking features according to standard deviation and mean–median differences [36]. Maseno et al. introduced a sequential feature-selection method for IoT IDSs using an optimized extreme learning machine (ELM) and an SVM classifier. A genetic algorithm (GA) optimizes the ELM parameters, which are then used in a forward-selection process to identify key features for SVM-based classification [37]. Alghanam et al. presented LS-PIO, an enhanced pigeon-inspired optimization (PIO) algorithm combined with a local-search mechanism for feature selection in IoT NIDSs. An ensemble of one-class classifiers was also integrated to further improve performance [38].

Stability analysis has also been employed to ensure the robustness of selected features against variations in the data. For example, Tu et al. proposed a method that improves RFECV by incorporating random sampling and regularization to enhance feature stability and address covariance issues in

regression tasks. However, their method does not address the challenges posed by adversarial settings, such as FIAs against IoT-based intrusion detection systems [39].

Although extensive research has examined adversarial attacks and feature-selection techniques independently, the intersection of these domains, particularly the effects of adversarial feature injection on the feature-selection process in ML-based IDSs, remains largely unaddressed. This oversight is particularly concerning in IoT networks, where resource constraints and decentralized data collection increase the risk of adversarial attacks infiltrating the feature-selection pipeline. To bridge this gap, we propose a framework that enhances the integrity of the feature-selection process and improves the classification accuracy of ML-based IDSs in adversarial settings.

3 Research Problem

The core problem addressed in this research is the vulnerability of ML-based IDSs to FIAs, an adversarial strategy that targets the feature space. In IoT networks, these attacks exploit the extensibility of data serialization formats to embed deceptive features into data payloads, thereby compromising the feature-selection process and introducing adversarial elements into the dataset. Such manipulation undermines model integrity and affects both training and inference.

To formally define the impact of FIAs on ML-based IDSs, let $D = \{(x_i, y_i) \mid i = 1, \dots, n\}$ represent the dataset used to train the model, where $x_i \in \mathbb{R}^m$ represents the legitimate feature vector of instance i , and y_i represents its corresponding label. The legitimate feature space is defined as $F = \mathbb{R}^m$, encompassing all possible feature vectors formed from m legitimate features.

In a typical ML-based IDS, the objective is to train a model M on D to classify instances accurately based on their legitimate features. However, under an FIA, an adversary expands the legitimate feature space F by introducing k additional adversarial features. This results in an expanded feature space, denoted by $F^{\text{adv}} = \mathbb{R}^{m+k}$.

Consequently, the original dataset is transformed into a poisoned dataset, $D^{\text{adv}} = \{(x_i^{\text{adv}}, y_i) \mid i = 1, \dots, n\}$, where $x_i^{\text{adv}} \in F^{\text{adv}}$ represents the poisoned feature vector of instance i , denoted by

$$x_i^{\text{adv}} = [x_i, z_i].$$

Here, $x_i \in F$ denotes the legitimate feature vector of instance i , whereas $z_i \in \mathbb{R}^k$ represents the adversarial feature vector injected by the attacker. These adversarial features are crafted to appear relevant, thereby deceiving conventional feature-selection methods into selecting them. Therefore, the

challenge lies in developing robust feature-selection methods that can distinguish legitimate features from adversarially injected features, protect ML-based IDSs against FIAs, and preserve their classification performance.

4 Framework

4.1 Conceptual Model

The proposed SEE-IDS framework aims to counter FIAs in ML-based IDSs by systematically improving feature selection and classification robustness. It consists of three stages, each addressing specific challenges posed by FIAs, as illustrated in Fig. 2. Stage 1 focuses on identifying relevant features while managing the increased dimensionality caused by FIAs. Stage 2 ensures robust feature selection by evaluating the consistency of feature importance across multiple data subsets and models. Finally, Stage 3 integrates a weighted ensemble-learning approach to optimize prediction accuracy while mitigating adversarial influence. Together, these stages provide a structured approach to improving IDS reliability under FIAs.

4.2 RFECV

The first stage of the framework addresses FIAs by systematically removing irrelevant and adversarially injected features from the feature space. This is achieved using RFECV approach, in which multiple models independently compute feature-importance scores. By exploiting the different sensitivities of these models to adversarial noise, the framework achieves a more robust feature-selection process.

Given the poisoned dataset D^{adv} , the RFECV stage aims to systematically reduce the feature space to a subset of legitimate features. The expanded feature vector $x_i^{\text{adv}} = [x_i, z_i]$ consists of $x_i \in F = \mathbb{R}^m$, which denotes the legitimate feature vector, and $z_i \in \mathbb{R}^k$, which denotes the adversarially injected feature vector. The objective is to retain features from the legitimate feature space F while eliminating irrelevant features, including adversarially injected features belonging to $F^{\text{adv}} \setminus F$.

The process begins by training each model M_j , where $j = 1, 2, \dots, J$, independently on the poisoned dataset D^{adv} . Each model computes a feature-importance score $I_j(f)$ for every feature $f \in F^{\text{adv}}$. For decision-tree-based models, feature importance is typically derived from impurity reduction, which measures the decrease in classification disorder when a feature is used to split a node. The importance score of feature f in model M_j is computed as

$$I_j(f) = \sum_{\tau \in \mathcal{T}_j} \sum_{v \in \mathcal{V}_\tau(f)} \Delta I_v(f), \quad (1)$$

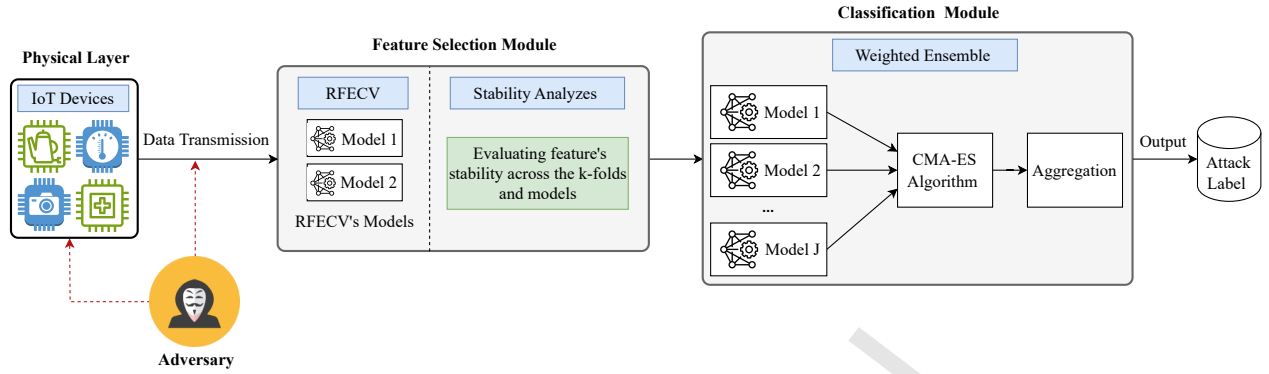


Fig. 2 SEE-IDS framework.

where $\mathcal{T}_j$ denotes the set of decision trees in model M_j , and $\mathcal{V}_\tau(f)$ denotes the set of nodes in tree τ at which feature f is used for splitting. The impurity reduction at node v is given by

$$\Delta I_v(f) = I(v) - \left(\frac{|v_L|}{|v|} I(v_L) + \frac{|v_R|}{|v|} I(v_R) \right), \quad (2)$$

where $I(v)$, $I(v_L)$, and $I(v_R)$ are the impurities of the parent, left-child, and right-child nodes, respectively. Moreover, $|v|$, $|v_L|$, and $|v_R|$ denote the numbers of samples reaching these nodes.

For decision-tree-based classification models, the impurity $I(v)$ can be defined using the Gini impurity:

$$I(v) = 1 - \sum_{c=1}^C P_{v,c}^2, \quad (3)$$

where $P_{v,c}$ is the proportion of samples belonging to class c at node v , and C is the total number of classes.

RFECV iteratively removes the least important features based on the importance scores $I_j(f)$. At iteration r , the feature subset considered by model M_j is denoted by

$$F_r^j \subseteq F^{\text{adv}}.$$

The feature with the lowest importance score is identified as

$$f_{r,j}^{\min} = \arg \min_{f \in F_r^j} I_j(f). \quad (4)$$

The identified feature is then removed, and the feature subset for the next iteration is updated as

$$F_{r+1}^j = F_r^j \setminus \{f_{r,j}^{\min}\}. \quad (5)$$

This process continues until a stopping criterion is met, such as reaching a minimum allowable number of features or observing a degradation in cross-validation performance.

To evaluate each candidate feature subset, RFECV incorporates K -fold cross-validation at every elimination iteration. For each feature subset F_r^j , model M_j is trained on $K - 1$ folds and evaluated on the remaining fold. This procedure is

repeated K times so that each fold serves as the validation fold once. The performance of model M_j using feature subset F_r^j is measured using an evaluation metric, such as accuracy or the F1-score. The average performance across all folds is computed as

$$\bar{\Psi}_j(F_r^j) = \frac{1}{K} \sum_{\ell=1}^K \Psi_{j,\ell}(F_r^j), \quad (6)$$

where $\Psi_{j,\ell}(F_r^j)$ represents the performance of model M_j on the ℓ -th validation fold using feature subset F_r^j .

The subset size that achieves the highest average cross-validation performance is selected as the optimal number of retained features for model M_j :

$$r_j^* = \arg \max_r \bar{\Psi}_j(F_r^j). \quad (7)$$

This criterion determines how many features model M_j should retain, ensuring that the selected subset size maximizes generalization performance across all validation folds.

At the end of the RFECV process, each model M_j produces a selected feature subset for each cross-validation fold, denoted by $F_{\text{RFECV}}^{(j,\ell)} \subseteq F^{\text{adv}}$, where $\ell = 1, 2, \dots, K$, each containing the r_j^* most important features identified in the corresponding fold. These subsets may differ across folds and models because the models can retain or discard particular features depending on their sensitivity to adversarial noise and the composition of the training-data partitions. Although RFECV reduces the feature space by eliminating many irrelevant features, some adversarial features may persist because of inconsistencies in feature rankings across models and cross-validation folds. To address this limitation, the subsequent stability-analysis stage strengthens the feature-selection process by promoting selection consistency across diverse models and data subsets.

Algorithm 1 Multi-Model RFECV Feature Selection

Input: Poisoned dataset D^{adv} , base models $\{M_j\}_{j=1}^J$, number of folds K .

Output: $\{F_{\text{RFECV}}^{(j,\ell)}\}$.

```

1: for each model  $M_j, j = 1, \dots, J$  do
2:   for each fold  $\ell = 1, \dots, K$  do
3:     Initialize  $r \leftarrow 1, F_1^j \leftarrow F^{\text{adv}}$ 
4:     while stopping criterion not met do
5:       Train  $M_j$  on the  $K-1$  training folds using  $F_r^j$ 
6:       Compute  $I_j(f)$  for all  $f \in F_r^j$ 
7:       Evaluate  $\Psi_{j,\ell}(F_r^j)$  on validation fold  $\ell$ 
8:        $f_{r,j}^{\min} \leftarrow \arg \min_{f \in F_r^j} I_j(f)$ 
9:        $F_{r+1}^j \leftarrow F_r^j \setminus \{f_{r,j}^{\min}\}; r \leftarrow r + 1$ 
10:    end while
11:  end for
12:  Compute  $\bar{\Psi}_j(F_r^j)$  for each subset size
13:  Select optimal size  $r^*$  maximizing  $\bar{\Psi}_j$ ; set  $F_{\text{RFECV}}^{(j,\ell)}$  to
    the size- $r^*$  subset from fold  $\ell$ 
14: end for
15: Return  $\{F_{\text{RFECV}}^{(j,\ell)}\}_{j=1, \ell=1}^{J,K}$ 

```

4.3 Stability Analysis

The second stage of the framework enhances the robustness of feature selection through a two-level stability analysis conducted across cross-validation folds and models. This stage improves resilience to adversarial noise and complements the RFECV stage in adversarial settings.

For each model M_j , the RFECV process produces a selected feature subset for each cross-validation fold ℓ , denoted by

$$F_{\text{RFECV}}^{(j,\ell)} \subseteq F^{\text{adv}}, \quad \ell = 1, 2, \dots, K.$$

These subsets may vary across folds because of differences in the corresponding training-data partitions. To quantify the stability of feature selection within each model, the fold-level selection frequency of each feature $f \in F^{\text{adv}}$ is computed as

$$\phi_j(f) = \sum_{\ell=1}^K \mathbb{I}(f \in F_{\text{RFECV}}^{(j,\ell)}), \quad (8)$$

where $\mathbb{I}(\cdot)$ is an indicator function defined as

$$\mathbb{I}(f \in F_{\text{RFECV}}^{(j,\ell)}) = \begin{cases} 1, & \text{if } f \in F_{\text{RFECV}}^{(j,\ell)}, \\ 0, & \text{otherwise.} \end{cases} \quad (9)$$

A fold-level stability threshold θ_{fold} is then applied to determine whether a feature is stable within model M_j . The stable feature set produced by model M_j is defined as

$$F_{\text{stable}}^j = \{f \in F^{\text{adv}} \mid \phi_j(f) \geq \theta_{\text{fold}}\}. \quad (10)$$

Here, θ_{fold} , where $1 \leq \theta_{\text{fold}} \leq K$, represents the minimum

number of folds in which a feature must be selected to be considered stable for model M_j . For example, when $K = 5$ and $\theta_{\text{fold}} = 5$, a feature must be selected in all five folds to satisfy the fold-level stability criterion.

Because different models, such as Random Forest, XGBoost, and Decision Tree, may exhibit different sensitivities to adversarial noise, their stable feature sets F_{stable}^j may differ. To further enhance robustness, the framework evaluates feature stability across all J models by aggregating their stable feature sets. The model-level selection frequency of feature f is computed as

$$\phi(f) = \sum_{j=1}^J \mathbb{I}(f \in F_{\text{stable}}^j). \quad (11)$$

Here, J is the total number of models, and the indicator function equals 1 when feature f belongs to the stable feature set of model M_j , and 0 otherwise.

A feature is considered stable across models if it satisfies the model-level stability threshold θ_{model} . Accordingly, the final stable feature set is defined as

$$F_{\text{final}} = \{f \in F^{\text{adv}} \mid \phi(f) \geq \theta_{\text{model}}\}, \quad (12)$$

where $1 \leq \theta_{\text{model}} \leq J$. This criterion ensures that only features consistently retained by a sufficient number of models are included in the final feature set. For example, when $\theta_{\text{model}} = 2$, a feature retained by both XGBoost and Decision Tree satisfies the model-level stability criterion, whereas a feature retained by only one model is excluded.

The stability-analysis stage produces a refined feature set, $F_{\text{final}} \subseteq F^{\text{adv}}$, whose features satisfy the specified consistency requirements across both cross-validation folds and models. By excluding features that are selected inconsistently, this stage aims to reduce the presence of adversarially injected and unstable features. The resulting feature set is subsequently used to train the final classification models in the next stage of the framework.

4.4 Weighted Ensemble Learning

Building on the stability-enhanced feature set F_{final} produced by the previous stage, the framework applies a weighted ensemble-learning approach to improve detection performance and reduce the residual influence of any adversarial features that may persist after feature selection. Multiple base models are independently trained using only the features in F_{final} , and their predicted class probabilities are combined through an optimized weighting strategy that balances the contributions of the diverse models.

Given the set of J base models, $\mathcal{M} = \{M_j \mid j = 1, \dots, J\}$, each model M_j is trained on the training data

Algorithm 2 Stability Analysis Across Folds and Models

Input: RFECV feature subsets $\{F_{\text{RFECV}}^{(j,\ell)}\}$, Thresholds $\theta_{\text{fold}}, \theta_{\text{model}}$.

Output: Final stable feature set F_{final} .

- 1: **for** each model $M_j, j = 1, \dots, J$ **do**
- 2: Compute fold-level selection frequency $\phi_j(f)$ for all $f \in F^{\text{adv}}$
- 3: $F_{\text{stable}}^j \leftarrow \{f \in F^{\text{adv}} \mid \phi_j(f) \geq \theta_{\text{fold}}\}$
- 4: **end for**
- 5: Compute model-level selection frequency $\phi(f)$ for all $f \in F^{\text{adv}}$
- 6: $F_{\text{final}} \leftarrow \{f \in F^{\text{adv}} \mid \phi(f) \geq \theta_{\text{model}}\}$
- 7: **Return** F_{final}

represented by the stable feature set F_{final} . For an input instance x , model M_j produces a probability distribution over the C classes:

$$\mathbf{P}_j(x) = [P_{j,1}(x), P_{j,2}(x), \dots, P_{j,C}(x)], \quad (13)$$

where $P_{j,c}(x)$ represents the probability assigned by model M_j to class c .

The ensemble combines the probability distributions produced by the base models using a weighted sum. The aggregated probability distribution for instance x is defined as

$$\mathbf{P}_{\text{ens}}(x; \mathbf{w}) = \sum_{j=1}^J w_j \mathbf{P}_j(x), \quad (14)$$

where

$$\mathbf{w} = [w_1, w_2, \dots, w_J]^\top$$

is the ensemble weight vector. The weights are subject to the constraints

$$w_j \geq 0, \quad \sum_{j=1}^J w_j = 1, \quad (15)$$

which ensure that the aggregated output remains a valid probability distribution.

The predicted class label for instance x is determined by the class with the highest aggregated probability:

$$\hat{y}(x; \mathbf{w}) = \arg \max_{c \in \{1, \dots, C\}} P_{\text{ens},c}(x; \mathbf{w}), \quad (16)$$

where $P_{\text{ens},c}(x; \mathbf{w})$ denotes the ensemble probability assigned to class c .

Because the predictive performance of the ensemble depends directly on the weight vector $\mathbf{w}$, the framework optimizes $\mathbf{w}$ using the covariance matrix adaptation evolution strategy (CMA-ES), an evolutionary optimization algorithm

suitable for nonlinear and nonconvex objective functions [40].

At iteration t , CMA-ES maintains a multivariate normal search distribution characterized by a mean vector $\mathbf{m}_t$, a covariance matrix $\mathbf{C}_t$, and a global step size σ_t . A population of λ candidate vectors is sampled as

$$\tilde{\mathbf{w}}_i^{(t)} \sim \mathcal{N}(\mathbf{m}_t, \sigma_t^2 \mathbf{C}_t), \quad i = 1, \dots, \lambda, \quad (17)$$

where $\tilde{\mathbf{w}}_i^{(t)} \in \mathbb{R}^J$ is the unconstrained candidate vector generated for candidate i .

Because the sampled components may be negative and may not sum to one, each candidate is transformed into a valid ensemble weight vector using the softmax function:

$$w_{i,j}^{(t)} = \frac{\exp(\tilde{w}_{i,j}^{(t)})}{\sum_{j'=1}^J \exp(\tilde{w}_{i,j'}^{(t)})}, \quad j = 1, \dots, J. \quad (18)$$

This transformation guarantees that

$$w_{i,j}^{(t)} > 0 \quad \text{and} \quad \sum_{j=1}^J w_{i,j}^{(t)} = 1.$$

Each candidate weight vector

$$\mathbf{w}_i^{(t)} = [w_{i,1}^{(t)}, \dots, w_{i,J}^{(t)}]^\top$$

is evaluated using a fitness function that measures the predictive performance of the resulting ensemble on validation data. Using K -fold cross-validation, the average fitness of a candidate weight vector $\mathbf{w}$ is defined as

$$\Psi(\mathbf{w}) = \frac{1}{K} \sum_{\ell=1}^K \Psi_\ell(\mathbf{w}), \quad (19)$$

where $\Psi_\ell(\mathbf{w})$ represents the selected performance metric, such as accuracy, macro-averaged F1-score, or macro-averaged recall, obtained on the ℓ -th validation fold.

Because CMA-ES is commonly formulated as a minimization algorithm, the optimization objective can be expressed as

$$\mathcal{L}(\mathbf{w}) = -\Psi(\mathbf{w}), \quad (20)$$

such that minimizing $\mathcal{L}(\mathbf{w})$ is equivalent to maximizing the ensemble performance $\Psi(\mathbf{w})$.

After evaluating all candidates, the μ best-performing candidates are selected as elite solutions. Let

$$\tilde{\mathbf{w}}_{1:\lambda}^{(t)}, \dots, \tilde{\mathbf{w}}_{\mu:\lambda}^{(t)}$$

denote the elite candidates ordered according to their fitness values. The mean of the search distribution is updated through weighted recombination:

$$\mathbf{m}_{t+1} = \sum_{i=1}^{\mu} a_i \tilde{\mathbf{w}}_{i:\lambda}^{(t)}, \quad (21)$$

where a_i is the recombination coefficient assigned to elite candidate i , satisfying

$$a_i > 0, \quad \sum_{i=1}^{\mu} a_i = 1. \quad (22)$$

In simplified form, the covariance matrix is updated based on the distribution of the elite candidates:

$$\mathbf{C}_{t+1} = \sum_{i=1}^{\mu} a_i \left(\tilde{\mathbf{w}}_{i:\lambda}^{(t)} - \mathbf{m}_{t+1} \right) \left(\tilde{\mathbf{w}}_{i:\lambda}^{(t)} - \mathbf{m}_{t+1} \right)^{\top}. \quad (23)$$

The step size σ_t and covariance matrix $\mathbf{C}_t$ are adaptively adjusted over successive iterations, allowing CMA-ES to explore different regions of the search space while progressively focusing on promising candidate solutions.

The optimization process continues until a stopping criterion is satisfied, such as reaching a predefined maximum number of iterations, observing negligible improvement in the fitness value, or detecting sufficient convergence of the search distribution. The best candidate obtained during the optimization process is selected as the optimal ensemble weight vector:

$$\mathbf{w}^* = \arg \max_{\mathbf{w} \in \mathcal{W}} \Psi(\mathbf{w}), \quad (24)$$

where the feasible weight space is

$$\mathcal{W} = \left\{ \mathbf{w} \in \mathbb{R}^J \mid w_j \geq 0, \sum_{j=1}^J w_j = 1 \right\}. \quad (25)$$

Using the optimized weights, the final aggregated probability assigned to class c is computed as

$$P_{\text{ens},c}(x; \mathbf{w}^*) = \sum_{j=1}^J w_j^* P_{j,c}(x). \quad (26)$$

The final class prediction is therefore given by

$$\hat{y}^*(x) = \arg \max_{c \in \{1, \dots, C\}} P_{\text{ens},c}(x; \mathbf{w}^*). \quad (27)$$

By training the base models on the stability-enhanced feature set and combining their outputs through CMA-ES-optimized weights, the framework exploits the complementary strengths of the base models: the refined feature set limits the presence of inconsistent and adversarially injected features, while the optimized weighting strategy assigns greater influence to models that achieve stronger validation performance. Together, these stages aim to improve classification robustness and predictive performance across diverse IoT network conditions.

4.5 Integrated SEE-IDS Pipeline

The proposed framework provides a comprehensive solution for mitigating the effects of FIAs by integrating RFECV-based feature elimination, stability analysis, and weighted ensemble

Algorithm 3 Weighted Ensemble Learning via CMA-ES

Input: Stable feature set F_{final} , base models $\{M_j\}_{j=1}^J$, number of folds K , population size λ , elite size μ .

Output: Optimal ensemble weight vector $\mathbf{w}^*$.

- 1: Train each model M_j using the features in F_{final}
 - 2: Initialize CMA-ES parameters $\mathbf{m}_0, \mathbf{C}_0, \sigma_0$; set $t \leftarrow 0$
 - 3: **while** stopping criterion not met **do**
 - 4: Sample λ candidates $\tilde{\mathbf{w}}_i^{(t)} \sim \mathcal{N}(\mathbf{m}_t, \sigma_t^2 \mathbf{C}_t)$
 - 5: Transform each candidate into a valid weight vector $\mathbf{w}_i^{(t)}$ via softmax
 - 6: Evaluate fitness $\Psi(\mathbf{w}_i^{(t)})$ via K -fold cross-validation
 - 7: Select the μ best candidates as elite solutions
 - 8: Update $\mathbf{m}_{t+1}, \mathbf{C}_{t+1}$, and σ_{t+1} ; set $t \leftarrow t + 1$
 - 9: **end while**
 - 10: $\mathbf{w}^* \leftarrow$ best weight vector found during optimization
 - 11: **Return** $\mathbf{w}^*$
-

learning into a unified pipeline. RFECV is first employed to control dimensionality growth by removing irrelevant features (Algorithm 1). Stability analysis then identifies features that remain consistently relevant across cross validation folds and models, thereby filtering out adversarial noise (Algorithm 2). Building on the resulting stability-enhanced feature set, the framework applies a weighted ensemble-learning strategy in which CMA-ES optimizes the model weights to effectively combine the contributions of diverse base models (Algorithm 3). Collectively, this pipeline enhances the robustness of IDSs against FIAs while improving predictive reliability.

5 Evaluation

5.1 Experiment Design

Empirical experiments are conducted to validate the effectiveness of SEE-IDS under controlled and reproducible conditions using benchmark intrusion-detection datasets. The experiments address the following questions: (Q1) How effectively does SEE-IDS improve the robustness of feature selection against FIAs compared with existing feature-selection methods? (Q2) How effectively does SEE-IDS improve intrusion-detection performance and reliability under adversarial feature-injection scenarios?

The experimental workflow consists of four stages. First, 30 adversarial features generated using random noise, correlated noise, and nonlinear transformations are injected into each dataset. Second, the resulting datasets are preprocessed through missing-value imputation, categorical-feature encoding, and numerical-feature standardization. Third, the preprocessed data are supplied to SEE-IDS to obtain the selected feature subset and train the optimized weighted

ensemble. Finally, the resulting feature-selection and detection outcomes are evaluated and compared with those of the baseline methods.

5.2 Datasets

The proposed framework was evaluated using two benchmark IoT intrusion detection datasets: UNSW-NB15 [41] and TON-IoT [42]. Both datasets are widely recognized for their quality and diversity, as they contain a comprehensive range of normal traffic and various types of cyberattacks, enabling robust testing in realistic IoT environments. However, these datasets exhibit significant class imbalance, with certain attack classes being underrepresented relative to others. This imbalance poses a challenge for classification models and highlights the need for ensemble-learning techniques to improve predictive performance, particularly in multiclass classification tasks. Key statistics and characteristics of the datasets, including the numbers of records and features and their class distributions, are summarized in Tables 1 and 2.

Table 1 UNSW-NB15 dataset.

Class	Count	Percentage
Normal	93000	36.09%
Generic	58871	22.85%
Exploits	44525	17.28%
Fuzzers	24246	9.41%
DoS	16353	6.35%
Reconnaissance	13987	5.43%
Analysis	2677	1.04%
Backdoor	2329	0.90%
Shellcode	1511	0.59%
Worms	174	0.07%
Total	257,673	100%

Table 2 TON-IoT dataset.

Class	Count	Percentage
Normal	30000	33.29%
DoS	10000	11.10%
DDoS	10000	11.10%
Injection	10000	11.10%
Password	10000	11.10%
Scanning	10000	11.10%
Xss	10000	11.10%
Mitm	112	0.12%
Total	90,112	100%

5.3 Baselines

To evaluate the performance of the proposed framework, we compare it with a vanilla model and several baseline methods. The vanilla model represents a simplified approach that highlights the limitations addressed by the proposed framework,

whereas the benchmark methods represent state-of-the-art techniques used to assess its competitive performance.

- **Vanilla Model:** A conventional model trained on the full feature set without feature selection or ensemble learning. This baseline reveals the model's vulnerability to FIAs and highlights the need for robust defense mechanisms.
- **SBFA [29]:** Applies score-based multi-feature selection by aggregating multiple filter and wrapper methods, followed by classification using an individual model.
- **TIKI-TAKA [27]:** Employs ensemble learning as a defense strategy against adversarial attacks. MLP, CNN, and C-LSTM classifiers are jointly aggregated using unanimous hard voting to produce the final decision.
- **FS-OR-Ensemble [28]:** Employs an ensemble model as a defense against evasion attacks. Multiple SVM classifiers, each trained on a different feature subset, are aggregated at the decision level through hard voting based on a logical OR operation.

5.4 Performance Evaluation Metrics

To evaluate the framework's performance, we used four key metrics: accuracy, precision, recall, and F1-score. Accuracy provides an overall measure of predictive performance, while precision and recall assess false-positive control and true-positive detection, respectively. The F1-score balances precision and recall, making it particularly suitable for evaluating performance on imbalanced datasets. Collectively, these metrics assess the framework's effectiveness in achieving robust and reliable intrusion detection in adversarial environments.

6 Experimental Result and Analysis

6.1 Experimental Result

The proposed framework was evaluated using two benchmark datasets, namely UNSW-NB15 and TON-IoT, across three primary stages: RFECV, stability analysis, and weighted ensemble learning.

The UNSW-NB15 dataset consisted of 81 features, including 51 legitimate features and 30 adversarially injected features. The XGB-RFECV model successfully eliminated all adversarially injected features and retained 30 legitimate features. In contrast, the DT-RFECV and ET-RFECV models retained all 81 features, including the adversarially injected features, as shown in Fig. 3.

The TON-IoT dataset contained 52 features, including 22 legitimate features and 30 adversarially injected features. The XGB-RFECV model retained 30 features, comprising 20

legitimate features and 10 adversarially injected features. The DT-RFECV model retained 15 features, including 11 legitimate features and 4 adversarially injected features, whereas the ET-RFECV model retained only one legitimate feature, as shown in Fig. 4.

The stability analysis evaluated the consistency of feature selection across models and cross-validation folds. For UNSW-NB15, Fig. 5 illustrates the number of folds in which each feature was selected by the three models during RFECV. Similarly, for TON-IoT, Fig. 6 shows the number of folds in which each feature was selected by the models during RFECV.

After applying the predefined stability thresholds, namely

$$\theta_{\text{fold}} = 5 \quad \text{and} \quad \theta_{\text{model}} = 3$$

for UNSW-NB15, and

$$\theta_{\text{fold}} = 5 \quad \text{and} \quad \theta_{\text{model}} = 2$$

for TON-IoT, the final feature sets contained 6 stable features for UNSW-NB15, as shown in Fig. 7, and 10 stable features for TON-IoT, as shown in Fig. 8. In both datasets, all adversarially injected features were successfully removed.

These findings answer Q1 by demonstrating that RFECV alone is insufficient to consistently eliminate adversarial features across different models. Integrating stability analysis with RFECV improves the robustness of feature selection by enforcing consistency across models and cross-validation folds, thereby filtering out adversarial features that RFECV alone cannot consistently eliminate.

The final stage involved training and testing the weighted ensemble model using the optimized model weights. For the UNSW-NB15 dataset, the proposed framework achieved an accuracy of 0.8364, a weighted F1-score of 0.8191, and a macro-F1 score of 0.6317. In comparison, the vanilla model and benchmark methods achieved lower performance, as reported in Table 3.

For the TON-IoT dataset, the proposed framework also outperformed the vanilla model and benchmark methods, achieving an accuracy of 0.9370, a weighted F1-score of 0.9362, and a macro-F1 score of 0.9132, as reported in Table 4.

These results answer Q2 by demonstrating that the proposed weighted ensemble-learning approach, optimized using CMA-ES, consistently outperforms the vanilla model and benchmark methods across all evaluation metrics. By optimizing the relative contributions of the base models, the proposed weight-allocation strategy improves predictive performance under adversarial conditions.

6.2 Discussion

The observed feature-selection behavior varies across datasets because of differences in traffic characteristics and feature distributions. In the UNSW-NB15 dataset (Fig. 3), XGB-based RFECV eliminates the injected irrelevant features because its gradient-boosting mechanism can model complex feature interactions and suppress adversarial noise. In contrast, DT- and ET-based RFECV are more sensitive to noisy feature perturbations in this setting, leading to the retention of injected features when these models are used independently.

For the TON-IoT dataset (Fig. 4), the feature-selection dynamics differ because of heterogeneous, multisource data collection across IoT/IIoT devices and system layers, which results in more diverse feature distributions. This diversity enables DT-based RFECV to isolate injected features more effectively, whereas XGB-based RFECV removes only a subset of them and ET-based RFECV becomes overly conservative. These dataset-dependent behaviors indicate that no single feature-selection method provides a consistently robust defense against FIAs.

To address this variability, SEE-IDS avoids relying on any individual model. Instead, robustness is achieved by enforcing feature stability across both cross-validation folds and heterogeneous learners. As illustrated in Figs. 5 and 6, substantial variability exists in feature selection across folds and models, motivating the introduction of fold-level and model-level stability thresholds. By enforcing these stability constraints, as shown in Figs. 7 and 8, SEE-IDS retains only consistently informative features while filtering out unstable and adversarial ones. This stability-driven design enables robust feature selection across diverse data partitions and attack manifestations, thereby reinforcing the effectiveness of multi-model stability analysis as a defense against FIAs.

Building on the stability-enhanced feature set, the weighted ensemble-learning stage further improves predictive performance in the presence of adversarial features. By optimizing model weights using CMA-ES, SEE-IDS leverages the complementary strengths of the individual classifiers while suppressing the influence of weaker or less robust models. This optimization contributes to the consistently superior performance of SEE-IDS in terms of accuracy, precision, recall, and F1-score compared with the vanilla model and benchmark methods, as reported in Tables 3 and 4.

Although CMA-ES is generally considered computationally intensive, its overhead in SEE-IDS is limited by design. The optimizer operates in a low-dimensional search space, tuning only a small set of ensemble weights, and is executed offline after the base models have been trained. Therefore, it

Table 3 Performance comparison on UNSW-NB15 dataset.

Method	Accuracy	W-Precision	W-Recall	W-F1-Score	M-Precision	M-Recall	M-F1-Score
SEE-IDS	0.8327	0.8341	0.8327	0.8166	0.7843	0.6069	0.6317
Vanilla Model	0.7802	0.7805	0.7802	0.7804	0.5556	0.5598	0.5576
SBFA [29]	0.7821	0.7820	0.7821	0.7821	0.5562	0.5570	0.5565
TIKI-TAKA [27]	0.8147	0.8146	0.8147	0.8114	0.6707	0.5733	0.6010
FS-OR-Ensemble [28]	0.8153	0.8145	0.8153	0.8114	0.6809	0.5718	0.6017

Table 4 Performance comparison on TON-IoT dataset.

Method	Accuracy	W-Precision	W-Recall	W-F1-Score	M-Precision	M-Recall	M-F1-Score
SEE-IDS	0.9370	0.9405	0.9370	0.9362	0.9245	0.9081	0.9132
Vanilla Model	0.9095	0.9095	0.9095	0.9095	0.8439	0.8385	0.8411
SBFA [29]	0.9146	0.9144	0.9146	0.9145	0.8771	0.8643	0.8703
TIKI-TAKA [27]	0.9147	0.9140	0.9147	0.9140	0.9110	0.8194	0.8419
FS-OR-Ensemble [28]	0.9194	0.9187	0.9194	0.9189	0.8971	0.8521	0.8694

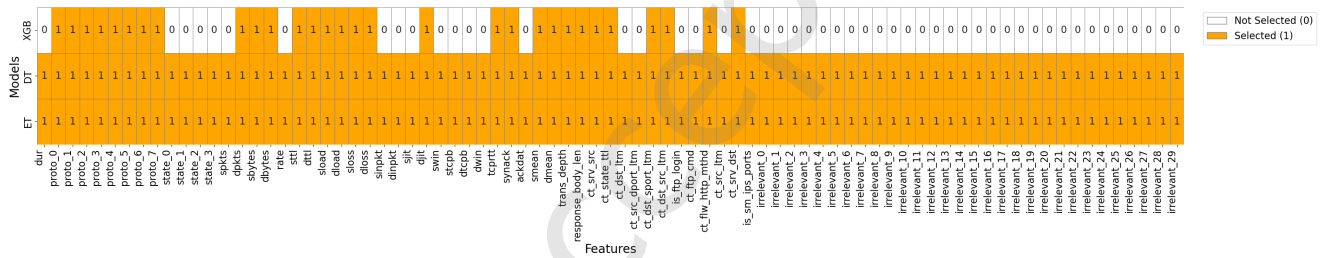


Fig. 3 Feature selection results using RFECV across models on the UNSW-NB15.

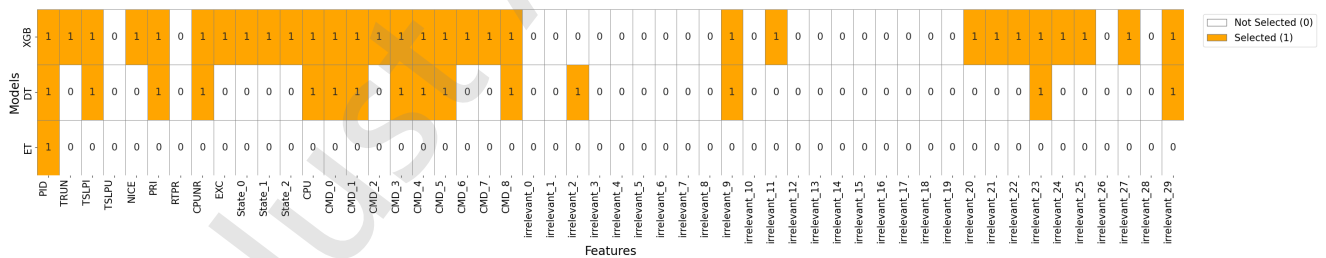


Fig. 4 Feature selection results using RFECV across models on the TON-IoT.

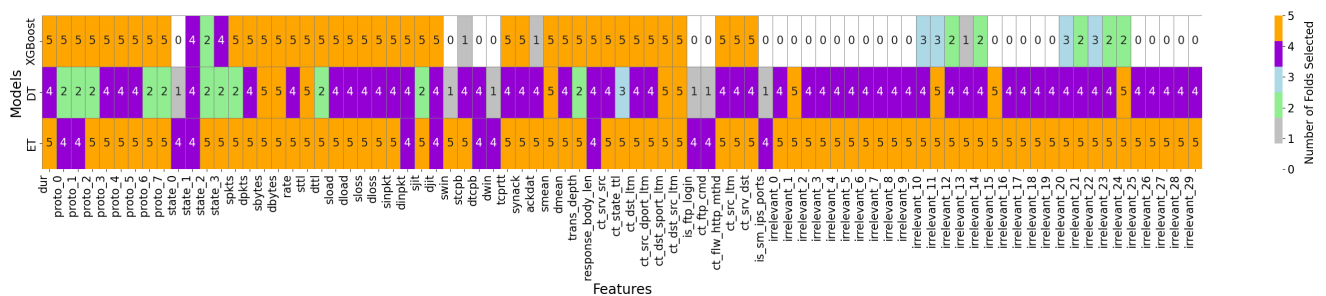


Fig. 5 Feature stability across models and folds on UNSW-NB15.

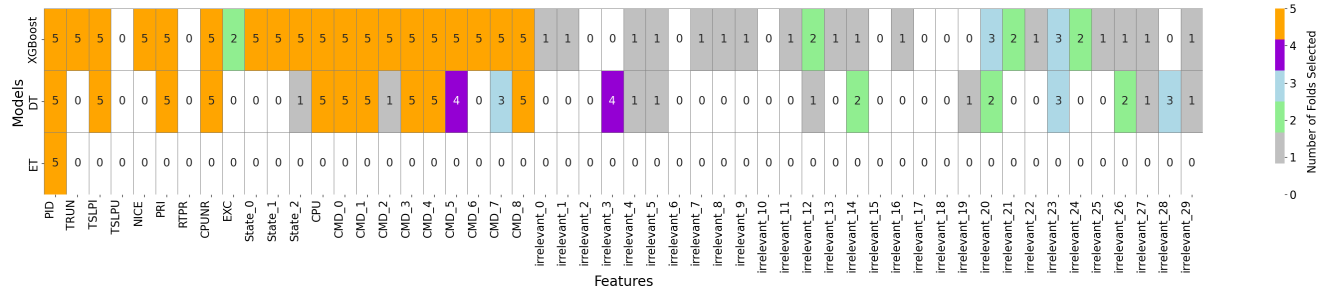


Fig. 6 Feature stability across models and folds on TON-IoT.

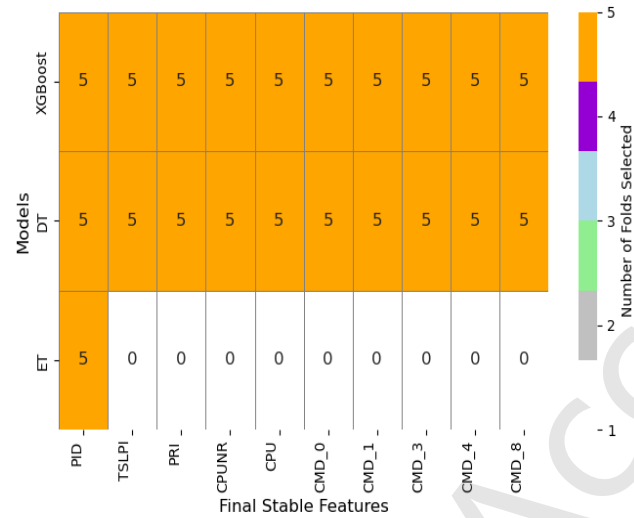


Fig. 7 Final feature set across models and folds on TON-IoT (after thresholding).



Fig. 8 Final feature set across models and folds on UNSW-NB15 (after thresholding).

introduces no additional computational cost during inference. Consequently, the added computational expense is modest relative to that of feature selection and model training, representing a practical compromise for improving robustness and predictive reliability under FIAs.

Although the absolute improvement in the weighted F1-score on UNSW-NB15 appears modest relative to the TIKI-TAKA and FS-OR-Ensemble benchmarks, this metric is dominated by majority classes and is therefore less sensitive to improvements in minority attack classes, which are particularly vulnerable to FIAs. SEE-IDS demonstrates a clearer advantage in macro-averaged performance, increasing the macro-F1 score from approximately 0.60 to 0.63 and thereby reflecting more consistent per-class robustness. This improvement can be attributed to the explicit mitigation of FIAs through stability-enhanced RFECV, which removes unstable and adversarially injected features before classification. In contrast, the benchmark methods primarily rely on hard-voting-based decision-level aggregation without enforcing feature stability. In a high-accuracy IDS setting, such macro-level gains indicate a substantive improvement in robustness rather than merely incremental classifier tuning.

The vanilla model, which does not incorporate feature selection or ensemble learning, is highly susceptible to FIAs, as reflected in its inferior performance across all evaluation metrics. The SBFA benchmark, which applies feature selection without stability analysis, achieves limited improvements but remains vulnerable because of inconsistent feature rankings under adversarial perturbations. The TIKI-TAKA benchmark relies on ensemble learning with unanimous hard voting, which improves decision robustness but offers no protection against feature-space expansion. The FS-OR-Ensemble benchmark combines feature-subset separation with hard-voting-based aggregation, providing moderate improvements while still lacking a mechanism for enforcing feature stability.

By jointly integrating stability-enhanced feature selection and weighted ensemble learning, SEE-IDS systematically removes adversarial features, optimizes model contributions,

and delivers consistent robustness across diverse scenarios. These results validate the effectiveness of SEE-IDS in addressing the challenges posed by FIAs.

7 Conclusion

This research proposed a robust framework, named SEE-IDS, to address FIAs in ML-based IDSs for IoT networks by integrating stability-enhanced RFECV with weighted ensemble learning optimized using CMA-ES. Experimental results obtained using the UNSW-NB15 and TON-IoT datasets demonstrated the effectiveness of the framework, which outperformed all benchmark methods. Stability analysis, based on a two-level evaluation across cross-validation folds and models, played a critical role in enhancing feature selection by reducing the retention of adversarially injected features. The optimized ensemble-learning approach further improved classification performance and resilience to adversarial noise. These findings demonstrate the ability of SEE-IDS to enhance the reliability of ML-based IDSs in adversarial IoT environments and provide a robust defense against FIAs. Future work will investigate hybrid defense strategies, such as combining adversarial training with stability analysis, to counter more complex attack vectors in IoT environments.

Acknowledgment

This study is partially funded by the Australian Research Council (ARC) through the Discovery Project scheme (Project ID: DP220101360).

Reference

- [1] A. H. Ngu, M. Gutierrez, V. Metsis, S. Nepal, and Q. Z. Sheng, Iot middleware: A survey on issues and enabling technologies, *IEEE Internet of Things Journal*, vol. 4, no. 1, pp. 1–20, 2016.
- [2] A. Fábrega, C. O. Pérez, A. Namavari, B. Nassi, R. Agarwal, and T. Ristenpart, Injection attacks against end-to-end encrypted applications, in *2024 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2024, pp. 2648–2665.
- [3] I. Sayar, A. Bartel, E. Bodden, and Y. Le Traon, An in-depth study of java deserialization remote-code execution exploits and vulnerabilities, *ACM Transactions on Software Engineering and Methodology*, vol. 32, no. 1, pp. 1–45, 2023.
- [4] K. Lian, L. Zhang, G. Yang, S. Mao, X. Wang, Y. Zhang, and M. Yang, Component security ten years later: An empirical study of cross-layer threats in real-world mobile applications, *Proc. ACM Softw. Eng.*, vol. 1, no. FSE, Jul. 2024.
- [5] W. Xue, X. Xia, P. Wan, P. Zhong, and X. Zheng, Adversarial attack on object detection via object feature-wise attention and perturbation extraction, *Tsinghua Science and Technology*, vol. 30, no. 3, pp. 1174–1189, 2024.
- [6] Y. Xu, D. Li, Q. Li, and S. Xu, Malware evasion attacks against iot and other devices: An empirical study, *Tsinghua Science and Technology*, vol. 29, no. 1, pp. 127–142, 2023.
- [7] C. Hazman, A. Guezzaz, S. Benkirane, and M. Azrour, Enhanced ids with deep learning for iot-based smart cities security, *Tsinghua Science and Technology*, vol. 29, no. 4, pp. 929–947, 2024.
- [8] M. C. Barbieri, B. I. Grisci, and M. Dorn, Analysis and comparison of feature selection methods towards performance and stability, *Expert Systems with Applications*, p. 123667, 2024.
- [9] A. E. Cinà, K. Grosse, A. Demontis, S. Vascon, W. Zellinger, B. A. Moser, A. Oprea, B. Biggio, M. Pelillo, and F. Roli, Wild patterns reloaded: A survey of machine learning security against training data poisoning, *ACM Computing Surveys*, vol. 55, no. 13s, pp. 1–39, 2023.
- [10] Y. Jiang, W. Zhang, and Y. Chen, Data quality detection mechanism against label flipping attacks in federated learning, *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 1625–1637, 2023.
- [11] X. Wang, C. Fang, M. Yang, X. Wu, H. Zhang, and P. Cheng, Resilient distributed classification learning against label flipping attack: An admm-based approach, *IEEE Internet of Things Journal*, vol. 10, no. 17, pp. 15 617–15 631, 2023.
- [12] L. Zang and Y. Li, Detection and mitigation of label-flipping attacks in fl systems with kl divergence, *IEEE Internet of Things Journal*, 2024.
- [13] E. Hallaji, R. Razavi-Far, M. Saif, and E. Herrera-Viedma, Label noise analysis meets adversarial training: A defense against label poisoning in federated learning, *Knowledge-Based Systems*, vol. 266, p. 110384, 2023.
- [14] R. Yang, H. He, Y. Wang, Y. Qu, and W. Zhang, Dependable federated learning for iot intrusion detection against poisoning attacks, *Computers & Security*, vol. 132, p. 103381, 2023.
- [15] J. Akram, A. Anaissi, A. Akram, R. S. Rathore, and R. H. Jhaveri, Adversarial label-flipping attack and defense for anomaly detection in spatial crowdsourcing uav services, *IEEE Transactions on Consumer Electronics*, 2024.
- [16] B. Hou, J. Gao, X. Guo, T. Baker, Y. Zhang, Y. Wen, and Z. Liu, Mitigating the backdoor attack by federated filters for industrial iot applications, *IEEE Transactions on Industrial Informatics*, vol. 18, no. 5, pp. 3562–3571, 2021.
- [17] S. Alharbi, Y. Guo, and W. Yu, Collusive backdoor attacks in federated learning frameworks for iot systems, *IEEE Internet of Things Journal*, 2024.
- [18] Y. Wang, D.-H. Zhai, Y. Xia, and D. Liu, Pervk: A robust personalized federated framework to defend against backdoor attacks for iot applications, *IEEE Transactions on Industrial Informatics*, 2023.
- [19] Y. Wang, D.-H. Zhai, D. Han, Y. Guan, and Y. Xia, Mitdba: Mitigating dynamic backdoor attacks in federated learning for iot applications, *IEEE Internet of Things Journal*, 2023.
- [20] Y. Zhang, F. Feng, Z. Liao, Z. Li, and S. Yao, Universal

- backdoor attack on deep neural networks for malware detection, *Applied Soft Computing*, vol. 143, p. 110389, 2023.
- [21] Z. Yao, H. Zhang, Y. Guo, X. Tian, W. Peng, Y. Zou, L. Y. Zhang, and C. Chen, Reverse backdoor distillation: Towards online backdoor attack detection for deep neural network models, *IEEE Transactions on Dependable and Secure Computing*, 2024.
- [22] R. Gunawardena, S. Jayawardena, S. Seneviratne, R. Masood, and S. S. Kanhere, Single-sensor sparse adversarial perturbation attacks against behavioural biometrics, *IEEE Internet of Things Journal*, 2024.
- [23] N. Alhussien, A. Aleroud, A. Melhem, and S. Y. Khamaiseh, Constraining adversarial attacks on network intrusion detection systems: transferability and defense analysis, *IEEE Transactions on Network and Service Management*, 2024.
- [24] M. N. Ali, M. Amer, and M. Elsis, Reliable iot paradigm with ensemble machine learning for faults diagnosis of power transformers considering adversarial attacks, *IEEE Transactions on Instrumentation and Measurement*, 2023.
- [25] X.-Y. Kong and G.-H. Yang, An intrusion detection method based on self-generated coding technology for stealthy false data injection attacks in train-ground communication systems, *IEEE Transactions on Industrial Electronics*, vol. 70, no. 8, pp. 8468–8476, 2022.
- [26] A. Ahmadi, M. Nabipour, S. Taheri, B. Mohammadi-Ivatloo, and V. Vahidinasab, A new false data injection attack detection model for cyberattack resilient energy forecasting, *IEEE Transactions on Industrial Informatics*, vol. 19, no. 1, pp. 371–381, 2022.
- [27] C. Zhang, X. Costa-Perez, and P. Patras, Adversarial attacks against deep learning-based network intrusion detection systems and defense mechanisms, *IEEE/ACM Transactions on Networking*, vol. 30, no. 3, pp. 1294–1311, 2022.
- [28] H. Rafiq, N. Aslam, U. Ahmed, and J. C.-W. Lin, Mitigating malicious adversaries evasion attacks in industrial internet of things, *IEEE Transactions on Industrial Informatics*, vol. 19, no. 1, pp. 960–968, 2022.
- [29] A. Eljaily, M. Y. Uddin, and S. Ahmad, Novel framework for an intrusion detection system using multiple feature selection methods based on deep learning, *Tsinghua Science and Technology*, vol. 29, no. 4, pp. 948–958, 2024.
- [30] M. Lefoane, I. Ghafir, S. Kabir, and I.-U. Awan, Unsupervised learning for feature selection: A proposed solution for botnet detection in 5g networks, *IEEE Transactions on Industrial Informatics*, vol. 19, no. 1, pp. 921–929, 2022.
- [31] Y. Han, Y. Wang, Y. Cao, Z. Geng, and Q. Zhu, A novel wrapped feature selection framework for developing power system intrusion detection based on machine learning methods, *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2023.
- [32] H. Wang, X. Di, Y. Wang, B. Ren, G. Gao, and J. Deng, An intelligent digital twin method based on spatio-temporal feature fusion for iot attack behavior identification, *IEEE Journal on Selected Areas in Communications*, 2023.
- [33] A. Ghubaish, Z. Yang, A. Erbad, and R. Jain, Lemda: A novel feature engineering method for intrusion detection in iot systems, *IEEE Internet of Things Journal*, 2023.
- [34] J. He, C. Tang, W. Li, T. Li, L. Chen, and X. Lan, Br-hidf: An anti-sparsity and effective host intrusion detection framework based on multi-granularity feature extraction, *IEEE Transactions on Information Forensics and Security*, 2023.
- [35] K. Sood, M. R. Nosouhi, D. D. N. Nguyen, F. Jiang, M. Chowdhury, and R. Doss, Intrusion detection scheme with dimensionality reduction in next generation networks, *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 965–979, 2023.
- [36] A. Thakkar and R. Lohiya, Fusion of statistical importance for feature selection in deep neural network-based intrusion detection system, *Information Fusion*, vol. 90, pp. 353–363, 2023.
- [37] E. M. Maseno and Z. Wang, Hybrid wrapper feature selection method based on genetic algorithm and extreme learning machine for intrusion detection, *Journal of Big Data*, vol. 11, no. 1, p. 24, 2024.
- [38] O. A. Alghanam, W. Almobaideen, M. Saadeh, and O. Adwan, An improved pio feature selection algorithm for iot network intrusion detection system based on ensemble learning, *Expert Systems with Applications*, vol. 213, p. 118745, 2023.
- [39] T. Tu, Y. Su, Y. Tang, W. Tan, and S. Ren, A more flexible and robust feature selection algorithm, *IEEE Access*, 2023.
- [40] N. Hansen and A. Ostermeier, Completely derandomized self-adaptation in evolution strategies, *Evolutionary computation*, vol. 9, no. 2, pp. 159–195, 2001.
- [41] N. Moustafa and J. Slay, The significant features of the unswnb15 and the kdd99 data sets for network intrusion detection systems, in *2015 4th international workshop on building analysis datasets and gathering experience returns for security (BADGERS)*. IEEE, 2015, pp. 25–31.
- [42] N. Moustafa, A new distributed architecture for evaluating ai-based security systems at the edge: Network ton_iot datasets, *Sustainable Cities and Society*, vol. 72, p. 102994, 2021.

Author biography



Waleed Almughalles is a Ph.D. candidate in Information Systems at the University of Southern Queensland. His research focuses on security and privacy challenges in Internet of Things, cloud computing, and artificial intelligence, with particular interests in intrusion detection, resource optimization for distributed IDS deployments, and privacy-preserving AI.



Jianming Yong is a professor of cybersecurity and information systems at the University of Southern Queensland. His research focuses on privacy-preserving cloud and 5G-IoT architectures, digital identity, blockchain, and secure distributed computing. He has published more than 140 peer-reviewed papers

and edited 14 research volumes. He also leads ARC-funded research on privacy for smart devices and contributes to industry-linked projects in e-health and smart-city applications.



Xiaohui Tao is a professor of artificial intelligence at the University of Southern Queensland. His research spans natural language processing, privacy-preserving machine learning, and healthcare applications. He has authored more than 250 peer-reviewed publications and received competitive funding, including an

ARC Discovery grant. His current work focuses on machine unlearning, knowledge-graph-enhanced models, precision psychiatry, and healthcare data privacy.



Yuefeng Li is a professor of computer science at Queensland University of Technology and co-leader of the Fundamental Methods in Data Science and AI theme at the QUT Centre for Data Science. His research focuses on knowledge and data engineering, pattern-based text mining, personalised ontology learning, and

machine-learning-based information filtering. He has authored more than 250 peer-reviewed publications and has received support from multiple ARC and Cooperative Research Centre projects.



Jun Shen is a renowned professor in the School of Computing & Information Technology at the University of Wollongong, known for his contributions to service-oriented computing, edge-cloud systems, AI-enhanced e-learning, and optimization for smart applications. As an IEEE Distinguished Lecturer,

Prof. Shen has authored over 170 peer-reviewed publications, earning more than 6,700 citations and an h-index of 41. His current projects include externally funded work through the NSFC, UGPN, and ARC-linked initiatives focused on data-driven micro-learning analytics.