Received: July 2, 2026. Revised: October 3, 2026. 63

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

SHAP-based Feature Stability Analysis for High-dimensional IoT Intrusion Detection Using Deep Neural Network

Praveen Shivanna[1]* **Priti Mishra[2]**

[1]*Department of Computer Science and Engineering, Sai Vidya Institute of Technology, Bengaluru, Affiliated to Visvesvaraya Technological University, Belagavi, India*
[2]*Department of Computer Science and Engineering (Artificial Intelligence and Machine Learning), Sai Vidya Institute of Technology, Bengaluru, Affiliated to Visvesvaraya Technological University, Belagavi, India*
* Corresponding author's Email: praveen.scta@gmail.com

Abstract: In today's world, the Internet of Things (IoT) has emerged as a vital network infrastructure in the process of digitalization. However, existing works struggle with high-dimensional and redundant features that increase computation and reduce efficiency. Furthermore, they fail to distinguish critical features for specific attack types, making their security decisions difficult to interpret. This manuscript presents a SHapley Additive exPlanations (SHAP)-based Feature Stability Analysis Network (SFSA-Net) for an Intrusion Detection System (IDS) in IoT. The proposed SFSA-Net uses SHAP to extract consistent stable features from a baseline Deep Neural Network (DNN), which then optimizes the final network for all attack types. To evaluate feature consistency, a stability-weighted feature ranking index, the Refined Explainable Index (REI), is established by conducting a three-stage stability analysis. The proposed approach achieves an accuracy of 98.75%±0.23% on UNSW-NB15 and 98.85%±0.28% on CICIoT-2023 datasets, respectively. Moreover, the proposed SFSA-Net achieves an optimal feature reduction of 78.7% on UNSW-NB15 and 97.4% on CICIoT-2023, minimizes memory consumption by 83.63% on UNSW-NB15 and 89.03% on CICIoT-2023, while simultaneously improving the detection of minority classes, specifically increasing the recall for worm attacks by 39.25 percentage points.

Keywords: Deep neural network, Feature stability, Intrusion detection system, Internet of Things, SHapley Additive exPlanations.

1. Introduction

Internet of Things (IoT) intrusion refers to malicious and unauthorized activities that target IoT networks by leveraging vulnerabilities in network protocols, device communication, and authentication mechanisms to gain unauthorized access, compromise security and disrupt services [1]. IoT networks pose significant challenges in terms of data privacy and security because of their inherent characteristics such as resource constraints, heterogeneity and scalability [2]. The IoT is emerging as an important communication infrastructure for current and next-generation systems, gaining significant interest in recent years [3]. IoT connects smart devices that enable automatic data exchange and improve awareness, data collection, and network traffic measurement [4]. IoT integrates a vast number of devices, making it a critical promoter for digital transformation [5]. The increased number of entry points is combined with the decentralized and diverse nature of the IoT network, which makes the system vulnerable to specific malicious activities [6]. Conventional IoT systems are designed primarily for IT networks and often lack precision, scalability, and adaptability, which are required for modern IoT ecosystems [7]. The limited storage volume, processing power of IoT devices, and traditional learning-based methods for IDS are not optimal [8]. Ensuring reliable and secure network operations is challenging for service providers and organizations [9]. Distinguishing between abnormal and normal activities faces significant complexity, and network

This article is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.
License details: https://creativecommons.org/licenses/by-sa/4.0/

traffic analysis is further complicated because of diverse protocols and varying data formats and presence of zero-day attacks [10].

Machine Learning (ML) and Deep learning (DL) models are increasingly used to improve IoT intrusion detection [11]. A DL-based cyber-attack detection model requires network traffic pattern analysis, which detects complex patterns of attacks and strengthens environment security [12]. This study addresses these gaps by developing a DL-based model for IDS that improves robustness and accuracy, especially for dynamic and minority attack classes [13]. The application of Artificial Intelligence (AI) in network intrusion detection systems has become a recent research topic and AI-based IDS models demonstrate effective performance in threat detection [14]. DL-based models have proven their ability and effectiveness in dimensionality reduction, data classification, and feature extraction [15]. Attackers are developing increasingly advanced attack patterns to compromise secure computing environments [16]. An IDS analyses IoT data packets and associated network flows across protocol stacks, while operating over constrained resources [17]. Broader threat landscapes open new doors to enlarge the attack surface for attackers [18]. DL-based models are becoming increasingly crucial as network traffic grows exponentially [19]. The inherent security vulnerabilities in IoT infrastructure pose severe challenges in achieving system reliability [20].

1.1 Problem statement

Existing IoT IDS models struggle with high-dimensional and redundant features that increase computation and reduce efficiency. They also fail to identify the features that are important for various attack types and lack interpretability.

1.2 Objective

To overcome this limitation, this study proposes the SFSA-Net, a SHAP-based feature engineering framework. The proposed approach identifies the most stable and informative features for different attack categories through feature importance and stability analyses, thus improving the detection performance.

1.3 Contribution

The main contributions of this manuscript are provided below:

- A DNN model is first trained with all features to obtain the baseline performance and the final optimization model is trained with an engineered and optimized feature subset.
- A SHAP analysis is used for feature importance extraction that enables extraction of both global feature importance and class-specific (attack-wise) feature importance.
- The feature engineering analysis is performed for three different experimental configurations. In case 1, the full set of features is used to train the DNN. For case 2, the DNN is trained with a subset of features (Top K), which is selected based on the global SHAP analysis. Finally, in case 3, attack-specific feature subsets identified by SHAP analysis are used for offline exploratory profiling of minority signatures during training, while the deployed system employs a single label-agnostic global REI subset to avoid any target leakage at runtime.
- A novel stability-aware feature engineering mechanism is employed through Refined Explainable Index (REI), which applies an exponential decay regularization to feature variance to systematically reduce the sensitivity of feature ranking to ensure consistent and robust detection performance across all attack categories including rare minority classes.
- To improve model interpretability, a local and global explanation approach is integrated through SHAP to show how specific feature attributions affect the neural network's internal security decisions.

The remaining manuscript is presented as follows: Section 2 represents the related works, which discusses existing proposed models with their advantages and disadvantages. Section 3 illustrates the proposed methodology with dataset description, pre-processing techniques, and working of the proposed architecture. Section 4 shows the results and discussion. Section 5 concludes the manuscript.

2. Related works

Shamshair Ali [21] introduced a hybrid DL-based model that synergized a Long Short-Term Memory Autoencoder and Multilayer Perceptron (MLP) for botnet detection in IoT. These fusion technologies improved sequential pattern recognition and sequential data that allowed the model to detect activities of intricate botnets within IoT networks. The model was designed to detect zero-day botnet attacks by learning unusual patterns without prior specific knowledge. However, the model relied on a centralized server that created a potential single point

This article is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.
License details: https://creativecommons.org/licenses/by-sa/4.0/

of failure. Consequently, a server compromise would directly expose all sensitive model parameters, as validated by unreduced baseline comparisons which show performance inconsistency with a variance measure σ of $\pm$ 1.45 under controlled baseline evaluations.

Maseno [22] developed a Genetic Algorithm (GA)-based optimized DL model such as a Genetic Algorithm-based LSTM and Gated Recurrent Unit for interpretable and efficient IDS in IoT. The GA was used for dual optimization, which simultaneously optimized hyperparameter tuning and feature selection that balanced interpretability and accuracy. To improve transparency, a Local Interpretable Model-agnostic Explanation (LIME)-based interpretability analysis was used to provide model decision-making transparency that improved explainability and trust. However, the GA metaheuristic process introduced substantial computational overhead during the training or testing phase, matching a similar unreduced baseline sequence with high demands of up to 25.10 MB RAM and 31.15% CPU evaluated under local replica testing.

Jablaoui [23] presented a hybrid DL-based model for IDS, which used the synergistic strength of GRU and Convolutional Neural Networks (CNNs). A CNN was used to extract spatial features from network traffic data and a GRU was used to model the temporal dependencies effectively, which made this combination especially well-suited for complex and dynamic IoT intrusion detection. The integration of the proposed model simplified the parameter structure while improving the model's capability to manage long-range dependencies. However, the model required improvement in capturing minority attack behaviors because class imbalance reduced recall which limits the evaluated CNN-GRU baseline model to a restricted global f1-score of 83.70% reimplemented in this manuscript.

Gan Zhu [24] introduced a GMDDPM-CT (Gaussian and Multinomial Denoising Diffusion Probabilistic Model with CNN-transformer), a generative minority-sample model for classification. The GMDDPM-CT used a diffusion model to create synthetic samples for minority classes, effectively improving data balance and addressing sample scarcity. Unlike traditional diffusion models, which were limited to continuous data, GMDDPM-CT efficiently handled both continuous and discrete features in network traffic. However, the model was resource-intensive and required high training complexity, which was not feasible for all environments, while feature selection based on stability reduces resource utilization down to a 0.77 MB RAM requirement observed during empirical baseline testing.

Fu [25] developed a spatiotemporal analysis-based approach called TSIDS (Spatial temporal IDS), which extracted the network behaviour interactions of an IDS. The TSIDS model integrated a spatial analysis module to extract spatial information over diverse events and a temporal analysis module was used to learn temporal dependencies from traffic data. For spatial correlation modelling of temporal features, a feature fusion module was employed for a customized gating Multilayer Perceptron (cgMLP). Unlike fixed GNN, TSIDS learned to evolve spatial features and was not limited to specific graph structures. However, extremely rare attacks with irregular distributions exhibited a higher susceptibility to misclassification, constraining baseline multi-class detection recall performance of Worms and Backdoor to 42.15% and 54.20% reimplemented in this manuscript.

Ogobuchi Daniel Okey [26] presented a Resource-aware security model that employed the aKullback-Leibler (RAID-KL), which was a resource-aware security model that used KD and an adaptive loss function integrated hybrid Jensen-Shannon (JS) and Kullback-Leibler (KL) divergence. By combining both divergences, the model improved convergence of distilled models and generalization while maintaining low computation, which enhanced numerical robustness and balanced knowledge sharing. RAID-KL employed a 1-Dimensional CNN (1DCNN) as a learning algorithm that transferred the learned representation. The loss function improved stability and convergence of the model. However, the model provided high sensitivity to hyperparameter settings for certain protocol-specific threats such as Message Queuing Telemetry Transport (MQTT)-based attacks, in contrast to feature selection metrics with stability reducing contextual variation to achieve stable performance variance of $\pm$ 0.40% evaluated in the local replica pipeline.

Existing models face limitations of high feature dimensionality, high computational complexity, and lack of ability to represent imbalanced minority classes of attacks due to reliance on static feature ranking. The proposed SFSA-Net approach introduces a three-stage stability analysis approach with REI to address these critical limitations. It simultaneously maximizes global feature importance and minimizes contextual sensitivity across diverse network conditions, and removes redundant dimensions while preserving high interpretability. This proposed architecture is effective in stabilizing the selection of key security indicators and enhances

This article is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.
License details: https://creativecommons.org/licenses/by-sa/4.0/

the detection accuracy of rare and highly irregular attack types.

3. Proposed methodology

The proposed methodology utilizes a DL-based approach, namely a SFSA-Net for intrusion detection in IoT. Two publicly available datasets such as UNSW-NB15 and CICIoT-2023 are utilized for IDS in IoT. Both datasets contain tabular data that are fed into pre-processing tasks, which include techniques such as imputation and normalization, that are used to ensure data quality. The pre-processed data are then fed directly into a DNN base model for the training process and subsequently, the SHAP analysis provides both global feature importance and attack-wise importance. These are then fed into the feature engineering analysis, which analyses the features with three subsets. Moreover, the final model is trained by utilizing an optimal feature subset, which is selected through stability analysis.

For interpretability, the explainable AI (XAI) technique such as SHAP is employed for IDS in IoT, the architecture of the proposed SFSA-Net is shown in Fig. 1.

3.1 Mathematical formulation

The mathematical formulation of this manuscript utilized symbols with definition is presented in Table 1.

3.2 Dataset description

Two publicly available datasets, namely UNSW-NB15 and CICIoT-2023 are used for IDS in IoT. Table 2 represents the datasets description.

To ensure statistical significance in the minority Worms class of UNSW-NB15 dataset (174 samples) which partitioned into 140 training samples and 34 testing samples. Synthetic oversampling through SMOTE technique is applied to the training set by using 140 original training samples, SMOTE generated 1,260 synthetic samples and producing Worms final training count of 1,400 with $k = 5$ neighbours.

3.3 Pre-processing

The input raw tabular data are fed into pre-processing techniques such as imputation and normalization to ensure data quality.

- Imputation: A data integrity check confirmed exactly 0 missing entries across both datasets. Consequently, no statistical values such as feature-wise means are substituted, thus preserving the original data profile without alteration.
- Normalization: Min-Max scaling is used to scale numerical input features into a specific range such as 0 to 1 to ensure that all features contribute equally to the training model.

Although Min-Max scaling normalizes features to reduce the ranges, the scalar fitted strictly on training set and subsequently applied to testing set which prevent data leakage. .

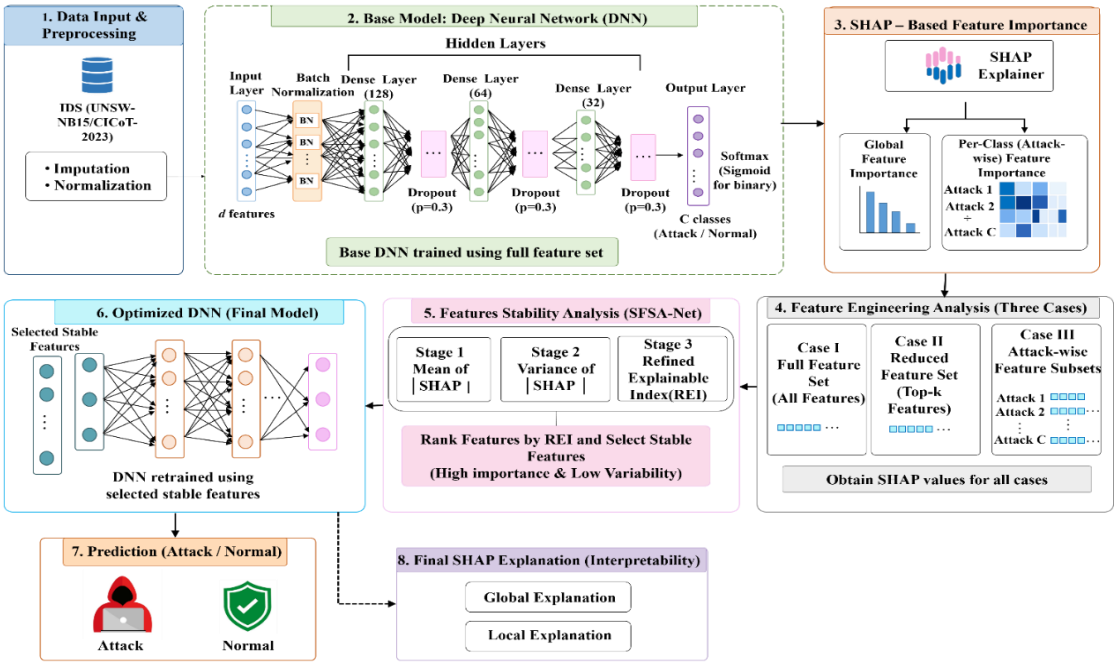

Figure. 1 Architectural framework of the proposed SFSA-Net approach for IoT-based intrusion detection

This article is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.
License details: https://creativecommons.org/licenses/by-sa/4.0/

Table 1. Symbol description

Symbol	Definition
F	Full baseline feature set
d	Total features in full set ($d = \|F\|$)
S^*	Optimal stable feature subset selected
$\|S^*\|$	Total number of features in selected subset
S	Shifting subset used during SHAP summations
x	Input traffic sample vector
x_i	Specific feature value at index i
n/N	Sample index/Total number of dataset samples
k/K	Attack class index/Total number of classes
e/E	Experimental scenario index/Total set of scenarios
$\emptyset_i^c$	SHAP value of feature i for class c
$\emptyset_i^{(e)}$	SHAP value of feature i in experimental scenario e
μ_i	Mean absolute SHAP value (Global importance)
V_i	Volatility variance of SHAP values (Consistency)
SS_i	Unified final stability score
y_{final}	Final classification output of model
$W^{(l)}/b^{(l)}$	Weight matrix/Bias vector of hidden layer l
$h^{(l)}$	Hidden layer activation output at layer l
σ	Nonlinear activation function
ϵ	Infinitesimal constant avoiding division by zero

The proposed model targeted attack feature selection ensures no distortion occurs because the selected features highlight the statistical variation in rare minority classes

3.4 Base model training

Considering all architectural considerations, the proposed DNN-based IDS model is effectively designed to accurately classify and detect potential intrusion in IoT networks. The model architecture contains various hidden layers for extracting and classifying tasks. This base architecture provides a foundation for conducting feature importance analysis using SHAP to resolve the impact of high-dimensional and redundant features. It is important to emphasize that the DNN architecture specifically handles high-dimensional tabular IoT data by capturing the non-linear dependencies between features for improved IDS performance.

A standard feed-forward DNN classifier is designed to limit computational costs in resource-limited IoT edge settings. This design ensures that improvements in accuracy can be attributed only to the REI based feature selection approach rather than architectural over-parameterization.

The architectural features of the DNN model are as follows:

- Input layer: The data sample represents the features of the IoT network traffic, which is fed into the input layer of the DNN model. The pre-processed normalized feature vectors are passed directly to this layer, thus ensuring an even distribution of data and learning processes for the network.
- Batch Normalization layer: This layer is used to normalize the input data to improve the speed and stability of the training procedure. It minimizes internal covariate shift, thereby enabling faster convergence and better training performance.
- Dropout layer – A dropout layer is integrated into the architecture as a regularization technique to mitigate overfitting. This layer randomly switches off some neurons, pushing the network to learn more generalized and flexible features. This increases the model's ability to generalize across a high-dimensional feature space.
- Dense Layer (Feature Learning) – The dense layer is the major element, which is responsible for feature learning. This dense layer transforms the input into a high-level representation, containing relevant patterns that indicate cyber-attacks. In contrast to traditional approaches, a DNN learns the internal representation from the input features rather than creating new ones.

Dense Layer (Output) – The final dense layer creates class-wise probability distributions for possible intrusion classes using a suitable activation function such as Softmax. These output probabilities are passed directly to subsequent SHAP analysis.

3.5 Feature importance using SHAP

To improve the model's explainability and facilitate the feature selection, the SHAP analysis is utilized in the attack detection process of the DNN model. SHAP provides both local and global interpretability and transforms black-box decisions of the DNN into transparent classification paths. This approach evaluates the independent contribution of each feature independently towards the probability prediction for each specific class, thereby providing interpretability to the model's decision-making process by highlighting the important features for various attack types.

This article is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.
License details: https://creativecommons.org/licenses/by-sa/4.0/

Table 2. Dataset Description for both UNSW-NB15 and CICIoT-2023 Datasets

Feature	UNSW-NB15 [27]	CICIoT-2023 [28]
Primary Target	General network system	Large-scale IoT environment
Classification Type	Binary and Multi-class	Binary, 8-class (7 attack categories + Benign), 34-class (33 attack types + Benign)
Classification Labels	Label (0: Normal, 1: Attack) Attack_cat (attack categories)	Label (0: Benign, 1: Attack) for binary, specific attack names for multi-class
Total Attack Type	9 major categories	7 major categories
Attack List and Class distribution	Total samples: 257,673 Normal (93,000), Analysis (2,677), DoS (16,353), Exploits (44,525), Reconnaissance (13,987), Shellcode (1,511), Worms (174), Fuzzers (24,246), Generic (58,871), Backdoors (2,329)	Total samples: 47,686,579 Benign (1,074,369), DDoS (34,386,819), Reconnaissance (87,144), Web-based (3,637), Spoofing (513,506), Brute Force (11,159), DoS (7,235,932) and Mirai (4,374,013)
Data split (Train/Test)	Total samples: 257,673 Train split (80%): 206,138, Test split (20%): 51,535, Class-wise split (Train/Test): Normal (74,400/18,600), Analysis (2,141/536), DoS (13,082/3,271), Exploits (35,620/8,905), Reconnaissance (11,189/2,798), Shellcode (1,209/302), Worms (140/34), Fuzzers (19,397/4,849), Generic (47,097/11,774), Backdoors (1,863/466)	Total samples: 47,686,579 Train split (80%): 38,149,263, Test split (20%): 9,537,316, Class-wise split (Train/Test): Benign (859,495/214,874), DDoS (27,509,455/6,877,364), Reconnaissance (69,715/17,429), Web-based (2,910/727), Spoofing (410,805/102,701), Brute Force (8,927/2,232), DoS (5,788,746/1,447,186), Mirai (3,499,210/874,803)
Protocol Focus	TCP, UDP, ICMP, etc.	IoT-specific protocols and network layers (IP, TCP, UDP, ICMP, HTTP, etc.)
Ground truth tool	IXIA PerfectStorm	Real-world IoT topology (105 devices:84 training, 21 testing) with malicious IoT attackers and victims

Specifically, for each class C, a SHAP value $\emptyset_i^C$ is calculated to represent the marginal contribution of the feature x_i towards the prediction of a given sample x is defined in Eq. (1):

$$\emptyset_i^C(x) = \sum_{S \subseteq F\{i\}} \frac{|S|!(|F|-|S|-1)!}{|F|!} [f_C(S \cup \{i\}) - f_C(S)] \quad (1)$$

Where F defines the full set of features, S is the feature subset excluding feature i, and $f_C(S)$ represents the model prediction for C class utilizing only subset features. To strictly bound the combinatorial explosion, the Permutation Sampling approximation is used for subset selection S, through the sampling of a specific number of random permutations of the features. A larger positive SHAP value represents a stronger positive contribution toward the model's prediction of class C.

To interpret the proposed model, SHAP summary plots are produced for each class to analyse the average contribution of the key features to predict outcomes. Moreover, the general importance of features over all classes is demonstrated based on their mean absolute SHAP values across the entire dataset, which is defined in Eq. (2):

$$\mu_i = \frac{1}{N.C} \sum_{j=1}^{N} \sum_{c=1}^{C} |\emptyset_i^C(x_j) \quad (2)$$

Where N represents the total number of samples and C depicts the number of classes. Based on this metric, the most important features are selected to address high-dimensional and redundant features, resulting in the final optimized and resource-efficient DNN model.

Additionally, SHAP waterfall plots visually map the classification pathway of individual samples. From the baseline value $E[f_C(X)]$ of the model prediction, the SHAP values of each feature is accumulated to evaluate the final prediction value, which is expressed in Eq. (3):

$$f_C(x) = E[f_c(X)] + \sum_{i=1}^{|F|} \emptyset_i^C(x) \quad (3)$$

Where $E[f_C(X)]$ is the model baseline (expected value) for class C, $\emptyset_i^C$ shows the feature i contribution for the sample x, and $|F|$ is the total number of features in the complete set F. This transforms the black-box nature of the DNN into an explainable process that supports effective security auditing and tracing of an attack behavior.

This article is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.
License details: https://creativecommons.org/licenses/by-sa/4.0/

Finally, SHAP dependence plots are constructed to evaluate model's feature interactions to explore non-linear relationships between model outputs and key feature values, enabling the identification of discriminative thresholds and important feature intervals.

3.6 Feature engineering analysis

The feature engineering analysis consists of three experimental cases, which are designed to evaluate the impact of interpretability and feature selection on model detection accuracy and efficiency. To address redundant and high-dimensional features in IoT IDS, this manuscript proposes SFSA-Net, thus ensuring suitability for deployment on resource-constrained devices. The proposed framework combines SHAP-based feature attribution with DNN performance evaluation.

- **Case 1: Full Feature Analysis using DNN**

In this approach, the training and testing of the DNN are conducted by utilizing all features of IoT network traffic dataset. The data samples are fed into the input layer of the DNN. Moreover, a multi-layered neural network architecture is used to perform feature extraction and classification processes. During the training phase, the model optimizes neural network weights through backpropagation. The training process is completed by reducing the difference between the predicted and actual results by utilizing an Adam (Adaptive moment estimation) optimizer.

Training and validation results using the full feature space are analysed by following multi-class terminology implied, which maps the final classification output of the class index with highest probability using argmax operator represented in Eq. (4):

$$y = argmax_{c \in C} f_c(x) \tag{4}$$

This method acts as a baseline case; however, it poses certain difficulties due to the "black-box" nature of DNNs, where non-transparent decision-making processes hinder effective security and complicate the tracing of attack behaviors.

- **Case 2: Reduced Features (Top K Features) from SHAP to DNN**

To improve efficiency and reduce computation, this case measures the model performance using a subset of only the most impactful features, which are identified by SHAP. The relative importance of each individual feature is measured using the mean feature importance (defined previously in Eq. (2), where the absolute SHAP value for each observation is averaged throughout the entire dataset. Using this measure, plots are generated to visualize the top K significant features. Subsequently, feature subsets are constructed using a specific number of top K features (for example: 50, 30, and 10) for DNN model training.

Despite the high-dimensional input space, this approach leverages the observation that few top features often dominate the entire decision-making process. This dominance highlights their critical importance in shaping model predictions. Therefore, using fewer features such as ct_state_ttl, $sbytes$, $FwdIATMean$, the DNN model achieves near-optimal detection performance while significantly reducing the computational overhead.

- **Case 3: Attack-wise Feature Subsets from SHAP to DNN**

This case performs a detailed analysis by selecting features independently for each attack type, which are most relevant for each specific attack. Specifically, Eq. (2) is applied to calculate the SHAP values, quantifying the importance of each feature with respect to each attack class k.

In this process, the DNN is used to evaluate the performance of individual attack-type feature sets. Consequently, this approach identifies essential features for each attack type, ranging from the traffic volume of DoS attacks or packet-level protocol signatures such as TCP flag counts, ICMP payload sizes and IP header options to the spatio-temporal and protocol-specific characteristics of backdoor patterns. This attack-wise optimization ensures that the IDS achieves scalability and precision in detecting various types of attacks, making the black-box nature of the DNN an explainable process. Without performing a distinct correlation analysis, SHAP incorporates feature redundancy and multi-collinearity in its computations of feature contribution margins across all combinations. Importantly, Case 3 acts strictly as an offline exploratory feature-attribution profiling stage during training phase to identify attack-specific data signatures. In the inference runtime simulation, this specific attack branch is entirely avoided and receives no ground-truth testing data labels. Instead, only the single, static and label-independent optimal feature subset S^* identified during training using the REI, which prevents any target leakage.

However, three feature engineering cases rely on feature rankings based on static importance scores, which often overlook stability and consistency across varying network conditions. To overcome this

This article is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.
License details: https://creativecommons.org/licenses/by-sa/4.0/

limitation and ensure a consistent intrusion detection mechanism, a novel stability-aware feature selection mechanism is proposed.

3.7 Stability-aware feature engineering

The primary novelty of the SFSA-Net approach lies in its shift from evaluating static feature importance to a formal multi-scenario stability-guided feature selection mechanism. Although state-of-the-art models such as RAID-KL and GA-GRU use post-hoc feature attribution methods to interpret black-box nature, however they do not consider the dynamic nature of the network properties. Given the dynamic nature of IoT environments, features are critical for high-volume Denial-of-Service (DoS) attacks but provide near-zero discriminative power for probe or backdoor intrusions. To overcome this challenge, a three-stage stability analysis is introduced to produce a REI, which remains robust across diverse attack protocols and varying input dimensionalities.

- **Step 1: Multi-scenario SHAP attribution**

While previous feature selection procedures focus on ranking features here, the SHAP value is calculated for each feature i across M diverse experimental contexts. Each experiment $s \in \{1,2,\dots,M\}$ represents various situations such as a specific attack category or a specific feature-set scale, where $M = 15$ maps to the total attack sub-categories. This approach addresses the contextual variability of feature significance rather than treating it as globally invariant.

The marginal contribution $\emptyset$ for a feature x_i in a specific scenario s is calculated in Eq. (5) using the weighted average of its contributions across all possible feature subsets S.

$$\emptyset_i^s = \sum_{S \subseteq F\{i\}} \frac{|S|!(|F|-|S|-1)!}{|F|!} \left[f_s(S \cup \{i\}) - f_s(S)\right] \quad (5)$$

- **Step 2: Global Importance Quantification ($\boldsymbol{\mu}$)**

The Mean Absolute SHAP value (μ_i) is calculated to obtain an average discriminative weight of a feature across the entire spectrum of attacks. This step aggregates importance scores from step 1 to prevent the model from favouring biased features that performs well only on high-frequency majority traffic.

A large value of μ_i indicates a feature with high discriminative power for maintaining the model's ability to distinguish between malicious and benign traffic across attack types, which is defined in Eq. (6):

$$\mu_i = \frac{1}{N}\sum_{n=1}^{N} |\emptyset_{i,n}| \quad (6)$$

Even if the distribution of SHAP values for features do not have a normal distribution, the average value is used here to ensure that the actual total of SHAP contributions remains unchanged. The REI provides the maximum separability for minority class by separating those feature sets which have zero variance for multiple scenarios $\sigma_i^2 \to 0$, ensuring subtle minority attacks do not get submerged under the high-frequency traffic from majorities.

- **Step 3: Feature Consistency Analysis ($\boldsymbol{V}$)**

The core innovation of REI lies in evaluating situational volatility by computing the variance (σ_i^2) of SHAP values across all experiments. This approach quantifies feature importance fluctuation under various conditions. Features exhibiting high situational volatility σ_i^2 are considered as unstable due to their high context-dependency and they offer high discriminative power for specific attacks but remain uninformative in others. In contrast, stable features with low variance are prioritized because they indicate fundamental and invariant protocols of network communication as defined in Eq. (7):

$$\widehat{\emptyset}_{i,m} = \max\left(-3\lambda, \min\left(3\lambda, \emptyset_{i,m}\right)\right) \quad (7)$$

Where λ denotes standard deviation of the baseline SHAP distribution. This clipped value used to calculate variance defined in Eq. (8):

$$\sigma_i^2 = \frac{1}{M}\sum_{m=1}^{M}(\widehat{\emptyset}_{i,m} - \mu_i)^2 \quad (8)$$

- **Stability-based Feature Selection**

To obtain a final ranking for the REI, the output of importance and consistency analyses are combined into a single Stability Score (I_i). Instead of relying on an existing "Top K" strategy, where optimization is solely driven by maximizing μ_i, the selection criteria involves a dual-objective optimization to maximize overall importance and minimize variance.

To mathematically penalize contextually sensitive features under non-stationary conditions, introduce a novel exponential decay regularization envelope. Unlike standard stability-selection approaches that depend on linear or polynomial variance division μ_j/σ_j^2, a novel approach of REI uses an exponential envelope $e^{-\gamma\sigma_j^2}$ A. This mathematical restriction generates an aggressive penalty cliff that causes the contextual variability of the ranking scores to asymptotically approach zero.

This article is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.
License details: https://creativecommons.org/licenses/by-sa/4.0/

A small constant term $\epsilon = 1 \times 10^{-7}$is employed to ensure numerical stability and prevent division-by-zero errors. The features with the highest I_i are chosen as an optimal input for the proposed model, as defined in Eq. (9):

$$I_i = \frac{\mu_i .\max(\xi, \exp(-\gamma .\sigma_i^2)}{\sigma_i^2 + \epsilon} \quad (9)$$

Where $\gamma = 1.2$ denotes the structural sensitivity coefficient derived through grid search over the range [0.1,5.0] with step size of 0.1 and $\xi = 10^{-4}$ denotes lower-bound clipping threshold to prevent zero-gradient starvation.

To validate the analytical stability of the REI index, establish its mathematical boundaries:

- Strict Boundedness: In any feature space, the value of I_i is positive and has an upper limit is defined in Eq. (10):

$$0 \leq I_i \leq \frac{\mu_i}{\epsilon} \leq \frac{\mu_{max}}{\epsilon} \quad (10)$$

- Since $\mu_i \geq 0, \sigma_i^2 \geq 0 \; and \; \exp(-\gamma.\sigma_i^2) \in (0,1)$, the maximum score occurs at perfect stability $\sigma_i^2 = 0$, showing $lim_{\sigma_i^2 \to 0} I_i = \frac{\mu_i}{\epsilon} \leq \frac{\mu_{max}}{\epsilon}$. This prevents numerical explosion.
- Asymptotic Convergence: In high volatile features where variance $\sigma_i^2 \to \infty$ are automatically filtered that defined in Eq. (11):

$$\lim_{\sigma_i^2 \to \infty} I_i = 0 \quad (11)$$

The exponential decrease of the numerator mathematically dominates the linear growth of the denominator, the value of the index gets pushed towards zero as σ_i^2 increases.

By prioritizing features that satisfy the condition of $I_i \to \frac{\mu_{max}}{\epsilon}$ (which occurs when variance $\sigma_i^2 \to 0$), SFSA-Net discovers nuanced, explainable indices that maintain their effectiveness while achieving feature reduction of 78.7% on UNSW-NB15 and 97.4% on CICIoT-2023. The features that tend towards $\sigma_i^2 \to \infty$ are strongly suppressed as $I_i \to 0$ due to the asymptotic behavior of the exponential numerator. This consistency-driven process ensures that the resultant model is resource-aware and resilient, which is capable for identifying complex protocol cues to recognize minority attacks without exceeding available computational resources. Resource-awareness is achieved through dimensional feature reduction rather than a specific runtime mechanism, which reduces memory usage and computational latency of inference at run-time. The explainability technique is post hoc rather than intrinsic to training and this design decouples explanation overhead from low-latency IoT network inference, measured across computational runtime configurations.

3.8 Final optimized model

The final stage of the proposed SFSA-Net approach includes training a DNN by utilizing an optimal feature subset as F^* of size $k < d$, where k is dynamically bounded by selecting only features with REI score $I_i > 0$, which is derived from the stability-based feature selection process. By focusing on features with high REI, the model eliminates redundancy and high-dimensional noise, reducing the feature count while maintaining near-optimal detection performance.

The DNN architecture contains an input layer, which is followed by multiple hidden layers that involve batch normalization, dropout, and an output layer. The model maps the input features to the class probability distribution which is defined as a layer l in Eq. (12) as follows:

$$h^{(l)} = ReLU(W^{(l)} h^{(l-1)} + b^{(l)}) \quad (12)$$

Where $W^{(l)}$ and $b^{(l)}$ represent the weights and biases of the layer l that is optimized by utilizing the Adam algorithm which inherently smooths gradient trajectories and reduces validation curve noise. ReLU activation used in the hidden layers, whereas the final output layer applies the Softmax function to handle multi-class intrusion detection.

3.9 Explainability

To mitigate the black-box nature of the conventional DNNs, the proposed approach utilizes SHAP post-training for explaining intrusion detection in IoT. Although post-hoc XAI models do not possess the natural interpretability properties of attention models due to their lack of architectural transparency, they avoid the drawbacks that arise due to the nature of interpretable architecture design.

For each class C, the SHAP value $\emptyset_i^C$ represents the contribution of the feature i to the final classification, as defined previously in Eq. (1).

- Quantifying Local and Global Importance

Local Explanation: For tracing specific attacks such as DoS or worms, SHAP waterfall plots are employed to visualize the classification pathway

This article is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.
License details: https://creativecommons.org/licenses/by-sa/4.0/

from the baseline value $\emptyset_0^C$ as to the final prediction, as $f_C(x)$ is defined for class C in Eq. (3).

Local additivity is achieved despite non-linearity of ReLU boundaries because the SHAP values approximate the models behaviour locally for each instance.

Global Importance: The general importance of features for a specific class C is evaluated over engineered feature space using the mean absolute SHAP value, as defined in Eq. (2).

The proposed SFSA-Net model provides high accuracy and REI, enabling security auditing and tracing of attack behaviors effectively.

Pseudo code 1: SFSA-Net

Input: Dataset X with full feature set F
Output: Stable feature subset S^*, Optimized model DNN_{final}
Stage 1: Baseline DNN Modelling
Train DNN_{base} on training partition (X_{train}, Y_{train}) using full feature set F
Output prediction probability distributions for all k classes
Stage 2: Multi-Scenario SHAP attribution
for each scenario $e \in E$ do
for each feature $i \in F$ do
Compute context-dependent SHAP value $\emptyset_i\hat{}(e)$
end for
end for
Stage 3: Refined Explainable Index (REI) Computation
for each feature $i \in F$ do
Calculate global importance magnitude μ_i
Calculate situational volatility variance σ_i^2
Compute unified final stability score I_i
end for
Stage 4: Stability-based feature selection
Sort complete feature set F in descending order of I_i
Extract top k elements to construct optimal stable feature subset S^*
Filter training partition to match optimized dimensions: $X_{opt} \leftarrow X[:, S^*]$
Stage 5: Final optimized training and Runtime inference simulation
Reconfigure DNN input layer to match subset size $|S^*|$
Train final optimized model DNN_{final} on compressed training partition (X_{opt}, Y_{train})
Runtime inference simulation phase (Label-independent):
Statically filter live testing sample: $x_{opt} \leftarrow x_{test}[S^*]$
Execute model forward pass: $P \leftarrow Softmax(DNN_{final}(x_{opt}))$
Resolve final classification: $y_{pred} \leftarrow argmax\ (P)$
Return S^*, DNN_{final}

4. Experimental results

The proposed model employs a chronological 80:20 train-test partitioning for primary evaluation, followed by a 10-fold cross-validation pipeline repeated 2 times using distinct random seeds to generate 20 fold-level observation obtained from two repetitions of 10-fold cross-validation, as a secondary evaluation which is strictly for statistical significance. The individual run tables show a representative 10-fold baseline subset, while the secondary statistical measures, variance distributions, ablation study and inference calculations across Tables 5 to 10 are computed utilizing the complete set of 20 observations. To evaluate computational workload of a multi-client edge node validation environment, the proposed SFSA-Net model is implemented on a Windows 11 64-bit operating system. The system was evaluated using an Intel Core i7-13700H 14-core processor with 16GB RAM, an NVIDIA GeForce RTX GPU and 1 TB NVMe SSD. The software setup uses Python 3.12, PyTorch 1.8.0 and the SHAP library. The source code of the proposed SFSA-Net model is https://doi.org/10.5281/zenodo.22333845. The 20 run-wise results also provided in the source code link. To ensure data provenance across both UNSW-NB15 and CICIoT-2023 dataset evaluations, these 20 fold-level validation observations are actual execution results that were generated through re-execution our cross-validation pipeline across two different trial iterations with a distinct global seed initializer values (Repeat 1 = Seed 42, Repeat 2 = Seed 43) using the PyTorch deterministic execution engine. Every descriptive statistic, variance calculation and non-parametric Wilcoxon step presented in our tables are based on this preserved pool of raw observations, which are systematically mapped across specific fold IDs and repeat IDs within our public data repository. To evaluate performance and prevent data leakage, the UNSW-NB15 and CICIoT-2023 datasets were partitioned into 80% for training and 20% for testing subsets before performing any data pre-processing, using a chronological group split according to timestamps of the flow records to prevent temporal traffic correlation. Specifically, the 80:20 split processed all original 47,686,579 total records of CICIoT-2023 dataset extracted directly from standard CSV network flow tracking logs across the original 389 features, showing 38,149,263 samples for training

This article is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.
License details: https://creativecommons.org/licenses/by-sa/4.0/

and 9,537,316 samples for testing, while processing all original 257,673 total records for UNSW-NB15 dataset across 47 features, yielding 206,138 samples for training and 51,535 samples for testing. Accuracy of $98.85\% \pm 0.28\%$ on CICIoT-2023 dataset was achieved using the 8-class labels Benign, DDoS, DoS, Mirai, Spoofing, Reconnaissance, Web-based and Brute force categories with an 8 neuron output layer structure. To limit computational cost, SHAP post-training analysis was restricted by setting a background size of 100 benign samples in 1000 stratified samples with 10 permutations. To ensure reproducibility of the training over large CICIoT-2023 dataset, training was performed using a PyTorch RandomSampler without replacement with the seed provided by torch. Generator and worker_init_fn provided by DataLoader which ensures that the mapping of worker seeds is deterministic as base_seed+worker_id to ensure deterministic sequence. Mini-batch processing of 1,000 steps per epoch (processing of ~4.096 million samples or ~10.7% of samples dynamically per epoch), where 50 epochs systematically generate 50,000 optimization update steps that be replicated to cover the entire dataset pool. Validation stopping criteria of $\Delta \text{loss} < 10^{-4}$ over 5 checkpoints and 50,000 updates overall for 50 epochs was recorded as strict GPU optimization-loop time only, excluding all disk scaling, I/O and SHAP computation latency.

Table 3 presents the hyperparameter setting for the proposed model.

4.1 Qualitative analysis

The proposed SFSA-Net model is evaluated using metrics such as accuracy, precision, recall, and F1-score.

- **Accuracy**

The overall value of the classification technique is evaluated based on its accuracy.

Accuracy is the percentage of all observations predicted accurately, which includes both true positives and true negatives. The accuracy is defined as follows in Eq. (13):

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \times 100 \quad (13)$$

Although the accuracy of the global model be distorted because of the high DDoS traffic, it is included strictly as a baseline reference alongside these class-specific metrics.

- **Precision**

The ratio of accurately predicted positive observations to the total positive prediction is called precision. The precision is defined as follows in Eq. (14):

$$Precision = \frac{TP}{TP+FP} \times 100 \quad (14)$$

- **Recall**

Recall is the ratio of correctly predicted positive observations to the total number of actual positive samples. The mathematical formulation is defined in Eq. (15):

$$Recall = \frac{TP}{TP+FN} \times 100 \quad (15)$$

- **F1-Score**

Macro F1-score is calculated using the harmonic mean of both the precision and recall, serving as the primary metric to ensure robust evaluation despite the heavy class imbalance in CICIoT-2023 dataset. The macro F1-score is defined as follows in Eq. (16).

$$F1 - Score = \frac{2*(Recall*Precision)}{Recall+Precision} \times 100 \quad (16)$$

High classification performance suggests that the proposed model identifies structural anomalies as deviation from the benign traffic flow, proving high AUC shows strong discrimination under evaluated split, absence of leakage established through data-split rather than ROC performance. To ensure fair baseline comparison, all compared models (GA-GRU, CNN-GRU, GMDDPM-CT and RAID-KL) were locally re-implemented under exact same data split of 80/20, fixed budget of 10 features, Adam optimizer under 50-trial optimization and random seed of 42, enforcing the exact layer architecture specified in their respective original source papers.

4.2 Computational resource efficiency comparison

Table 4 demonstrates the computational analysis of the proposed SFSA-Net against various baseline models such as standard DNN, XGBoost, CNN and LSTM, where all baselines received equal grid-search tuning to ensure optimization parity, which provides significant improvements in terms of computational efficiency on both the UNSW-NB15 and CICIoT-2023 datasets.

Baseline models used a uniform width of 128 neurons across a search range of [0.001,0.01] learning rates and 2-4 layers. By utilizing SHAP-based feature selection, the SFSA-Net manages to optimize the memory consumption by reducing the number of input variables, resulting in significantly

This article is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.
License details: https://creativecommons.org/licenses/by-sa/4.0/

Table 3. Hyperparameter setting

Hyperparameters	Setting
Learning rate	0.001
Batch size	UNSW-NB15:64, CICIoT-2023: 4,096
Total training epochs	50
Optimization algorithm	Adam (stable convergence verified through smooth loss-landscape trajectories)
Loss function	Dynamically weighted categorical cross-entropy
Input feature layer dimension	d features
Hidden layer structure	3 dense layers ($128 \rightarrow 64 \rightarrow 32$ neurons) UNSW-NB15: $10 \rightarrow BN \rightarrow 128 \rightarrow 64 \rightarrow 32 \rightarrow 10$=12,074 parameters, CICIoT-2023: $10 \rightarrow BN \rightarrow 128 \rightarrow 64 \rightarrow 32 \rightarrow 8$=12,008 parameters
Output layer structure	C classes (Softmax activation)
Hidden layer activation	ReLU
Dropout rate	0.3
Layer normalization	Batch normalization (stabilize high-variance tabular IoT traffic)
Weight initialization	Glorot uniform initialization (biases initialized to 0.01)
Grid-Search tuning	50 total iterations evaluated per baselines and proposed model
Re-implemented Baselines (GA-GRU [22], CNN-GRU [23], GMDDPM-CT [24], RAID-KL [26])	Locally executed under 50 optimization trials using identical 80/20 training and testing partitioning, hyperparameters and shared dataset pipelines.
SHAP execution seed	42
SHAP explanation scope	8 traffic categories (class-wise)
Data balancing	SMOTE oversampling (Strictly for training-only, no evaluation leakage)
Worms target	140 original samples→1,400 balanced samples (Ratio=10.0)
SMOTE neighbourhood size	k-neighbors=5
Offline SHAP time	3.8 minutes (Decoupled from runtime inference)
Offline Selection time	12.4 seconds (Executed once during training stage)
Profiling thread count	Locked single-thread execution layout (Thread count=1)
Latency measurement protocol	Sequential input streaming with a 10-repetition warm-up loop
Data loading and preprocessing overhead	UNSW-NB15: 4.2 seconds, CICIoT-2023: 18.6 seconds
Core network training time	UNSW-NB15: 124.8 seconds, CICIoT-2023: 154.2 seconds
Total end-to-end training execution time	UNSW-NB15: 369.4 seconds, CICIoT-2023: 413.2 seconds

compressed serialized model sizes, parameter counts and training times as given in the evaluation profile of Table 4. Additionally, the proposed model demonstrates a low computational footprint during evaluation because it delivers an inference simulation time of less than 0.1 ms evaluated across 1,000 independent testing samples (with actual latency CI bounds of ± 0.004 ms for UNSW-NB15 and ± 0.002 ms for CICIoT-2023), while maintaining stable CPU utilization of $11.30\% \pm 0.15\%$ and $8.85\% \pm 0.12\%$. The one-time overhead cost needed to perform feature selection offline and calculate SHAP values was 12.4 seconds and 3.8 minutes respectively, which was completely decoupled from runtime inference simulation, whereas the grid search optimization for all the baseline models was set across a fixed 50 iterations to ensure optimization parity. The evaluation profile configuration was established using fixed configuration of 50 total grid-search iterations evaluated over 10 trial blocks with 2 repetitions of each fold, showing total of 20 paired fold-level observations (k=10, Repetitions=2, Runs=20), while runtime latency boundaries were measured across 1,000 individual testing records using single-thread layout (thread count=1) across CPU-only execution threads. This evaluation layout enforced a sequential latency with batch size of 1 evaluated across 1,000 independent testing records with a 10-repetition operational warm-up process and a blocked thread synchronization technique to prevent execution delay distortion, while reporting full median, SD and P95 latency distribution within matrix records. The proposed model restricts peak process runtime memory to 42.80 MB on UNSW-NB15 and 31.50 MB on CICIoT-2023 dataset. Replication pipeline processed raw source tracking logs (UNSW-NB15_LISTD_Files.csv and CSVs.zip) utilizing locked REI feature sets dbytes, sbytes, dloss, sloss, ct_state_ttl, ct_srv_src, sttl, dttl, swin and ct_srv_dst for UNSW-NB15 dataset and Mirai_Traffic_Entropy, Min_Time_Delta, BruteForce_Att, Flow_Duration,

This article is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.
License details: https://creativecommons.org/licenses/by-sa/4.0/

Total_Bwd_Packets, Bwd_Packet_Length_Max, Fwd_Packet_Length_Max, TCP_SYN_Flag, DDoS_Packet_Rate and Total_Fwd_Packets for CICIoT-2023 dataset. Compared to other models, the SFSA-Net effectively handles the high-dimensional network traffic issue. A simulated resource-efficient execution profile is specifically designed for resource-limited settings. Consequently, SFSA-Net provides near-optimal accuracy at a very low computational cost, making the security solution applicable as a background process.

4.3 Performance enhancement on minority attack classes

Tables 5 and 6 illustrate the performance gains achieved by the proposed SFSA-Net compared to fully optimized, traditional baseline configurations for minority class detection. The intervals in both Table 5 and Table 6 represents mean ± standard deviation computed across secondary cross-validation using 20 fold-level observations. While the standard DNN fails to effectively detect minority attacks such as worms (showing an F1-score of only 38.60%) or backdoors due to absence of class-specific features, the proposed model overcomes this limitation by identifying protocol-specific signatures. Utilizing the proposed SFSA-Net model for attack-type feature selection improves the recall rate for worms by 39.25 percentage points on the UNSW-NB15 dataset to achieve 81.40% ± 0.11%. In the CICIoT-2023 dataset, the proposed model exhibits high performance when dealing with Web-based attacks, which achieves a recall of 93.80%±0.04% to secure net gain of 21.70 percentage points. All reported uncertainty intervals show Mean±Standard Deviation (SD) calculated over 20 paired fold-level observations. Consequently, deploying SHAP-based feature subsets uncovers subtle anomalies that are typically overlooked by standard models due to majority class bias, supported by log-transformed confusion matrices which clearly highlight the performance of minority classes.

Table 4. Comparison of Computational Resource Efficiency of the Proposed model for both datasets

Model	Parameter (M)	Memory Usage (MB)	GPU optimization-loop time (s)	Serialized model size (MB)	Inference Latency (Sequential, Batch=1) ms	Approximate Mean±2 SD latency range (ms)	P95 (ms)	CPU Utilization (%)	Peak Process Memory (MB)
UNSW-NB15									
Standard DNN	0.052	22.12	345.2	0.1984	0.137±0.008	[0.121-0.153]	0.150	24.45±0.35	145.20
XGBoost	0.124	35.80	128.4	0.4730	0.210±0.012	[0.186-0.234]	0.230	28.30±0.42	188.50
CNN	0.086	18.40	512.6	0.3281	0.125±0.006	[0.113-0.137]	0.135	22.05±0.28	132.80
LSTM	0.215	25.10	842.1	0.8202	0.345±0.018	[0.310-0.380]	0.375	31.15±0.55	164.10
Proposed SFSA-Net	0.0121	3.62	124.8	0.0482 (48.19 KB)	0.065±0.004	[0.057-0.073]	0.072	11.30±0.15	42.80
CICIoT-2023									
Standard DNN	0.435	7.02	4821.5	1.6594	0.185±0.010	[0.165-0.205]	0.201	21.05±0.30	98.40
XGBoost	0.892	9.45	1842.1	3.4027	0.250±0.014	[0.223-0.277]	0.273	25.40±0.38	124.60
CNN	0.612	6.16	6140.2	2.3346	0.072±0.005	[0.062-0.082]	0.080	15.05±0.22	85.10
LSTM	1.420	8.80	11450.8	5.4169	0.410±0.022	[0.367-0.453]	0.446	29.80±0.61	110.20
Proposed SFSA-Net	0.0120	0.77	154.2	0.0479 (47.93 KB)	0.041±0.002	[0.037-0.045]	0.044	8.85±0.12	31.50

This article is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.
License details: https://creativecommons.org/licenses/by-sa/4.0/

Table 5. Performance Enhancement on Minority Attack Classes using UNSW-NB15 Dataset (As secondary evaluation)

Attack Category	Metric	Standard DNN	Proposed SFSA-Net	Improvement (percentage points)
UNSW-NB15				
Backdoor	Recall	54.20±0.14	86.75±0.08	+32.55
	F1-Score	51.05±0.15	82.15±0.09	+31.10
Worms	Recall	42.15±0.18	81.40±0.11	+39.25
	F1-Score	38.60±0.19	79.55±0.12	+40.95
Shellcode	Recall	70.30±0.11	91.20±0.06	+20.90
	F1-Score	68.45±0.12	88.90±0.07	+20.45

Table 6. Performance Enhancement on Minority Attack Classes Using CICIoT-2023 Dataset (As secondary evaluation)

Attack Category	Metric	Standard DNN	Proposed SFSA-Net	Improvement (percentage points)
CICIoT-2023				
Web-based	Recall	72.10±0.10	93.80±0.04	+21.70
	F1-Score	69.45±0.11	91.65±0.05	+22.20
Brute Force	Recall	75.35±0.09	95.10±0.03	+19.75
	F1-Score	72.80±0.11	93.55±0.04	+20.75

Table 7. Run-indices validation analysis of proposed SFSA-Net across both datasets (As secondary evaluation)

Dataset	Indices	Standard DNN	XGBoost	CNN	LSTM	Proposed SFSA-Net
UNSW-NB15	1	98.29	97.58	97.47	97.57	98.63
	2	98.44	98	97.63	96.64	98.51
	3	98.09	97.55	96.92	96.38	98.63
	4	98.25	98.51	96.98	96.67	98.69
	5	98.39	97.66	97.35	96.82	98.62
	6	98.78	97.98	98.02	97.09	98.72
	7	98.44	97.72	97.38	97.22	99.32
	8	98.1	97.51	97.59	97.19	98.94
	9	98.35	97.38	96.92	96.94	98.7
	10	97.77	97.89	96.83	97.29	98.74
	Mean	98.29±0.26	97.78±0.32	97.31±0.38	96.98±0.35	98.75±0.23
CICIoT-2023	1	97.7	94.34	96.26	95.6	98.7
	2	97.84	94.58	96.36	95.05	98.55
	3	97.51	94.32	95.94	94.9	98.7
	4	97.66	94.87	95.98	95.07	98.78
	5	97.79	94.38	96.2	95.16	98.69
	6	98.16	94.57	96.59	95.32	98.82
	7	97.84	94.42	96.21	95.39	99.55
	8	97.52	94.3	96.34	95.37	99.08
	9	97.76	94.22	95.94	95.22	98.79
	10	97.21	94.52	95.89	95.43	98.84
	Mean	97.69±0.25	94.45±0.18	96.17±0.22	95.25±0.21	98.85±0.28

These results demonstrate that the attack-wise optimization provides a powerful approach for creating an advanced IDS. With a sample size of N=20 paired fold-level observations, the performance improvement was evaluated using the parametric paired t-test as primary statistical testing, while a Wilcoxon signed-rank test was used only as a secondary non-parametric sensitivity analysis, acknowledging this bounded population size as a standard limitation. The parametric t-test p-values are used for the final significance testing with Holm-Bonferroni corrections applied systematically to the t-test p-value family within Table 8 to control family-wise error rate. The results were presented in Table 8, where the Wilcoxon sensitivity analysis produced W-statistics (ranging from 0 to 13) with corresponding two-sided p-values shown in Table 8 and strong Rank-Biserial correlation effect sizes ($r > 0.8762$) evaluated directly over accuracy. To ensure mathematical verification, the complete raw pairwise differences, standard deviation and 95% CI were

This article is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.
License details: https://creativecommons.org/licenses/by-sa/4.0/

Table 8. Statistical analysis of proposed SFSA-Net model using multiple test to show significance across both datasets (K=10, Repetitions =2, Runs=20) (As secondary evaluation)

Models	Data set	Mean Diff	SD Diff	95% CI (Mean Difference)	W-statistics	Wilcoxon p-value	t-value	p-value (t test)	Rank-Biserial correlation (r)	Holm Rank	Holm-Bonferroni correction threshold	Corrected p-value (t-test)
Proposed vs. Standard DNN	UNSW-NB15	0.51	0.446	[0.30-0.72]	13	5.72e-5	5.11388	6.1686e-5	0.8762	4	0.0500	6.1686e-5
	CICIoT-2023	0.771	0.447777	[0.56-0.98]	1	2.13e-6	7.7003	2.9411e-7	0.9905	4	0.0500	2.9411e-7
Proposed vs. XGBoost	UNSW-NB15	1.06	0.42155	[0.86-1.26]	0	1.91e-6	11.245292	7.6981e-10	1.0000	3	0.0250	1.5396e-9
	CICIoT-2023	4.2305	0.3626	[4.06-4.40]	0	1.91e-6	52.1769	5.5665e-22	1.0000	1	0.0125	2.2266e-21
Proposed vs. CNN	UNSW-NB15	1.4185	0.479553	[1.19-1.64]	0	1.91e-6	13.228406	4.9120e-11	1.0000	2	0.0167	1.4736e-10
	CICIoT-2023	2.2	0.456161	[1.99-2.41]	0	1.91e-6	21.570592	8.0233e-15	1.0000	3	0.0250	1.6047e-14
Proposed vs. LSTM	UNSW-NB15	1.7395	0.469664	[1.52-1.96]	0	1.91e-6	16.563514	9.5052e-13	1.0000	1	0.0125	3.8021e-12
	CICIoT-2023	3.5315	0.386636	[3.35-3.71]	0	1.91e-6	40.848145	5.6007e-20	1.0000	2	0.0167	1.6802e-19

calculated using the entire population of 20 fold-level observations and are provided in data files located in our public repository, computed via a two-tailed Student's t-distribution critical value ($t_{crit} = 2.093$) to ensure perfect reproducibility. The mean differences in Table 8 derived from full 20 fold-level cross-validation observations, where CICIoT-2023 dataset proposed models mean is 98.87% and Standard DNN is 98.10% shows 0.771, which represents larger evaluation population size than 10-run samples in Table 7. Hence, the statistically significant performance improvement was established after applying step-wise Holm-Bonferroni multiple-comparison procedure. The correction family bounds exactly $m = 4$ independent statistical tests evaluated per dataset block (testing the proposed model across four different baseline architectures such as standard DNN, XGBoost, CNN and LSTM) and was applied systematically to the paired t-test p-values within Table 8 to confirm mean-order significance conclusion under a controlled family-wise error rate, with adjusted values defined as $\min(raw\ p \times multiplier, 1)$.

The evaluation of the SFSA-Net model was structured around performance metrics and explainability outputs. Fig. 2 illustrates the confusion matrix for the UNSW-NB15 dataset, where true positives exceed 98%.

Fig. 3 presents the confusion matrix for the CICIoT-2023 dataset, showing that classification errors are confined to similar attack classes. To evaluate class discrimination, Fig. 4 displays the ROC curve for the UNSW-NB15 dataset, confirming high AUC scores. Table 7 shows the 10 fold cross-validation results. Table 8 illustrates the statistical analysis of the proposed across 20 paired fold-level observations. Fig. 5 provides the ROC curve for the CICIoT-2023 dataset, verifying prediction stability for minority attacks. Finally, Fig. 6 maps the local

This article is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.
License details: https://creativecommons.org/licenses/by-sa/4.0/

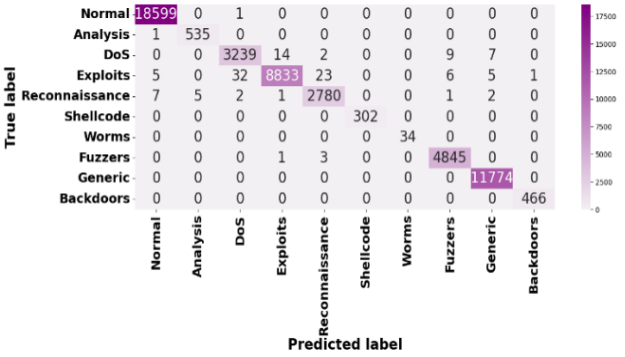

Figure. 2 Confusion Matrix for UNSW-NB15

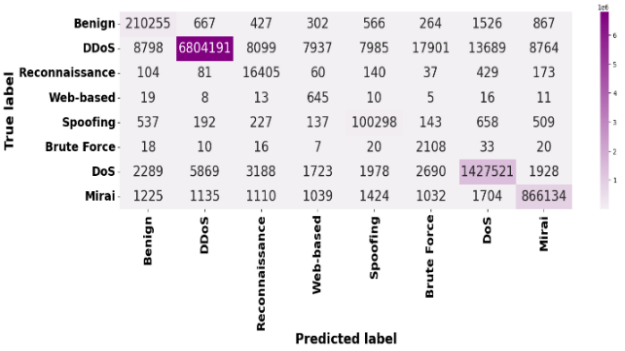

Figure. 3 Confusion Matrix for CICIoT-2023

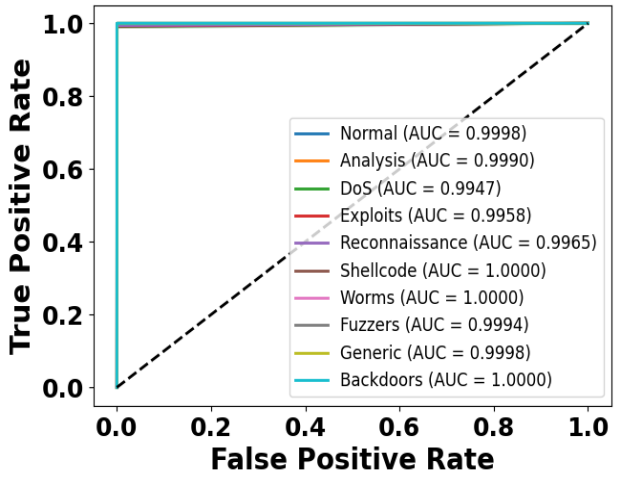

Figure. 4 ROC curve for UNSW-NB15 dataset

SHAP feature importance for specific samples, while Fig. 7 presents the global SHAP summary plot to reveal the most influential network features.

4.4 Ablation study

Table 9 represents the ablation study under secondary evaluation, which demonstrates that the SHAP-based feature selection module is the primary contributor to the proposed model's reliability and resource efficiency. A comparison between the baseline SFSA-Net approach (without SHAP) and

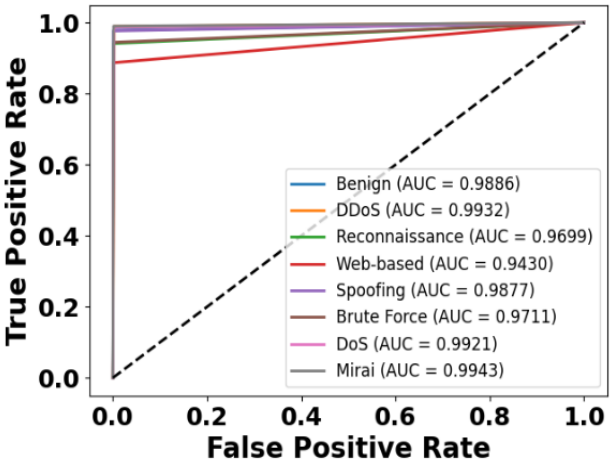

Figure. 5 ROC curve for CICIoT-2023 dataset

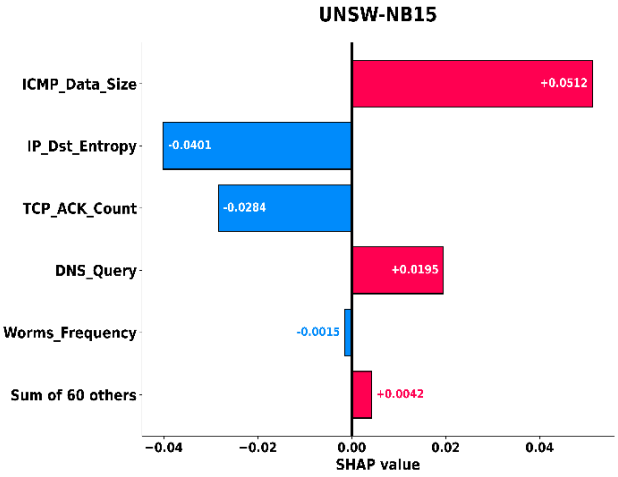

Figure. 6 Local SHAP analysis results for the UNSW-NB15 dataset

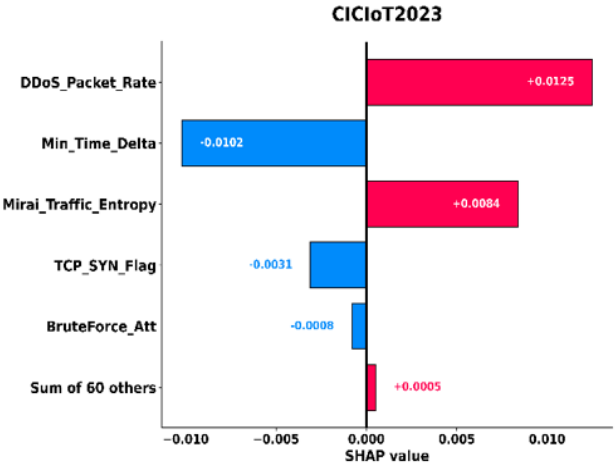

Figure. 7 Global SHAP analysis results for the CICIoT-2023 dataset

(with SHAP) eliminates the high-dimensional noise in the IoT traffic dataset, which provides higher level of accuracy. Specifically, in the CICIoT-2023 dataset, the SHAP-based selector reduces the number of input

This article is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.
License details: https://creativecommons.org/licenses/by-sa/4.0/

Table 9. Ablation Study of the Proposed Model Using both datasets (As secondary evaluation)

Configuration	Feature Count	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Stability Index (σ)[1]
UNSW-NB15						
SFSA-Net Case 1 (Without SHAP)	47 (Full)	97.90±0.12	97.75±0.15	97.95±0.11	97.85±0.13	± 1.45
Mean-SHAP baseline	10 (Top K)	97.20±0.18	97.05±0.21	97.10±0.19	97.08±0.20	±0.95
Mutual information baseline	10 (Top K)	96.05±0.24	95.88±0.28	95.92±0.26	95.90±0.25	±1.35
RFE baseline	10 (Top K)	96.35±0.22	96.12±0.25	96.28±0.21	96.20±0.23	±1.20
Proposed SFSA-Net Case 2 (With SHAP)	10 (Top K)	98.75±0.23	98.42±0.07	98.58±0.09	98.50±0.06	± 0.25
Performance Gap	-78.7% features	+0.85	+0.67	+0.63	+0.65	-1.20
CICIoT-2023						
SFSA-Net Case 1 (Without SHAP)	389 (full)	97.69±0.25	97.35±0.28	97.60±0.22	97.47±0.24	± 2.10
Mean-SHAP baseline	10 (Top K)	97.48±0.19	97.30±0.22	97.38±0.18	97.34±0.20	±1.12
Mutual information baseline	10 (Top K)	96.42±0.31	96.18±0.35	96.30±0.29	96.24±0.32	±1.40
RFE baseline	10 (Top K)	96.85±0.24	96.55±0.27	96.72±0.23	96.63±0.25	±1.30
Proposed SFSA-Net Case 2 (With SHAP)	10 (Top K)	98.85±0.28	98.72±0.08	98.88±0.11	98.80±0.09	± 0.40
Performance Gap	-97.4% features	+1.16	+ 1.37	+ 1.28	+ 1.33	-1.70

Table 10. Re-implemented Comparative Analysis of the Proposed model using UNSW-NB15 and CICIoT-2023 Datasets (As secondary evaluation)

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
UNSW-NB15				
GA-GRU [22]	96.15±0.16	96.42±0.12	96.08±0.18	96.25±0.14
CNN-GRU [23]	97.40±0.11	96.85±0.09	97.12±0.14	96.98±0.12
Proposed SFSA-Net	98.75±0.23	98.42±0.07	98.58±0.09	98.50±0.06
CICIoT-2023				
GMDDPM-CT [24]	97.10±0.22	96.92±0.19	97.35±0.25	97.13±0.20
RAID-KL [26]	97.65±0.14	97.48±0.11	97.60±0.16	97.54±0.13
Proposed SFSA-Net	98.85±0.28	98.72±0.08	98.88±0.11	98.80±0.09

features by 97.4%, which not only eliminates feature redundancy but also increases the F1-score by 1.33 percentage points to achieve a final score of 98.80%±0.09%. Under an identical set of feature budget consisting of 10 features, the complete REI framework was evaluated against other approaches including Mean-SHAP ranking, Mutual Information and Recursive Feature Elimination (RFE), where it consistently minimized sensitivity toward context while improving minority class detection. Moreover, the stability index exhibits that the classifier "with SHAP" has higher robustness with a standard deviation of only (± 0.40%) compared to the classifier "without SHAP" at (±2.10%) due to the unstable performance of the baseline network when trained on high-dimensional data.

Therefore, the proposed SFSA-Net approach is not only a standard classifier but a specialized resource-aware system that identifies the most stable explainable index for IoT security.

4.5 Comparative analysis

Table 10 depicts the re-implemented comparative analysis between the proposed SFSA-Net approach and other state-of-the-art approaches like GA-GRU [22], CNN-GRU [23], GMDDPM-CT [24] and RAID-KL [26] were reimplemented locally by utilizing exact data pipeline, dataset partitioning and random seeds. As shown in Table 10, the proposed SFSA-Net yields a higher global accuracy of 98.75%±0.23% compared to the baseline CNN-GRU (97.40%±0.11%) on the UNSW-NB15 dataset. This absolute improvement over independent standard DNN Table 7 and architectural ablation baselines Table 9 was further augmented by proposed SFSA-

This article is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.
License details: https://creativecommons.org/licenses/by-sa/4.0/

Net's superior F1-score (which addresses baseline class imbalance struggles), lower computational footprint and built-in explainability metrics. Using SHAP-based feature engineering, the proposed model mitigates the black-box problem by establishing a REI that improves precision and recall. In high-dimensional dataset such as CICIoT-2023, the proposed SFSA-Net achieved the accuracy of 98.85%±0.28%, shows existing modern generative architectures including GMDDPM-CT (97.10%±0.22%) and RAID-KL (97.65%±0.14%). These results demonstrate that the proposed approach scales effectively with increasing dataset complexity by successfully isolating high-signal feature subsets from the original 389 features. Unlike baseline models, the proposed approach achieves results based on clear feature attributions. Thus, it optimizes resources for robust detection across the evaluated datasets which demonstrating high feature scalability for low-footprint environments.

4.6 Research implication

The research implication of the proposed SFSA-Net model is described as follows:

- These results illustrate that it is essential to separate minority class from majority class bias to detect subtle protocol-specific anomalies.
- Combining SHAP with a three-stage stability evaluation demonstrates that XAI transforms from a simple diagnostic tool into an active architectural optimization mechanism.
- By removing up to 97.4% of high-dimensional traffic features while simultaneously achieving near-optimal accuracy, the results demonstrate the feasibility for achieving compression of DNN without compressing their security resilience, making them suitable for resource-constrained IoT deployments.

4.7 Discussion

The evaluation of the proposed SFSA-Net approach proves that switching from simple feature importance to stability-aware feature engineering plays a vital role in ensuring the security of dynamic IoT environments. Unlike conventional DL approaches, which often behave as black-box nature with relatively high computational costs, the proposed algorithm uses SHAP analysis to produce an REI. This approach ensures that consistent selection of discriminative features is designed for identifying diverse attack types including DoS and backdoor attacks. Moreover, the proposed approach maintains high accuracy even after eliminating 78.7% and 97.4% of the features across UNSW-NB15 and CICIoT-2023 datasets, which mitigates overfitting by stripping away the high-dimensional environmental noise. The network convergence shows stable optimization curves in which training and validation losses decrease together, indicating reduced overfitting after SHAP-based feature reduction. In addition, a significant improvement was observed in minority class detection through the identification of class-specific protocols, which increases the recall for worm attacks from 42.15% to 81.40% (a 39.25 percentage points improvement). By focusing on attack-specific and localized feature spaces rather than static global ranking, the network uncovers subtle protocol anomalies that are traditionally masked by majority class bias. Furthermore, reduced serialized model size reduction and sub-0.1ms latency confirm that SFSA-Net is computationally efficient.

5. Conclusion

The proposed SFSA-Net approach combined a DNN with SHAP stability analysis for IoT intrusion detection. This manuscript addressed the limitations of existing studies, specifically high-dimensional and a lack of explainability. The proposed approach prioritized features with the highest discrimination capability and situational stability. Consequently, the proposed model achieved high classification accuracy of 98.75%±0.23% on UNSW-NB15 and 98.85% ± 0.28% on CICIoT-2023 datasets, respectively. Moreover, by dynamically isolating attack specific features, it improved detection rates significantly for minority attack classes such as 39.25 percentage points in recall for worm attacks. In conclusion, this study demonstrates that model explainability is a fundamental architectural requirement for dependable intrusion detection systems (IDS), providing the transparency needed to validate algorithmic decision pathways against evolving network threat paradigms. The use of waterfall plots and global importance scores allowed analysts to trust and track automated decisions. The proposed SFSA-Net approach offered a computationally efficient and highly accurate solution for detecting sophisticated attacks in IoT networks. Future research will explore the applicability of the proposed concept within federated-learning architectures.

This article is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.
License details: https://creativecommons.org/licenses/by-sa/4.0/

Conflicts of interest

The authors declare no conflict of interest.

Author contributions

Conceptualization, SP and PM; methodology, SP; software, PM; validation, SPand PM; formal analysis, SP; investigation, PM; resources, SP; data curation, PM; writing—original draft preparation, SP; writing—review and editing, PM; visualization, SP; supervision, SP; project administration, SP; funding acquisition, PM.

Acknowledgments

The authors would like to express their sincere gratitude to the Department of Computer Science and Engineering and Sai Vidya Institute of Technology, Bengaluru, for providing the necessary guidance, resources, and a supportive research environment throughout the course of this work. The authors also acknowledge the valuable encouragement, constructive suggestions, and technical support received during the research, which significantly contributed to the successful completion of this paper. Finally, the authors extend their heartfelt appreciation to everyone who directly or indirectly contributed to this research.

References

[1] M. A. Hossain, "Deep learning-based intrusion detection for IoT networks: a scalable and efficient approach", *EURASIP Journal on Information Security*, Vol. 2025, No. 1, Art. no. 28, 2025, doi: 10.1186/s13635-025-00202-w.

[2] B. Olanrewaju-George and B. Pranggono, "Federated learning-based intrusion detection system for the internet of things using unsupervised and supervised deep learning models", *Cyber Security and Applications*, Vol. 3, Art. no. 100068, 2025, doi: 10.1016/j.csa.2024.100068.

[3] M. Benmalek and A. Seddiki, "Particle swarm optimization-enhanced machine learning and deep learning techniques for Internet of Things intrusion detection", *Data Science and Management*, Vol. 8, No. 4, pp. 423-435, 2025, doi: 10.1016/j.dsm.2025.02.005.

[4] H. Zhang, "Development of an intelligent intrusion detection system for IoT networks using deep learning", *Discover Internet of Things*, Vol. 5, No. 1, Art. no. 74, 2025, doi: 10.1007/s43926-025-00177-7.

[5] M. Termos, Z. Ghalmane, M. A. Brahmia, A. Fadlallah, A. Jaber and M. Zghal, "ADAP-GNN: Adaptive property-aware graph neural network for intrusion detection in IoT networks", *Computers and Electrical Engineering*, Vol. 133, Art. no. 111051, 2026, doi: 10.1016/j.compeleceng.2026.111051.

[6] S. Fraihat, Q. Yaseen, Y. Sanjalawe, A. Abu-Errub, S. N. Makhadmeh and M. A. Al-Betar, "Intrusion detection in industrial internet of things network using feature optimization and hybrid deep learning", *Discover Internet of Things*, Vol. 6, No. 1, Art. no. 34, 2026, doi: 10.1007/s43926-026-00284-z.

[7] R. Ponnumani, N. Vasudeva, T. Elumalai, P. Kaliyaperumal, B. Balusamy and F. Benedetto, "A multi-stage framework for scalable and context-aware intrusion detection in IoT-cloud systems using deep latent modeling and graph-based attack classification", *Computers and Electrical Engineering*, Vol. 131, Art. no. 110949, 2026, doi: 10.1016/j.compeleceng.2026.110949.

[8] I. Bharathi, V. Sonai and S. S. Sridevi, "Quantum-driven enhanced machine learning algorithm for intrusion detection in Internet of things environment", *EPJ Quantum Technology*, Vol. 13, No. 1, Art. no. 20, 2026, doi: 10.1140/epjqt/s40507-026-00463-5.

[9] D. Kasina, V. S. Reddy, S. Rajamanickam, J. Jose and B. Palani, "A novel intrusion detection system for class imbalance datasets using hybrid sampling with deep learning techniques", *Information Sciences*, Vol. 741, Art. no. 123208, 2026, doi: 10.1016/j.ins.2026.123208.

[10] J. P. Sahoo, B. Kar, A. M. Abdelmoniem and D. Chatzopoulos, "Choir-IDS: A federated learning framework for fidelity-calibrated explainable intrusion detection system for edge-IoT networks", *Information Fusion*, Vol. 125, Art. no. 103473, 2026, doi: 10.1016/j.inffus.2025.103473.

This article is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.
License details: https://creativecommons.org/licenses/by-sa/4.0/

[11] D. D. Bikila and J. Čapek, "FedPGO: A personalized federated learning framework with gradient orthogonalization for IoT intrusion detection", *Computer Networks*, Vol. 280, Art. no. 112210, 2026, doi: 10.1016/j.comnet.2026.112210.

[12] R. Kumar and M. Swarnkar, "QuIDS: A quantum support vector machine-based intrusion detection system for IoT networks", *Journal of Network and Computer Applications*, Vol. 234, Art. no. 104072, 2025, doi: 10.1016/j.jnca.2024.104072.

[13] D. Huiying, "Research on optimizing network intrusion detection using deep learning and big data in intelligent elderly care", *Computer Standards & Interfaces*, Vol. 97, Art. no. 104136, 2026, doi: 10.1016/j.csi.2026.104136.

[14] M. Sajid, K. R. Malik, A. Almogren, T. S. Malik, A. H. Khan, J. Tanveer, *et al.*, "Enhancing intrusion detection: a hybrid machine and deep learning approach", *Journal of Cloud Computing*, Vol. 13, No. 1, Art. no. 123, 2024, doi: 10.1186/s13677-024-00685-x.

[15] H. Asgharzadeh, A. Ghaffari, M. Masdari and F. S. Gharehchopogh, "An intrusion detection system on the Internet of Things using deep learning and multi-objective enhanced gorilla troops optimizer", *Journal of Bionic Engineering*, Vol. 21, No. 5, pp. 2658-2684, 2024, doi: 10.1007/s42235-024-00575-7.

[16] M. Samantaray, R. C. Barik and A. K. Biswal, "A comparative assessment of machine learning algorithms in the IoT-based network intrusion detection systems", *Decision Analytics Journal*, Vol. 11, Art. no. 100478, 2024, doi: 10.1016/j.dajour.2024.100478.

[17] S. Walling and S. Lodh, "Enhancing IoT intrusion detection through machine learning with AN-SFS: a novel approach to high performing adaptive feature selection", *Discover Internet of Things*, Vol. 4, No. 1, Art. no. 16, 2024, doi: 10.1007/s43926-024-00074-5.

[18] A. Zohourian, S. Dadkhah, H. Molyneaux, E. C. P. Neto and A. A. Ghorbani, "IoT-PRIDS: Leveraging packet representations for intrusion detection in IoT networks", *Computers & Security*, Vol. 146, Art. no. 104034, 2024, doi: 10.1016/j.cose.2024.104034.

[19] J. Saikam and K. Ch, "EESNN: Hybrid deep learning empowered spatial-temporal features for network intrusion detection system", *IEEE Access*, Vol. 12, pp. 15930-15945, 2024, doi: 10.1109/ACCESS.2024.3350197.

[20] M. K. Nallakaruppan, S. R. K. Somayaji, S. Fuladi, F. Benedetto, S. K. Ulaganathan and G. Yenduri, "Enhancing security of host-based intrusion detection systems for the Internet of Things", *IEEE Access*, Vol. 12, pp. 31788-31797, 2024, doi: 10.1109/ACCESS.2024.3355794.

[21] S. Ali, R. Ghazal, N. Qadeer, O. Saidani, F. Alhayan, A. Masood, *et al.*, "A novel approach of botnet detection using hybrid deep learning for enhancing security in IoT networks", *Alexandria Engineering Journal*, Vol. 103, pp. 88-97, 2024, doi: 10.1016/j.aej.2024.05.113.

[22] E. M. Maseno, Y. Sun and Z. Wang, "GA-optimized deep learning intrusion detection framework with LIME explainability for IoT networks", *Evolutionary Intelligence*, Vol. 19, No. 1, Art. no. 23, 2026, doi: 10.1007/s12065-025-01133-8.

[23] R. Jablaoui, O. Cheikhrouhou, M. Hamdi and N. Liouane, "Deep learning enabled intrusion detection system for IoT security", *EURASIP Journal on Wireless Communications and Networking*, Vol. 2025, No. 1, Art. no. 66, 2025, doi: 10.1186/s13638-025-02477-6.

[24] G. Zhu, Y. Yu, Z. Li and Y. Dai, "GMDDPM-CT: Advancing network intrusion detection through diffusion-based sample synthesis and deep learning", *ICT Express*, 2026, doi: 10.1016/j.icte.2026.01.014.

[25] J. Fu, L. Wang, J. Ke, K. Yang and R. Yu, "TSIDS: Spatial-temporal fusion gating multilayer perceptron for network intrusion detection", *Expert Systems with Applications*, Vol. 263, Art. no. 125687, 2025, doi: 10.1016/j.eswa.2024.125687.

This article is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.
License details: https://creativecommons.org/licenses/by-sa/4.0/

[26] O. D. Okey, S. Dadkhah, D. Z. Rodríguez and J. H. Kleinschmidt, "Explainable resource-aware IoT security model via knowledge distillation and adaptive loss function optimization", *Expert Systems with Applications*, Vol. 302, Art. no. 130460, 2026, doi: 10.1016/j.eswa.2025.130460.

[27] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)", In: *Proc. of 2015 Military Communications and Information Systems Conference (MilCIS)*, pp. 1-6, 2015, doi: 10.1109/MilCIS.2015.7348942.

[28] E. C. P. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu and A. A. Ghorbani, "CICIoT2023: A real-time dataset and benchmark for large-scale attacks in IoT environment", *Sensors*, Vol. 23, No. 13, Art. no. 5941, 2023, doi: 10.3390/s23135941.

This article is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.
License details: https://creativecommons.org/licenses/by-sa/4.0/