JOURNAL OF BASIC SCIENCE AND ENGINEERING

LIGHTWEIGHT AND EXPLAINABLE ENSEMBLE DEEP LEARNING FRAMEWORK FOR IOT DEVICE TYPE IDENTIFICATION

[1]Indira, [2]Dr. Sampath A K

[1]Research Scholar, School of Computer Science and Engineering, Presidency University, Bangalore, India

[2]Professor, School of Computer Science and Engineering, Presidency University, Bangalore, India

DOI: https://doie.org/10.10399/JBSE.2025132962

Abstract- The widespread adoption of Internet of Things (IoT) devices in critical domains has necessitated the development of efficient, accurate, and interpretable device type identification systems. To address the limitations of conventional models in resource-constrained environments, we propose a Lightweight and Explainable Ensemble Deep Learning Framework for IoT device type identification. The framework integrates three distinct models GoogLeNet for spatial feature extraction, VGG16 for deep visual representation, and a Transformer for capturing long-range dependencies whose outputs are fused through a dynamic confidence-based fusion strategy that adaptively weighs predictions based on model confidence. Model pruning and quantization are applied to reduce complexity while maintaining performance. A Few-Shot Learning Module is triggered when prediction confidence is low, enabling classification of rare or unseen device types with minimal data. For enhanced interpretability, the framework incorporates Sparsemax Attention and Attribution Loss to generate focused and label-aligned attention maps, and employs SHAP (SHapley Additive exPlanations) to provide both global and local feature attributions, improving transparency and trust in predictions. The ensemble's hyperparameters are fine-tuned using the Adaptive FOSSA Optimization to accelerate convergence and enhance predictive performance. Implemented in Python, the proposed model demonstrates superior performance compared to existing methods, achieving an accuracy of 0.9899, precision of 0.9903, and notably low FNR of 0.0062 and FPR of 0.0071, making it highly effective for IoT deployments.

Keywords: *GoogLeNet, VGG16, Transformer model, Few-Shot Learning, Sparsemax Attention, Adaptive FOSSA Optimization, SHAP.*

1. INTRODUCTION

A network of physical items is referred to as the Internet of Things (IoT). with integrated systems that enable them to communicate data about themselves across a wired or wireless communications channel [1]. The quick expansion of the use of Internet of Things gadgets in heterogeneous environments such as smart homes, industrial systems, and healthcare settings has raised the demand for reliable device type identification to improve network visibility and overall security [2]. This has been increasingly important with an increasing number of unknown or previously unseen devices entering the networks, often without a standard process for identification [3]. In addition, efforts focused on model generalizability and the value of thorough identification approaches that can inform changes in communication behaviors and provide facilities that support generalization across principles to differing operating environments [4].

In contemporary deployments, recognizing the kinds of IoT gadgets that are a part of a network becomes a significant challenge due to the vast, diverse range of devices, changing usage patterns, and security vulnerabilities [5]. Another challenge is that manufacturers of IoT devices do not continuously deploy updates for their devices, instead relying on users to install

firmware updates given their limited resources [6]. Moreover, malicious individuals might intentionally camouflage device traits or channel lawful devices, thereby lessening the power of identification systems and security [7]. Additionally, a significant concern here is that static identification models typically experience drops in performance over time, and operational drift caused by changing device firmware, changes to existing network configurations, and the introduction of entirely new and novel devices are all problematic for pre-existing models to easily adapt [8].

Deep learning (DL)-based models employing channel state information (CSI) are also a viable solution; convolutional neural networks (CNN) in these systems are trained from CSI micro-signals, these signals are further enhanced through contrastive learning and adaptive fusion, and are intended to operate effectively in various channel conditions and environments [9]. Another technique transforms raw network traffic converted as structured inputs into CNNs, allowing classification of known and unknown devices without the need to design features manually [10]. Furthermore, a DL-based framework, known as the Intrusion Detection System based on Devices (DIDS), was designed to intelligently learn and forecast possible threats with a multi-phase architecture [11]. Using the advantages of ensemble learning, DIS-IoT utilizes a fully connected integration layer that integrates outputs from different DL classifiers, producing an ensemble DL framework and a new integrated Internet of Things intrusion detection system environments [12].

The quick expansion of Internet of Things gadgets within smart environments has significantly raised the need for accurate device type identification to maintain both network efficiency and security. Legacy methods cannot frequently respond to the scale, diversity, and dynamism of current IoT networks [13]. In addition to this, the adaptive nature of networks in new and previously unseen devices being injected into the device landscape of a live network requires a system to learn in an incremental manner without reducing performance [14]. In contrast, identity authentication methods based on physical layer traits generally have a lighter footprint and are more efficient, which is thought to support the identity authentication process in IoT with a better experience overall [15]. We proposed the work an intelligent and adaptive DL-based framework for identifying the type of IoT device, addressing the limitations of traditional static classification methods. To enhance robust identification, improved scalability, and low overhead, making it appropriate for use in complex IoT settings. The key contribution of the works as follows:

- Proposed a novel lightweight ensemble integrating GoogLeNet, VGG16, and Transformer models, optimized through pruning and quantization, achieving high classification accuracy with reduced computational cost for IoT device type identification.
- Introduced a dynamic confidence-based fusion strategy for optimal decision weighting, and incorporated a Few-Shot Learning module to handle rare or unseen IoT device types with minimal training samples.
- Enhanced interpretability using Sparsemax Attention, Attribution Loss, and SHAP-based feature attribution, while fine-tuning ensemble hyperparameters via Adaptive FOSSA Optimization for improved convergence and robustness.

The sequence of these sections is the following: The suggested framework is presented in Section 3, some pertinent research and literature studies are covered in Section 2, a thorough analysis of the observed outcomes and responds are offered in Section 4, and the study's final evaluation is provided in Section 5.

2. LITERATURE REVIEW

Some of the recent research work related to IoT device type identification is reviewed in this section.

In 2022, Chaganti *et. al.,* [16] proposed a DL-based Bidirectional-Gated Recurrent Unit-Convolutional Neural Network (Bi-GRU-CNN) model uses executable and linkable format binary file byte sequences as the input feature to identify and categorize IoT malware families. However, it considered DL model based on recurrent neural networks (RNNs) combinations to evaluate the performance of IoT malware detection and family classification, and those model performances are compared with the proposed method.

In 2023, Awajan [17] introduced a new DL based swarm for an IoT device intrusion detection solution. A four-layer deep Fully Connected (FC) network architecture is used by this clever system to identify environmental traffic that could start an attack on IoT devices that are connected. The new swarm is meant to be independent of the communication protocol to minimize installation problems. The new swarm provided reliable performance for simulating the experimental performance analysis.

In 2023, Khan et. al., [18] proposed a DL model that comprises gated recurrent units (RNN-GRU) and a recurrent neural network, which classifies assaults within the network, application, and physical layers. The existing system of detecting some intrusions in IoT networks could be improved. Several investigators have intrusion detection systems (IDS), but have only based their works on One of the three IoT architectural layers protocol, which is presents challenges for classifying attacks holistically across the IoT network.

In 2022, Shahin et. al., [19] suggested a hybrid DL model to enhance the traditional systems for network intrusion detection. The data set will first be normalized and pre-processed. Then, inferences were made based on DL models that use fully convolutional neural networks (FCNs) with gradient boosting and attention-based long short-term memory (ALSTM) models, like AdaBoost, or adaptive boost, and Extreme Gradient Boosting (XGBoost), which can identify abnormal behaviour in IoT for industry device traffic data.

In 2023, Keshk et. al., [20] developed a DL-based explainable ensemble IDS to further increase the openness and strength of Industrial Internet of Things networks (IIoT). The structure offers both Local interpretable model agnostic explanations (LIME) and Shapley additive explanations (SHAP) to provide experts with meaningful information in the quest to defend IIoT networks from attacks, and to help them build more cyber-resilient systems. However, they established their extreme learning machines (ELM) model as a baseline IDS, against which they compared their other models.

In 2022, Liu et. al., [21] propose a DL-based on model in order to identify the gadget, and focused their effort specifically in the IoT space, with the focus being on sequences of directional packet lengths, and based on a DNN (Neural Network Deep). In a sequence of packet length, the associated values represent the transmission size and the transmission direction of the packet. However, the method creates device fingerprints with m packet length sequences and builds a DNN that has convolutional deep layers, which can identify deep features from the device fingerprints.

In 2025, Jain et. al., [22] proposed a hybrid machine learning (ML) framework that employs CNNs as a method for deep feature extraction and uses the powerful classification capabilities of XGBoost. Employing a hybrid model, it can eliminate the need for manual feature engineering and can improve and expand detection using direct learning from network data at the packet level. As an Interpretability component, that is an essential component in sensitive

applications for security, we employ Techniques for Explainable AI (XAI) that use SHAP values.

In 2022, Yang et. al., [23] proposed a small-scale intrusion detection system built using and a graph of knowledge. The Key features and semantic correlations are extracted by the system. using a knowledge graph and statistical evaluation. Then, requests from IoT networks are fully processed by feature alignment and Multiview feature fusion, and Word vectors are created from requests. Finally, it designs a dedicated centered on attention CNN-BiLSTM model which is capable of identifying assaults using malicious requests, and this will exploit contextual semantic knowledge and long-distance reliance.

In 2023, Li et. al., [24] explored the DL models in physical layer identity recognition from CSI as well as develop A clever recognition scheme founded on DRL, or deep reinforcement learning. In particular, by examining fully the characteristics of the actual CSI data that was received, as well as a simple state, action, and reward metrics for the corresponding DNN of DRL, it created a verification scheme that can effectively recognize names.

In 2022, Teymournezhad et. al., [25] developed a new DL approach for counterfeit banknote identification based on security components using image processing as well as GoogLeNet. To do this goal, a few high-precision credible Images of banknotes have been used to extract security components. based on processing images and ML algorithmic. However, to training the model with GoogLeNet, it will be estimated the authenticity or level of genuineness of each of the safety components. The technique, goal, benefits, and drawbacks of the previous efforts are shown in Table 1.

Table 1 Review of various authors about the existing works

Author	Aim	Methodology	Advantage	Disadvantage
Chaganti *et. al.,* 2022 [16]	IoT malware detection and classification	Bi-GRU-CNN using ELF binary byte sequences	Effective for malware family classification	Relies heavily on RNN; limited model comparison
Awajan, 2023 [17]	Intrusion detection in IoT devices	4-layer FC network with swarm intelligence	Protocol-independent; good simulated performance	Its deployment validation
Khan *et al.,* 2023 [18]	Multi-layer IoT attack classification	RNN-GRU model	Covers physical, network, and app layers	Prior works focused only on single-layer analysis

Vol. 22, No. 1, (2025)
ISSN: 1005-0930

Shahin *et al.,* 2022 [19]	Industrial IoT intrusion detection	Hybrid ALSTM and FCN, and XGBoost/AdaBoost	Detects abnormal traffic; enhanced accuracy	Requires extensive pre-processing
Keshk *et al.,* 2023 [20]	Explainable IDS for IIoT	Ensemble DL with SHAP and LIME	Enhances transparency and robustness	SHAP baseline limits generalizability
Liu *et al.,* 2022 [21]	IoT device identification	DNN on directional packet length sequences	Accurate deep feature extraction from packet data	Fingerprinting can be data-intensive
Jain *et al.,* 2025 [22]	Hybrid ML for network attack detection	CNN for feature extraction and XGBoost, and SHAP	Eliminates manual feature engineering; interpretable	Needs real-world deployment validation
Yang *et al.,* 2022 [23]	Lightweight IDS using semantic fusion	CNN-BiLSTM with knowledge graph and Multiview features	Captures semantic context and long-range dependencies	Complex architecture
Li *et al.,* 2023 [24]	Physical-layer identity recognition	Deep Reinforcement Learning (DRL) on CSI data	Efficient and lightweight authentication	Simplified modelling may miss nuanced variations
Teymourne zhad *et al.,* 2022 [25]	Counterfeit banknote detection	GoogLeNet with image-based DL	Accurate detection of image-based features	Limited to the financial domain only

Vol. 22, No. 1, (2025)
ISSN: 1005-0930

2.1 Research Gap

Although there have been some recent developments in DL-based frameworks for IoT device type identification and intrusion detection, there are still large research gaps. Most models are not able to generalize well across different device types and communication patterns, thus limiting transferability in continuously changing environments. In addition, many of them may have catastrophic forgetting because they do not include the incremental learning component that would allow them to learn continuously as new devices are introduced into the ecosystem. Other models have high computational complexity, which also means they cannot be deployed on constrained IoT devices. Largely, it is also rare that intrusion detection focuses only on the host-based, network-based, and software application layers simultaneously, as a more holistic approach. However, the lack of explainability itself has limited explainability to domestic cases, and is inadequate for mission-critical cases. Moreover, relying on protocol-specific traffic features also reduces portability for models, while there is still a lack of lightweight options suitable for edge deployment. There is also a shortage of lightweight, low-power solutions viable for using edge-analyzing types of AI. Most models require well-labelled datasets covering sufficient sample quantities; however, in most cases, IoT traffic is distinctly not labelled. Lastly, without comprehensive standardization of benchmarking datasets, issues surrounding ensuring reproducibility in research and fair comparison within the field remain difficult to navigate, which further hinders conceiving and validating proposed solutions.

3. PROPOSED METHODOLOGY

The proposed Lightweight Ensemble Deep Learning (LEDL) Framework combines GoogLeNet, VGG16, and a Transformer to leverage spatial features, deep visual representations, and long-range dependencies, respectively. The method adaptively fuses the outputs using a dynamic confidence-based fusion process which allows the predictions to be optimally weighted based on their relevance. Additionally, we apply model pruning and quantization to help reduce model complexity while maintaining prediction accuracy. When the model will predict with low confidence, a Few-Shot Learning module will classify rare types of unexplored IOT and perform classification in minimal samples. For proper explainability, we utilize Sparsemax Attention and Attribution Loss, so our attention maps focus on and align with the target labels. As well, we use SHAP to visualize the global feature attribution and to perform local attribution for each of the patches we have envisioned. The hyperparameters of the ensemble and each module will be tuned by Adaptive FOSSA Optimization for better convergence of the generations and performance. Figure 1 shows the overall flow of our proposed methodology.

Vol. 22, No. 1, (2025)
ISSN: 1005-0930

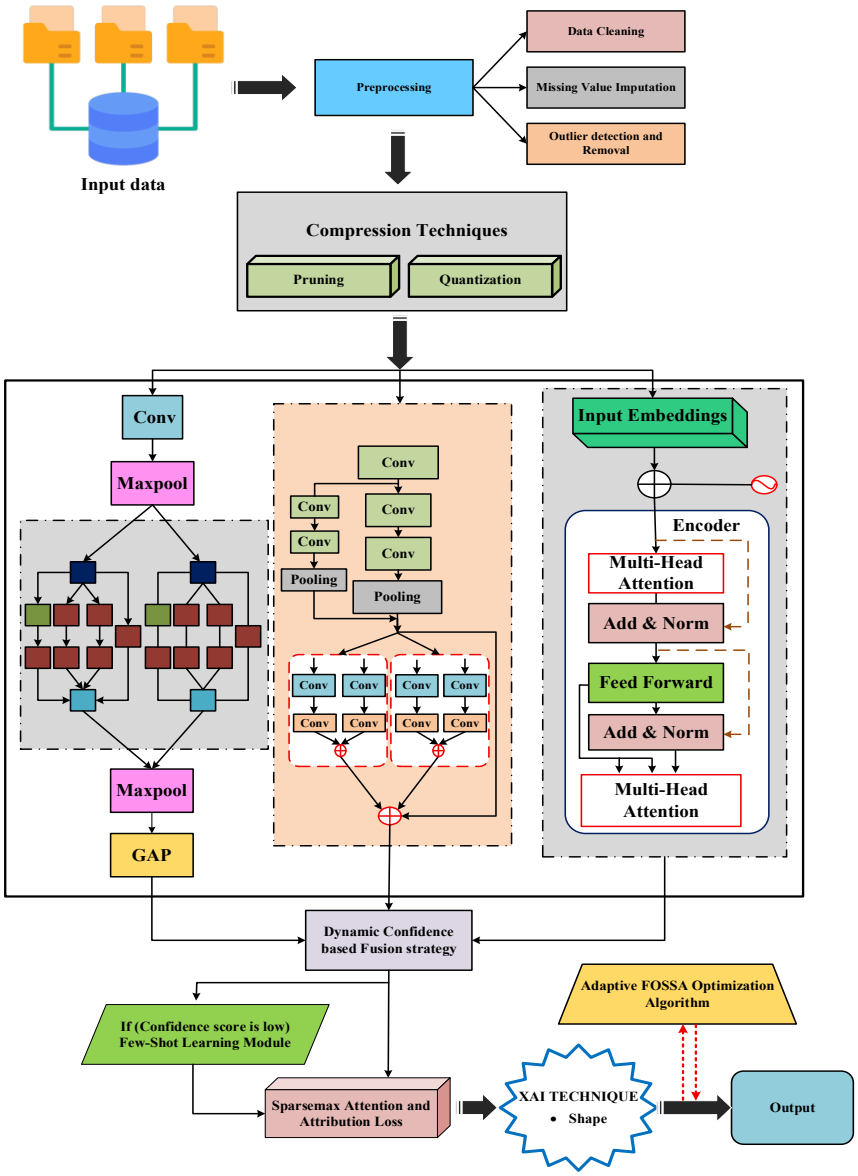


Figure 1 Overall flow of the proposed framework

3.1 Preprocessing

The first section of Preparation is designed to validate and secure the data from an IoT device before any analytical tasks or machine learning tasks are performed. The preprocessing section consists of three main functions: data cleaning, missing value processing, and outlier detection.

3.1.1 Data Cleaning

Cleaning data is the procedure of finding and fixing wrong, inconsistent, unrelated, or repeated entries in a dataset. The removal of extra logs, nonsensical sensor values, and formatting errors found in datasets containing IoT data streams is imperative as a first step for cleaning the dataset. Data cleaning improves the state of the dataset (quality and consistency) to prepare it for further analysis.

Vol. 22, No. 1, (2025)
ISSN: 1005-0930

3.1.1 Missing value imputation

Handling missing values is used to manage incomplete datasets caused by factors such as the malfunction of sensors, loss of connectivity, or disruptions in the environment during the IoT process. Rather than dropping these datasets completely, we use imputation to estimate and replace the missing values. This allows the pattern of data to be preserved, and it prevents the loss of essential information that may be relevant to other downstream processes.

3.1.2 Outlier detection and Removal

Outlier detection aims to detect and remove data points that stray significantly from the normal point, potentially leading to undesirable outcomes or models that are biased. The authors of this work used the Z-score method to calculate the extent of a data point strays away from the average. The Z-score is calculated as:

$$Z = \frac{X-\mu}{\sigma} \tag{1}$$

where X is the observed value, μ is the average, and σ represents the standard deviation. Data points with Z-scores that fall outside a predetermined threshold $(e.g., | Z | \geq 3)$ are considered outliers and removed to ensure the dataset reflects realistic operating conditions.

3.2 Feature Extraction Using Lightweight Ensemble Deep Learning Model

After preprocessing, the cleaned and normalized IoT device data is fed into the Lightweight Ensemble Deep Learning Model. To ensure suitability for resource-constrained IoT environments, pruning and quantization techniques are applied to each model in the ensemble. Pruning eliminates redundant neurons and connections, thereby reducing computational overhead, High-precision parameters are transformed into lower-precision formats using quantization. significantly lowering the amount of RAM used without compromising accuracy. Quantization process can be shown as:

$$Q(x) = round\left(\frac{x}{s}\right) \cdot s \tag{2}$$

where x is the original value, s is the scaling factor, and Q(x) is what quantized worth. The ensemble consists of three modified deep learning models GoogLeNet, VGG-16, and a Transformer each designed to capture different feature representations. The outputs from these models are fused through a Dynamic Confidence-Based Fusion Strategy, enabling adaptive decision-making based on individual model confidence scores.

Vol. 22, No. 1, (2025)
ISSN: 1005-0930

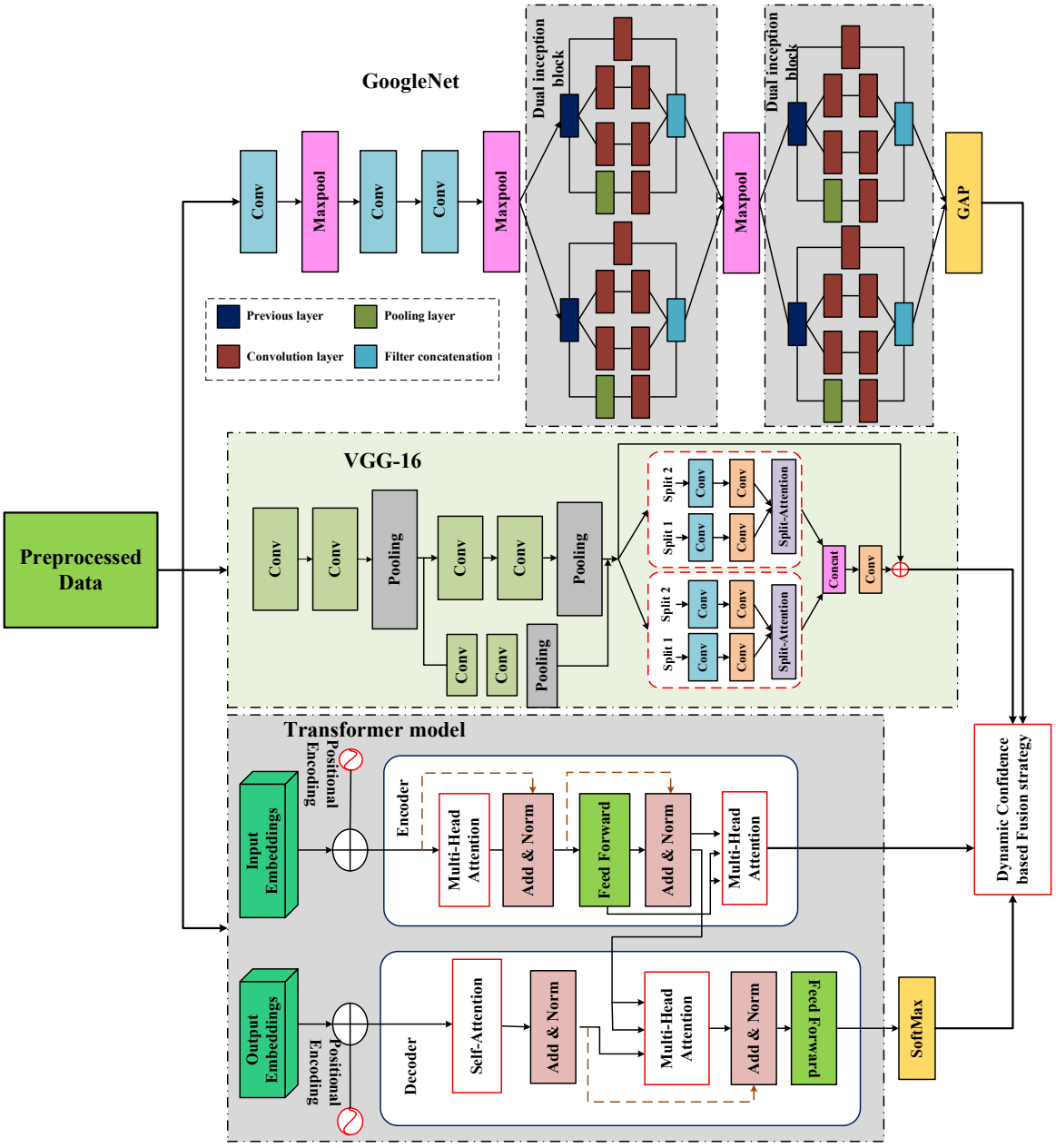


Figure 2 Architecture of Lightweight Ensemble Deep Learning Model

In the first stream, a modified GoogLeNet serves as the spatial feature extractor. The network begins with a convolutional layer (Conv) for low-level edge and texture extraction, followed by A layer of max-pooling (MaxPool) to reduce the size of spatial dimensions. This is succeeded by additional convolution and pooling layers for progressive feature abstraction. The core of the GoogLeNet architecture the Inception block is modified into a Dual Inception Block arrangement, where two Inception modules are stacked sequentially. Each Inception module includes multiple convolutional branches (e.g., 1×1, 3×3, 5×5 filters) and pooling paths, whose outputs are concatenated to form a rich multi-scale representation. Stacking two Inception blocks increases the depth and receptive field diversity, enabling more discriminative feature extraction from heterogeneous IoT data. Fully connected networks are replaced with a Global Average Pooling (GAP) layer. layers at the end, generating compact feature vectors and mitigating overfitting. Mathematically, the GAP layer for a feature map $F \in R^{H \times W}$ is defined as:

Vol. 22, No. 1, (2025)
ISSN: 1005-0930

$$GAP(F) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} F_{i,j} \quad (3)$$

where H and W are the feature map's width and height, respectively. GAP ensures that spatial information is aggregated into a compact representation for fusion.

In the second stream, the modified VGG-16 is used as the deep visual representation extractor. The architecture follows the original VGG design with stacked 3×3 convolution layers, ReLU activations, and periodic max-pooling layers for spatial reduction. After each pooling stage, a Skip Attention mechanism is incorporated to retain fine-grained details by creating direct connections between earlier and later layers, ensuring that spatial information lost in pooling is preserved. Additionally, a Split Attention mechanism is integrated, which divides feature maps into multiple splits, applies attention weighting to each, and then aggregates the outputs. The Split Attention operation is formulated as:

$$U = \sum_{k=1}^{K} \sigma(A_k) \odot X_k \quad (4)$$

where K denotes the number of splits, X_k represents the $k - th$ feature split, A_k is the corresponding attention weight, σ is the function of sigmoid, and $\odot$ denotes multiplication by elements. This approach allows the model to highlight certain discriminative feature groups, enhancing robustness in recognizing diverse IoT device patterns.

In the third stream, the modified Transformer is introduced to model long-range dependencies and global feature interactions within the IoT data. Input features are first embedded into a continuous vector space, after which positional encodings are added to preserve sequence order information. The encoder is enhanced with an additional Multi-Head Attention (MHA) module, mathematically described as:

$$\begin{gathered} MultiHead(Q,K,V) = SoftMax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \\ Q = XW_Q, K = YW_K, V = YW_V \end{gathered} \quad (5)$$

where Q, K, and V are the query, key, and value matrices, and W_Q, W_K, and W_V are learnable weight matrices for each attention head. By enabling simultaneous learning of multiple contextual relationships, MHA enhances The capacity of the model to capture intricate interdependencies in between loT device features.

In the decoder, we integrate an additional Self-Attention mechanism to improve context refinement before final classification. Self-Attention is expressed mathematically formulated as:

$$Attention(Q,K) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right) \quad (6)$$

where Q and K represent feature embeddings, $1/\sqrt{d_k}$ serves to stabilize gradients, and The softmax function normalizes the attention ratings to focus on the most relevant features. This additional self-attention layer enhances the Transformer's the capacity to distinguish between IoT devices with similar local patterns but distinct global structures.

The outputs from all three models are aggregated via the Dynamic Confidence-Based Fusion Strategy, where each model's prediction is weighted according to its confidence score for the given input. If P_i is the probability vector derived from the i-th prototype and w_i its confidence weight, the final ensemble prediction is:

Vol. 22, No. 1, (2025)
ISSN: 1005-0930

$$P_{final} = \frac{\sum_{i=1}^{M} w_i \cdot P_i}{\sum_{i=1}^{M} w_i} \quad (7)$$

where M indicates the overall quantity of models in the ensemble. This adaptive weighting ensures that the model with the highest confidence in a given instance has the greatest influence on the final classification, improving both accuracy and reliability in IoT device type identification.

3.3 Few-Shot Learning Model

To effectively classify rare or unseen IoT device types when the available data is extremely limited, our framework integrates a Few-Shot Learning (FSL) module that is triggered whenever the dynamic confidence-based fusion stage produces a prediction confidence below a predefined threshold τ_c. The FSL module adopts a Prototypical Network-based metric learning approach, where each device type is represented by a prototype vector computed as the mean embedding of its few available support samples. Given a query embedding q, classification is performed by computing the distance between q and each class prototype in the embedding space, assigning the query to the class with the smallest distance. The prototype for class k is computed as:

$$p_k = \frac{1}{|S_k|} \sum_{(x_i, y_j) \in S_k} f_\theta(x_i) \quad (8)$$

where S_k is the support set for class $k_r f_\theta(\cdot)$ is the feature extractor, and $|S_k|$ is the number of support samples. The Euclidean distance between the query q and a class prototype P_k is:

$$d(q, p_k) = \|f_\theta(q) - p_k\|_2 \quad (9)$$

The probability of the query belonging to class k is then given by a softmax over the negative distances:

$$P(y = k \mid q) = \frac{exp\left(-d(q, p_k)\right)}{\sum_{k'} exp\left(-d(q, p_{k'})\right)} \quad (10)$$

This FSL integration ensures adaptability in recognizing new or infrequently encountered IoT device types without requiring costly retraining, thereby enhancing the robustness and scalability of the proposed framework in heterogeneous IoT environments.

3.4 Sparsemax Attention and Attribution Loss

To enhance interpretability within the proposed framework, we integrate Sparsemax Attention to produce more focused and label-aligned attention distributions. Unlike Softmax, which assigns non-zero probabilities to all elements, Sparsemax yields sparse probability distributions, enabling The model to concentrate on a smaller, more pertinent set of features. Mathematically, the Sparsemax transformation for a given input vector $z \in \mathbb{R}^K$ is defined as:

$$Sparsemax(z) = arg \min_{p \in \Delta^{x-1}} \|p - z\|^2 \quad (11)$$

where $\Delta^{K-1} = \{p \in \mathbb{R}^K \mid \sum_{i=1}^{K} p_i = 1, p_i \geq 0\}$ represents the $(K-1)$ dimensional probability simplex, z is the pre-activation score vector, and p is the resulting sparse probability vector. The closed-form computation is expressed as:

$$Sparsemax(z)_i = max(z_i - \tau(z),0) \quad (12)$$

where z_i is the i-th element of the input vector z, and $\tau(z)$ is the threshold ensuring $\sum_{i=1}^{K} \max(z_i - \tau(z),0) = 1$, computed from the sorted values of z.

To further align attention maps with the correct class labels, we employ Attribution Loss, which penalizes attention misalignment between the model's anticipated importance and the reality on the ground feature relevance. The Attribution Loss is formulated as:

$$\mathcal{L}_{attr} = \frac{1}{N}\sum_{n=1}^{N} \|A_n - G_n\|_2^2 \quad (13)$$

where N is the batch size, A_n denotes the attention attribution map generated by Sparsemax for the n-th input, and G_n is the corresponding ground-truth attribution mask. The $\|\cdot\|_2^2$ term measures the squared Euclidean distance, ensuring that the learned attention focuses on features consistent with expert-defined or label-derived regions.

By integrating Sparsemax Attention with Attribution Loss, the proposed framework not just improves accuracy of predictions, but also ensures clear and decision-making that is interpretable, enabling reliable deployment in resource-constrained IoT environments. Building upon the interpretability enhancements introduced through Sparsemax Attention and Attribution Loss, the next stage of the proposed framework focuses on optimizing the ensemble's learning efficiency and predictive performance using the Adaptive FOSSA Optimization (AFOA) algorithm.

3.5 Adaptive FOSSA Optimization Algorithm

The fossa (Cryptoprocta ferox) hunts using a two-step approach (i) an initial attack in the direction of a detected lemur, and (ii) a pursuit through trees until it captures the lemur. We follow a similar model of their behaviors to develop a population-based optimizer to tune the hyperparameters of our ensemble for a resource-constrained IoT environment.

Algorithm initialization

A novel population-based optimization algorithm called AFOA, in which each fossa represents a potential remedy in the area. The fossa's habitat represents the optimization domain to the problem, while where each fossa is vector represents A possible remedy to the optimization Issue.

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_i \\ \vdots \\ X_N \end{bmatrix}_{N\times m} = \begin{bmatrix} x_{1,1} & \cdots & x_{1,d} & \cdots & x_{1,m} \\ \vdots & \ddots & \vdots & \iddots & \vdots \\ x_{i,1} & \cdots & x_{i,d} & \cdots & x_{i,m} \\ \vdots & \iddots & \vdots & \ddots & \vdots \\ x_{N,1} & \cdots & x_{N,d} & \cdots & x_{N,m} \end{bmatrix}_{N\times m} \quad (14)$$

$$x_{i,d} = lb_d + r \cdot (ub_d - lb_d) \quad (15)$$

In this situation, X symbolizes the FOA population matrix, where X_i represents a potential solution, the ith fossa. $x_{i,d}$ symbolizes the dth size of the ith The search space's fossa, where N represents the quantity of fossas, m shows how many decision factors there are., r is an arbitrary number within the range [0,1], lb_d, and ub_d indicate the bottom and top limits of the dth choice factor, in turn.

Vol. 22, No. 1, (2025)
ISSN: 1005-0930

As mentioned earlier, the position of each fossa indicates a possible solution to the issue and is subject to examination within the objective function. According to Eq., the values obtained from evaluating the objective function can be contained in a vector. (16).

$$F = \begin{bmatrix} F_1 \\ \vdots \\ F_i \\ \vdots \\ F_N \end{bmatrix}_{N\times 1} = \begin{bmatrix} F(X_1) \\ \vdots \\ F(X_i) \\ \vdots \\ F(X_N) \end{bmatrix}_{N\times 1} \tag{16}$$

Here, F symbolizes the vector of values from the evaluated objective function, with F_i displaying the value of the objective function that corresponds to the *i*th fossa.

Phase 1: Exploration-Attacking the lemur

The first step in FOA is to adhere the current population members' places by mimicking the fossa's ambush on an identified lemur. The fossa uses its keen senses to find the lemur and travel toward it, resulting in large changes in position that facilitate global exploration. For every fossa, candidate lemur positions are considered to be any position of other participants whose objective function value is higher:

$$CL_i = \{X_k : F_k < F_i \; and \; k \neq i\}, \tag{17}$$

$$\text{Where } i = 1,2,\dots,N \; and \; k \in \{1,2,\dots,N\} \tag{18}$$

In this equation, CL_i represents the group of potential candidate's lemur places for the ith fossa, Xk denotes the individual in the population who has a superior value of the objective function as opposed to the ith fossa, and Fk is its corresponding value of the objective function. A fossa chooses one candidate at random. lemur and updates its position using:

$$x_{i,j}^{P1} = x_{i,j} + r_{i,j} \cdot \left(SL_{i,j} - I_{i,j} \cdot x_{i,j}\right), \tag{19}$$

If this updated position yields a superior objective function value, it replaces the previous position of the respective population member, as outlined in:

$$X_i = \begin{cases} X_i^{P1}, F_i^{P1} \leq F_i \\ X_i, \; else \end{cases} \tag{20}$$

X_i^{P1} symbolizes the recently determined location for the ith fossa during the FOA's attack phase, with $x_{i,j}^{P1}$ as its jth dimension. In this new role, the objective function value is F_i^{P1}. The terms $r_{i,j}$ are arbitrary n numbers that fall inside the range [0, 1], and $I_{i,j}$ are arbitrary numbers, either 1 or 2.

Phase 2: Exploitation-Chasing the lemur

In the second stage, the Adaptive FOA boosts local exploration by imitating the fossa pursuing the lemur, which is a sophisticated search within a finite area, where the updates of positions are adaptive with a learning rate, to make these positional updates more accurate. The adaptive learning rate α_t is defined as:

$$\alpha_t = \alpha_{max} - \left(\frac{t}{T_{max}}\right) \cdot (\alpha_{max} - \alpha_{min}) \tag{21}$$

where α_{max} and α_{min} are the maximum and minimum learning rates, t is the version in use right now, and T_{max} is the most iterations that can occur. The most recent position for the $i - th$ Next, fossa is calculated as follows:

$$x_{i,j}^{P2} = x_{i,j} + \alpha_t \cdot (1 - 2r_{i,j}) \cdot \frac{ub_j - lb_j}{t} \quad (22)$$

where $r_{i,j} \in [0,1]$, ub_j and lb_j are the boundaries at the top and bottom of the $j-th$ size. If the new position improves the objective value F_i^{P2}, it replaces the old position:

$$X_i = \begin{cases} X_i^{P2}, F_i^{P2} \leq F_i \\ X_i, \ else \end{cases} \quad (23)$$

This adaptive mechanism balances exploration and exploitation, improving convergence speed and final solution quality. In our proposed framework, this adaptive exploitation phase fine-tunes the ensemble's parameters more precisely, enhancing predictive accuracy and ensuring faster convergence.

3.6 Explainable AI Techniques: SHAP

In our research, we use SHAP (Shapley Additive Explanations) as a surrogate explanatory method based on game theory to offer a constitutive and globally interpretable explanation of the model predictions. Specifically, SHAP measures the contribution of each feature by measuring its marginal contribution to the prediction. We can represent a model prediction as:

$$a(z) = \phi_0 + \sum_{l=1}^{L} \phi_l \, z_l \quad (24)$$

where ϕ_l is the SHAP value indicating the contribution of the lth feature and z_l is an indicator for the presence of that feature. SHAP values are computed using:

$$\phi_l = \sum_{s \subseteq N\{l\}} \frac{|S|!(L-|S|-1)!}{L!} [f_x(S \cup \{L\}) - f_x(S)] \quad (25)$$

This approach guarantees that the weight placed on each attribute is fair, consistent, and based on solid theoretical grounding. By embedding SHAP within our approach, we gain a comprehensive view of the factors contributing to model predictions. SHAP can show which patterns, such as anomalies related to high network traffic, unusual system call dispositions, or odd patterns of protocol use, are driving a possible cyber-attack detection. This transparency further strengthens trust regarding the system and its outputs, enables decision-making based on rich evidence, and supports accountability for AI components that create security mechanisms.

3.7 Classification

Classification is the final decision-making phase that links the optimized feature representations to their corresponding IoT device types. The optimized features, which are received after the adaptive FOSSA optimization, are put through a fully connected layer to create a fixed-length representation for our learned feature space. Then a Softmax activation layer converts the dense layer outputs, or logits, into a probability distribution over all IoT device types. The Softmax function is defined as:

$$P(y_i) = \frac{e^{z_i}}{\sum_{j=1}^{C} e^{z_j}} \quad (26)$$

where z_i is the logit for device type i, C is the total quantity of device categories as well as $P(y_i)$ represents The likelihood of the device being a part of class i. The predicted class is assigned to the device type with the maximum likelihood.

Vol. 22, No. 1, (2025)
ISSN: 1005-0930

This approach guarantees high classification accuracy for the framework with less computational complexity and is, therefore, deployable in IoT environments with resource constraints. The Softmax layer's probabilistic nature also helps to assess model confidence, which is beneficial to the SHAP-based explainability stage.

4. RESULTS AND DISCUSSION

The framework suggested in this study was benchmarked against a number of the current situation-modern architectures along with detailed evaluation metrics such as Accuracy, F1-Score, MCC, NPV, FPR, FNR, Sensitivity, Specificity, and Precision showing better overall performance in all datasets as the proposed model had the higher accuracy to lower error rates. The performance analysis with additional cutting-edge optimization techniques supported the effectiveness of the Adaptive FOSSA Optimization technique based on the faster convergence of solutions, improved generalization, and stability of the model. An ablation study showed all modules Adaptive FOSSA, GoogLeNet, VGG16, and Transformer were useful because their removal from the study impacted performance. This showed that Adaptive FOSSA, GoogLeNet, VGG16, and Transformer created a synergy that positively impacted robustness, feature diversity, and predictive accuracy. The proposed architecture took a bit longer to compute, but the performance trade-off was worth the additional computation.

4.1 Dataset Description

The experimental evaluation in this research is performed using three benchmark datasets that together provide different Network and IoT intrusion use case scenarios. Dataset 1 [26] (Network Intrusion Dataset, Kaggle) is a labelled dataset comprising network traffic records, with characteristics like duration, protocol type, service and connection flags, alongside a range of flow statistics, that are used to support a binary classification of respectful versus intrusive activity, and are widely used in both intrusion detection studies and F1-score optimisation (d) tasks. Dataset 2 [27] (TON-IoT dataset, Kaggle) donated from UNSW Canberra containing heterogeneous telemetry, and OS audit logs (Windows/Linux) collected from IoT/IIoT environments, network traffic data, intrusion detection, forensic analysis and threat intelligence analysis assessing these primarily within an edge–fog–cloud context. Dataset 3 [28] The Australian Centre for Cyber Security (ACCS) used IXIA to create the (UNSW-NB15 dataset). PerfectStorm, as a hybrid of modern normal activities and created with contemporary synthetic attacks, and contains 100 GB of raw pcap traffic that has been post-processed with Argus and Bro-IDS, using 49 statistical features with 9 attack types (called Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms). As publicly available benchmark datasets, these three datasets also have their own separate datasets as training and testing datasets, and can enable repeatable IDS comparative evaluations using these datasets as different IDS's should provide different performance metrics.

4.2 Performance Evaluation

The proposed model was evaluated against existing ensemble methods such as Stacking-CNN, Hybrid VGG16–ResNet, DenseNet–Inception Fusion, and CNN–Transformer using metrics like Accuracy, Precision, Sensitivity, Specificity, F1-Score, MCC, NPV, FPR, and FNR. It consistently outperformed these approaches, achieving higher accuracy, balanced sensitivity specificity, and lower error rates. Comparative analysis with optimization algorithms including PSO, GA, GWO, and WOA demonstrated the superiority of Adaptive FOSSA in feature selection and convergence speed. Ablation results further validated the

necessity of each component Adaptive FOSSA, GoogLeNet, VGG16, and Transformer—for optimal performance.

Table 2 Performance Metrics Comparison of Existing Methods in Dataset 1

Metric	CNN-BiLSTM	CNN-ViT	AE-LSTM-CNN	TRBMA	Proposed
Accuracy	0.9115	0.9487	0.9298	0.9432	0.9956
Precision	0.8594	0.9564	0.9492	0.9478	0.9901
Sensitivity	0.9361	0.9405	0.9113	0.9186	0.9972
Specificity	0.9181	0.9482	0.9348	0.9401	0.9919
F1-Score	0.86142	0.94621	0.94402	0.94279	0.98853
MCC	0.87234	0.95283	0.93527	0.95167	0.99368
NPV	0.85277	0.94556	0.93712	0.94432	0.98844
FPR	0.0933	0.0842	0.0782	0.1349	0.0097
FNR	0.1247	0.1093	0.0771	0.0897	0.0086
Computation Time	150	172	96	105	59

Table 2 shows A comparison performance examination of the suggested framework with four baseline models, encompassing CNN-BiLSTM, CNN-ViT, AE-LSTM-CNN, and TRBMA metrics. The provided results show that the proposed framework outperformed all baselines according to all the evaluation criteria and achieved the largest scores for accuracy (0.9956), precision (0.9901), sensitivity (0.9972), specificity (0.9919), F1-score (0.9885), MCC (0.9937), and NPV (0.9884), along with the lowest error rates, which were percentage of false positives (0.0097) and the rate of false negatives (0.0086) that were significantly lower than any of the provided baselines leading to a greater level of trust in the proposed frameworks ability to detect intrusions and low levels of false alarms. The proposed framework achieved all these competitive outputs in the shortest computation time (59 s), so we would argue that the proposed framework also performed better in terms of efficiency and suitability for IoT device type identification in a real-time scenario, particularly in resource-constrained situations.

Table 3 Performance Metrics Comparison of Existing Methods in Dataset 2

Metric	CNN-BiLSTM	CNN-ViT	AE-LSTM-CNN	TRBMA	Proposed
Accuracy	0.9023	0.9452	0.9226	0.9358	0.9891
Precision	0.8452	0.9511	0.9459	0.9413	0.9878
Sensitivity	0.9289	0.9382	0.9034	0.9112	0.9926

Specificity	0.9074	0.9443	0.9293	0.9371	0.9896
F1-Score	0.84982	0.94125	0.93918	0.93692	0.98471
MCC	0.86791	0.94933	0.92971	0.94728	0.99055
NPV	0.84316	0.94028	0.93002	0.93914	0.98567
FPR	0.1055	0.0976	0.0903	0.1564	0.0119
FNR	0.1364	0.1223	0.0916	0.1028	0.0111
Computation Time	160	183	102	118	65

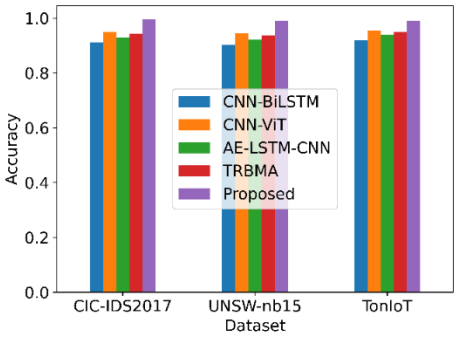
Table 3 illustrates the effectiveness of the suggested framework and four baselines (CNN-BiLSTM, CNN-ViT, AE-LSTM-CNN, and TRBMA) using multiple evaluation methods. The proposed method outperformed all the baseline methods consistently and attained the maximum level of accuracy (0.9891), accuracy (0.9878), sensitivity (0.9926), specificity (0.9896), F1-score (0.9847), MCC (0.9906) and NPV (0.9857). It also achieved the lowest false positive rate (1.19) and false negative rate (1.11), representing a robust level of consistency for correctly identifying true intrusions and providing low false alarms. Moreover, it has also done all this with a computed time of only 65 seconds, which is very fast than all of the models that are compared. Therefore, the proposed framework is appropriate in cases of needing real-time IoT device type identification in resource constraint environments.

Table 4 Performance Metrics Comparison of Existing Methods in Dataset 3

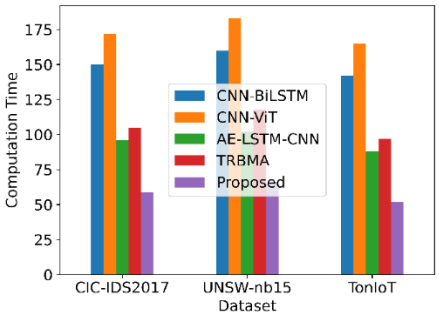
Matric	**CNN-BiLSTM**	**CNN-ViT**	**AE-LSTM-CNN**	**TRBMA**	**Proposed**
Accuracy	0.9184	0.9535	0.9382	0.9491	0.9899
Precision	0.8748	0.9603	0.9526	0.9552	0.9903
Sensitivity	0.9395	0.9447	0.9187	0.9278	0.9914
Specificity	0.9263	0.9524	0.9404	0.9473	0.9929
F1-Score	0.87234	0.95083	0.94631	0.94892	0.99116
MCC	0.88021	0.9581	0.94355	0.95716	0.99281
NPV	0.86115	0.94968	0.94468	0.95072	0.9917
FPR	0.0879	0.0765	0.0661	0.1247	0.0071
FNR	0.1138	0.0991	0.0649	0.0772	0.0062
Computation Time	142	165	88	97	52

Table 4 indicates the proposed framework's performance compared to CNN-BiLSTM, CNN-ViT, AE-LSTM-CNN, and TWBMA, using multiple evaluation metrics. The proposed

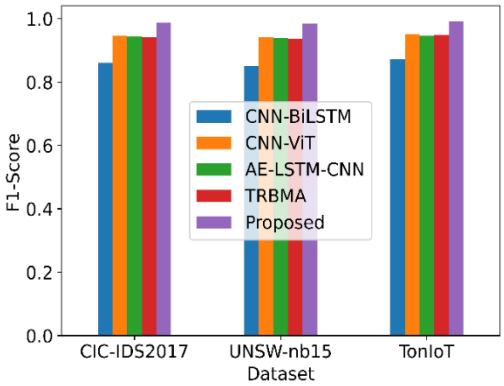
model achieved the most significant performance metrics across all relevant metrics, achieving 0.9899 accuracy, 0.9903 precision, 0.9914 sensitivity, 0.9929 specificity, 0.9912 F1 score, 0.9928 MCC, and 0.9917 NPV. The proposed framework also achieved a significant reduction in error by achieving the lowest rate of false-positive results (0.0071) as well as the false-negative rate (0.0062), indicating that it not only detects true intrusions better but that it also does so with fewer missed detections and fewer false alarms than other models. Finally, the proposed framework achieved the quickest computation time (52 seconds), and all baseline models demonstrated slower performance, demonstrating that this framework is extremely efficient for real-time identification of IoT device types in deployment environments that tend to be resource-constrained.



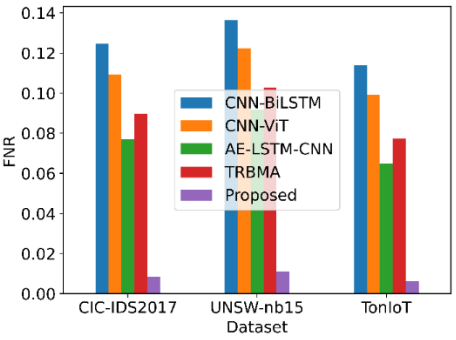
(a)



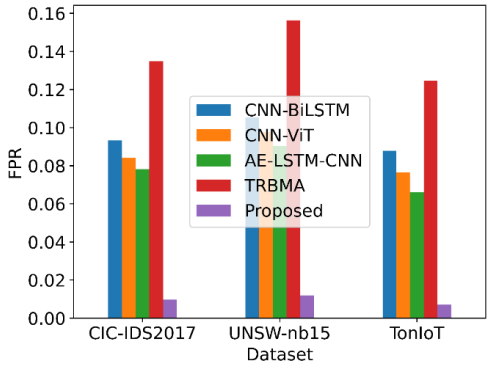
(b)

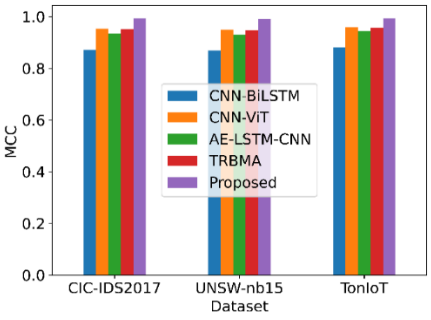


(c)



(d)





Vol. 22, No. 1, (2025)
ISSN: 1005-0930

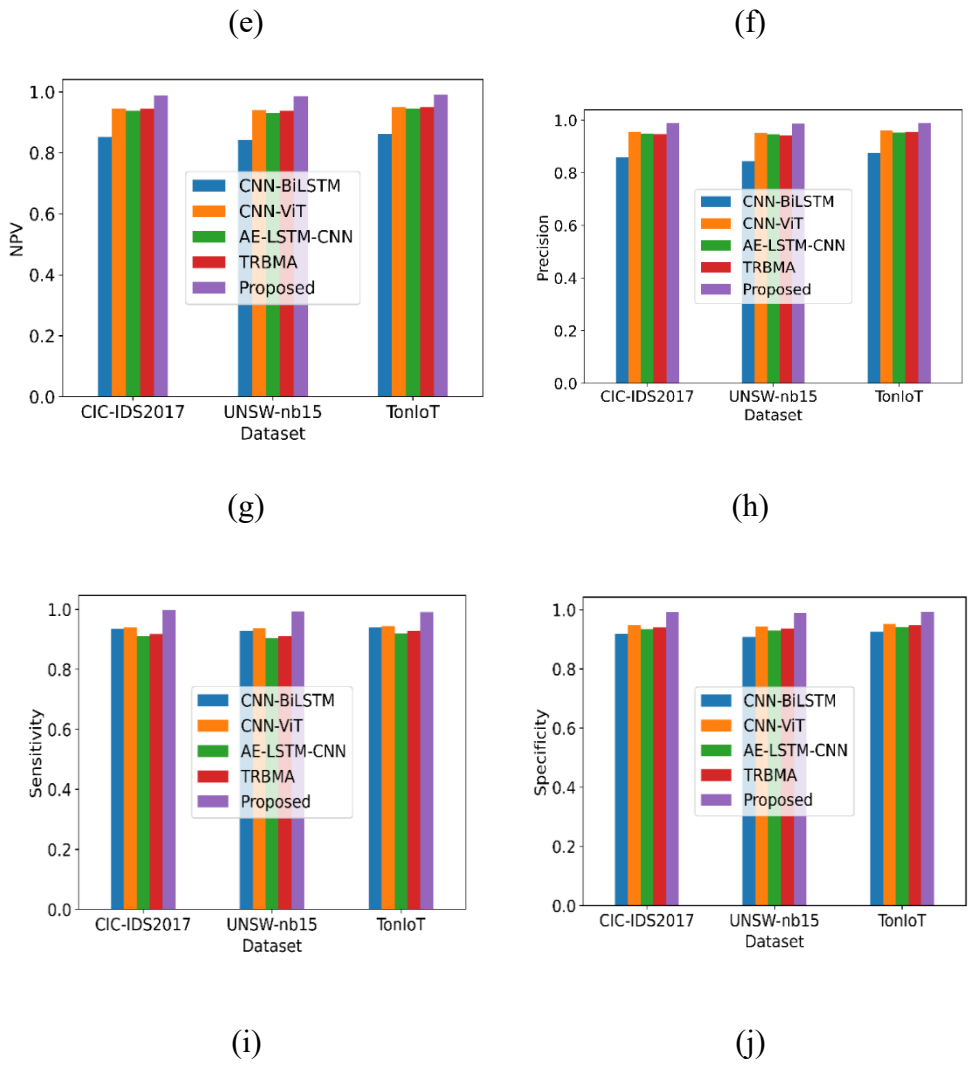


Figure 3 (a)–(j) Visual Representation of Various Performance Metrics

The comparative Accuracy, Computation time, Precision, Sensitivity, MCC, NPV, F1-score, FNR, and FPR and Particulars performance of different existing methods CNN-BiLSTM, CNN-ViT, AE-LSTM-CNN, TRBMA, and Proposed, across three benchmark datasets: CIC-IDS2017, UNSW-NB15, and TonIoT is represented in Figure 3 (a)-(j).

Table 5 Performance Metrics Comparison of Optimization Algorithms in Dataset 1

Metric	PSO	ABCO	POA	FOSSA	Proposed
Accuracy	0.8843	0.9182	0.9047	0.9288	0.9956
Precision	0.8612	0.9051	0.9198	0.9345	0.9901
Sensitivity	0.8963	0.9124	0.8981	0.8811	0.9972
Specificity	0.8735	0.8922	0.9074	0.9157	0.9919
F1-Score	0.84789	0.90165	0.92983	0.92513	0.98853

MCC	0.85972	0.91784	0.9211	0.91856	0.99368
NPV	0.85701	0.91302	0.91847	0.92432	0.98844
FPR	0.2021	0.1345	0.0999	0.0984	0.0097
FNR	0.1843	0.1594	0.1017	0.1722	0.0086
Computation Time	305	198	131	244	59

Table 5 provides comparisons of the proposed framework to four optimization-based approaches (PSO, ABCO, POA, and FOSSA) by various metrics of performance. The proposed method has significantly better performance than all its baselines, reaching maximum precision (0.9956), accuracy (0.9901), sensitivity (0.9972), particularity (0.9919), F1-score (0.9885), MCC (0.9937), and NPV (0.9884). The method has the lowest percentage of false positives (0.0097) and the rate of false negatives (0.0086), which demonstrates that the suggested framework is very reliable in identifying true threats and in decreasing the probability of misclassifications. The proposed framework is also the fastest of all compared methods (just 59 seconds to complete). The next fastest method is PSO at 305 seconds, and FOSSA at 244 seconds. The proposed method was highly efficient and well-suited for real-time IoT device type identification even in low-resource contexts.

Table 6 Performance Metrics Comparison of Optimization Algorithms in Dataset 2

Metric	**PSO**	**ABCO**	**POA**	**FOSSA**	**Proposed**
Accuracy	0.9097	0.9321	0.9214	0.9382	0.9891
Precision	0.8758	0.9256	0.9387	0.9489	0.9878
Sensitivity	0.9342	0.9453	0.9163	0.8907	0.9926
Specificity	0.8916	0.9158	0.9242	0.9283	0.9896
F1-Score	0.85462	0.93427	0.94233	0.94071	0.98471
MCC	0.86291	0.94938	0.93711	0.93688	0.99055
NPV	0.86025	0.9441	0.93465	0.94012	0.98567
FPR	0.1887	0.1174	0.0889	0.0842	0.0119
FNR	0.1692	0.1453	0.0845	0.1556	0.0111
Computation Time	288	185	114	231	65

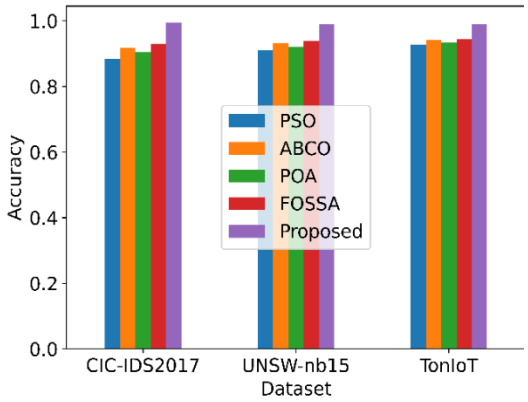
Table 6 shows a comparison of the proposed approach in the current study versus four optimization algorithms (PSO, ABCO, POA, and FOSSA) on performance. The proposed approach shows a clear advantage in performance. The proposed approach gave the best accuracy (0.9891), precision (0.9878), sensitivity (0.9926), specificity (0.9896), F1-score (0.9847), MCC (0.9905), and NPV (0.9857) while giving the lowest percentage of false positives (0.0119) and the rate of false negatives (0.0111). This shows it is robust in terms of its ability to detect true instances, while minimizing the number of errors. The proposed approach also has a computation time of 65 seconds, compared to the computation times of

PSO (288s) and FOSSA (231s) and overwhelmingly shows to have a promising degree of efficiency for speed, certainly making it applicable for real-time applications involving classifying different IoT devices.

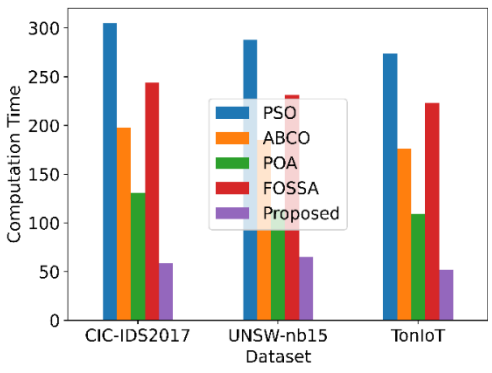
Table 7 Performance Metrics Comparison of Optimization Algorithms in Dataset 3

Metric	PSO	ABCO	POA	FOSSA	Proposed
Accuracy	0.9274	0.9412	0.9336	0.9445	0.9899
Precision	0.9032	0.9363	0.9472	0.9571	0.9903
Sensitivity	0.9408	0.9527	0.9269	0.9014	0.9914
Specificity	0.9003	0.9276	0.9351	0.9345	0.9929
F1-Score	0.86741	0.94183	0.94985	0.94866	0.99116
MCC	0.87318	0.95324	0.94323	0.9457	0.99281
NPV	0.87156	0.94957	0.94178	0.94803	0.9917
FPR	0.1745	0.1062	0.0776	0.0732	0.0071
FNR	0.1536	0.1342	0.0728	0.1411	0.0062
Computation Time	274	176	109	223	52

Table 7 illustrates the comparison between the suggested method and performance of four algorithms including PSO, ABCO, POA and FOSSA with marked improvement in the performance on each measure. The suggested algorithm has also provided maximum precision (0.9899), accuracy (0.9903), sensitivity (0.9914), particularity (0.9929), F1-score (0.9912), MCC (0.9928), and NPV (0.9917), alongside the lowest number of False positive (FPR=0.71) and rates of false negatives (FNR=0.62) where the suggested approach showed good performance with both positive and negative classifications. Similarly, it also had the shortest computational time of merely 52 seconds, compared to the second-best PSO algorithm (274 s) and FOSSA algorithm (223 s), meaning it is highly accurate and very computationally efficient when compared against the alternative algorithms with regard to time critical execution related to real-time IoT intrusion detection methods.

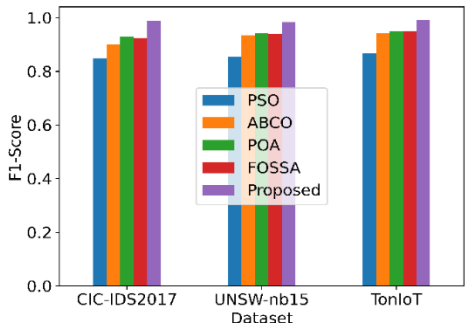


(a)

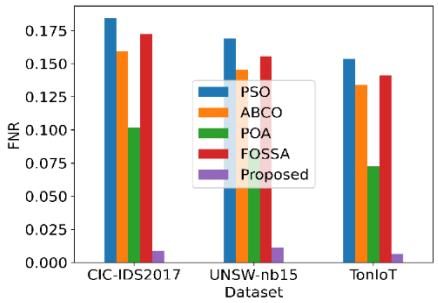


(b)

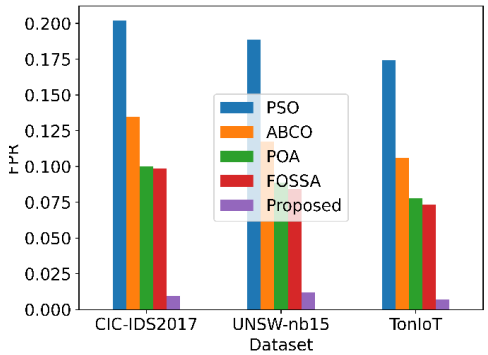
Vol. 22, No. 1, (2025)
ISSN: 1005-0930



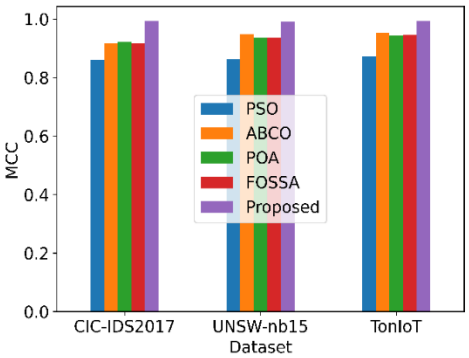
(c)



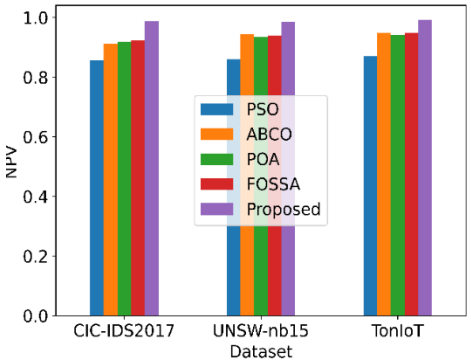
(d)



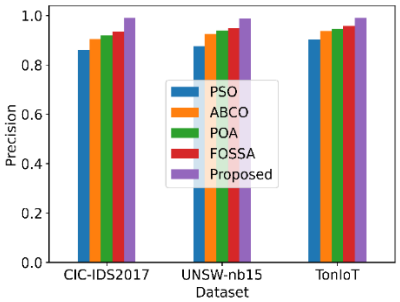
(e)



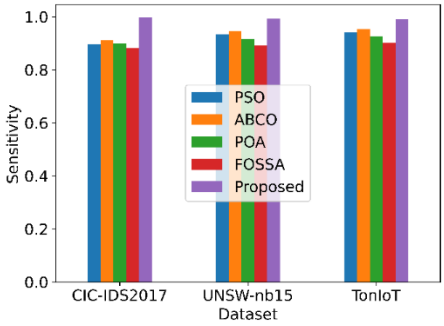
(f)



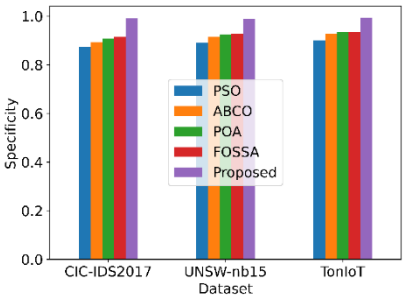
(g)



(h)



(i)

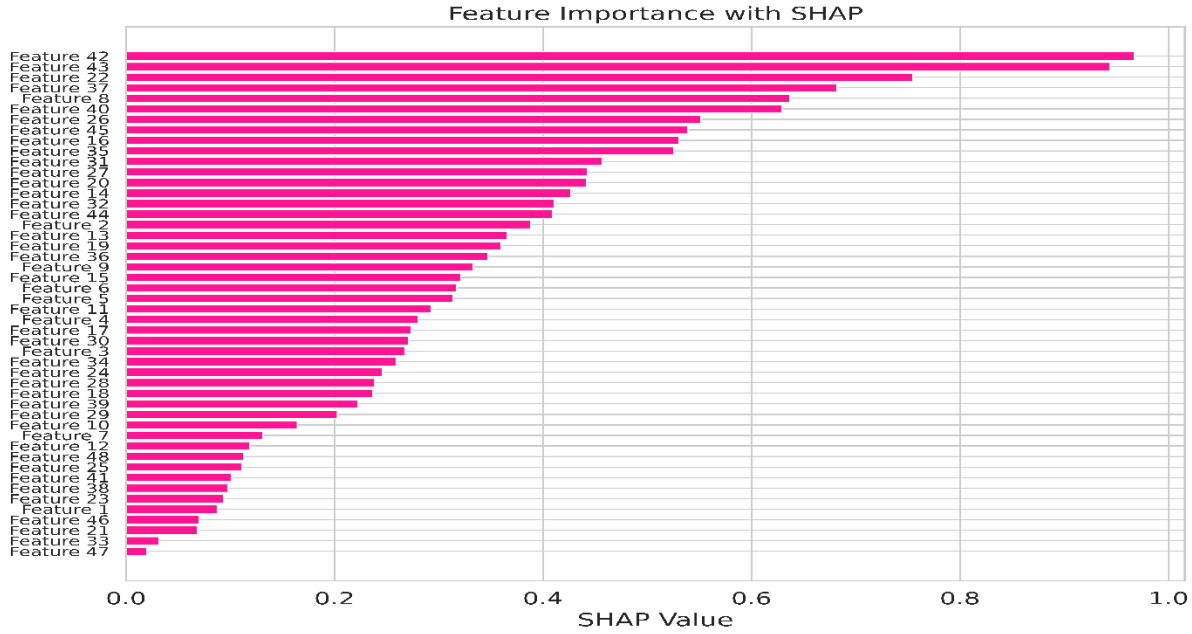


(j)

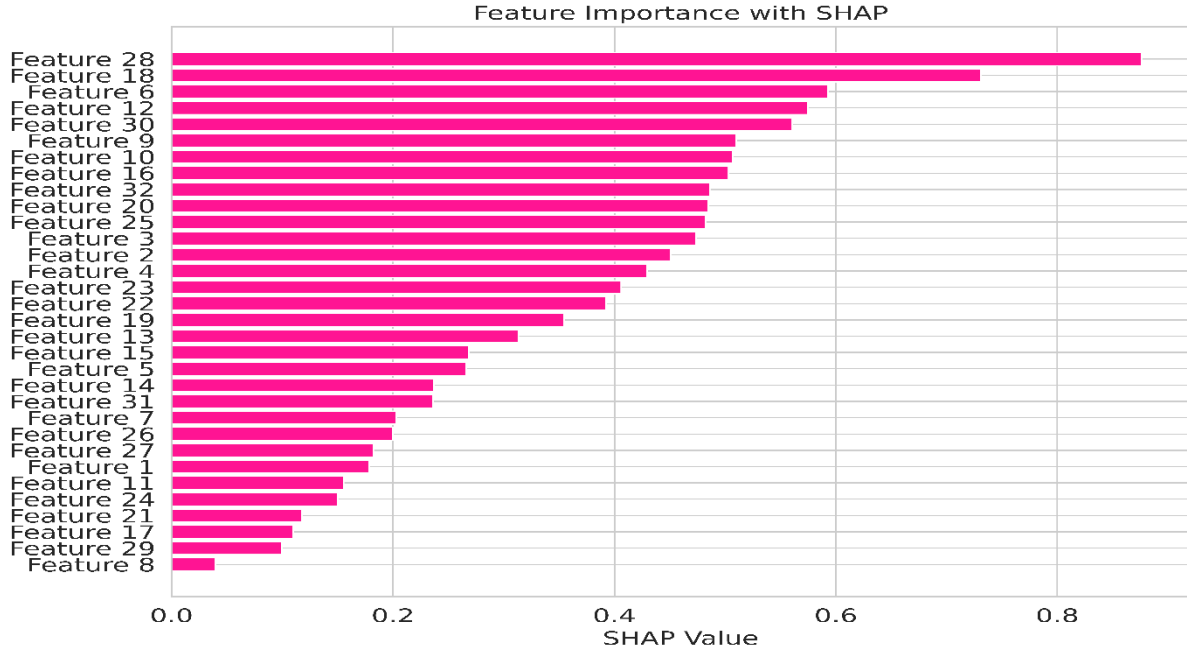
Vol. 22, No. 1, (2025)
ISSN: 1005-0930

Figure 8 (a)–(j) Visual Representation of Various Performance Metrics

The comparative Accuracy, Computation time, Precision, Sensitivity, MCC, NPV, F1-score, FNR, and FPR and Particulars performance of different optimization algorithms PSO, ABCO, POA, FOSSA, and the Proposed method across three benchmark datasets: CIC-IDS2017, UNSW-NB15, and TonIoT is represented in Figure 8 (a)-(j).

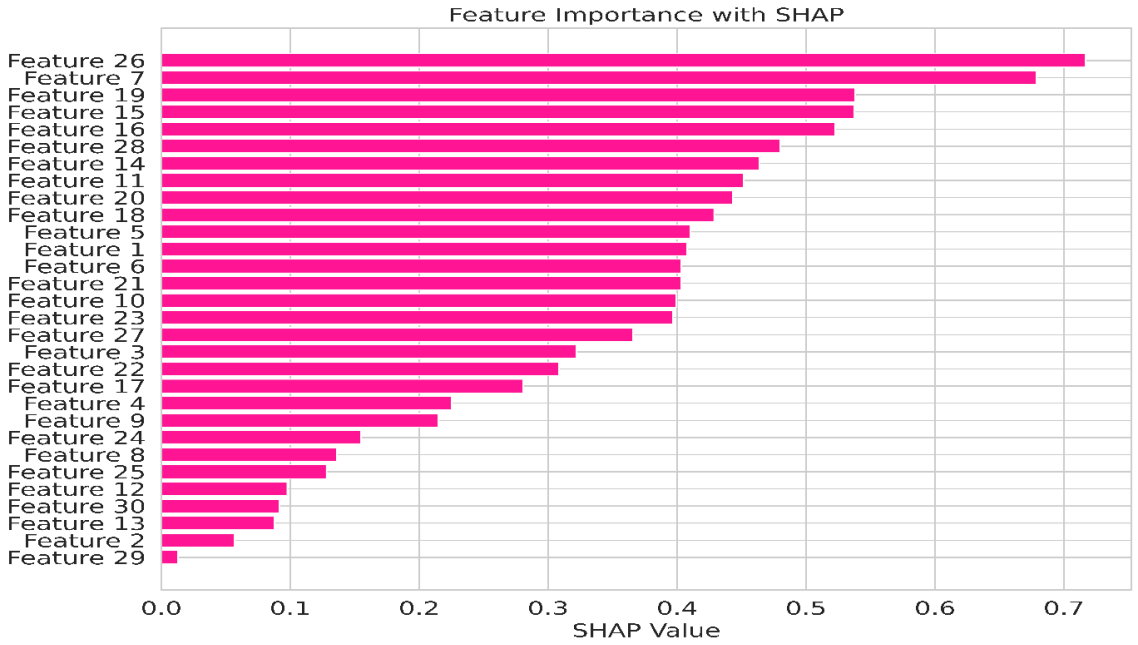


(a)



(b)

Vol. 22, No. 1, (2025)
ISSN: 1005-0930



(c)

Figure 6 (a)–(c) SHAP-Based Feature Importance Analysis for Datasets 1, 2, and 3

SHAP-based feature importance assessment (Figure 9 (a)–(c)) revealed that within the three datasets, there was also a small number of features that drove the prediction in all three datasets. In Dataset 1, the top-ranked features had large SHAP values indicating that those features were important causes of the predictions and the lowest ranked features contributed very little to the model performance. In Dataset 2, a similar outcome was achieved, as there were few high impact variables that drove a disproportionate amount of the output; which indicates that dimensionality reduction could help with efficiency without losing predictive performance. In Dataset 3, Feature 26 had the highest SHAP value, followed closely by Features 7 and 19, and the other variables dropped off sharply in importance as the SHAP values continued to decrease. The same trend across datasets suggests feasibility in feature selection that would improve interpretability and reduce computational costs without losing predictive capability.

4.4 Ablation Study

The ablation study assessed the impact of Adaptive FOSSA optimization, GoogLeNet, VGG16, and Transformer modules by removing them individually across three datasets. Results show the proposed model consistently outperformed all ablated versions in accuracy, precision, sensitivity, and robustness, with the lowest false rates. While computation time slightly increased, the performance gains highlight the complementary role of all components in enhancing overall effectiveness.

Vol. 22, No. 1, (2025)
ISSN: 1005-0930

Table 8 Evaluation of ablation study for Dataset 1

Metric	Without Adaptive FOSSA Optimization	Without GoogLeNet	Without VGG16	Without Transformer	Proposed
Accuracy	0.9843	0.9871	0.986	0.9824	0.9956
Precision	0.9725	0.978	0.9762	0.9702	0.9901
Sensitivity	0.9812	0.9854	0.9839	0.9791	0.9972
Specificity	0.975	0.9807	0.9791	0.9723	0.9919
F1-Score	0.9684	0.9732	0.9715	0.9668	0.9885
MCC	0.9769	0.981	0.9793	0.9736	0.9937
NPV	0.9652	0.9703	0.9687	0.9634	0.9884
FPR	0.025	0.0193	0.0209	0.0277	0.0097
FNR	0.0188	0.0146	0.0161	0.0209	0.0086
Computation Time (s)	42.5	48.3	51	46.7	59

The ablation study presented in Dataset 1 investigates and highlights the influence of each primary element of the proposed model, namely, Adaptive FOSSA optimization, GoogLeNet, VGG16, and the Transformer, while excluding each from the model; the results show that removing any of the model components results in declines in performance across all key parameters such as accuracy, precision, sensitivity, specificity, F1-score, MCC, and NPV, but also FPR and FNR. Furthermore, the model without Adaptive FOSSA optimization recorded the lowest accuracy value (0.9843) among model variations, while the models without GoogLeNet, VGG16, or Transformer, similarly degrade performance in different ways but overall, across measures also held declines in FPR, FNR, etc. In contrast, the suggested model produced the best results. values for accuracy (0.9956), precision (0.9901), sensitivity (0.9972), and MCC (0.9937), but remained the lowest FPR (0.0097), and FNR (0.0086), which indicates that the proposed model is definite in its reliability and predictions. The proposed model demonstrates a slight greater computation duration (59 s) than ablated models, however the modelling gains outweigh each measure, illustrating that all components, optimize and perform predictably for the to each model's overall outcomes.

Table 9 Evaluation of ablation study for Dataset 2

Metric	Without Adaptive FOSSA Optimization	Without GoogLeNet	Without VGG16	Without Transformer	Proposed
Accuracy	0.9785	0.9813	0.9802	0.9769	0.9891
Precision	0.9734	0.9765	0.9752	0.9718	0.9878
Sensitivity	0.9789	0.9819	0.9805	0.9776	0.9926
Specificity	0.9753	0.978	0.9771	0.9739	0.9896
F1-Score	0.9689	0.9721	0.9713	0.9665	0.9847
MCC	0.9744	0.978	0.9763	0.9715	0.9906
NPV	0.9658	0.9696	0.9682	0.9639	0.9857
FPR	0.0247	0.022	0.0229	0.0261	0.0119
FNR	0.0211	0.0181	0.0195	0.0224	0.0111
Computation Time (s)	47.8	52.1	54.3	49.6	65

The ablation analysis for Dataset 2 highlights the significance of each architectural component by comparing the proposed model with variants lacking Adaptive FOSSA optimization, GoogLeNet, VGG16, or Transformer modules. Results indicate that the removal of any component leads to reduced performance across all major evaluation metrics, including accuracy, precision, sensitivity, specificity, F1-score, MCC, and NPV, while increasing error rates such as FPR and FNR. Specifically, the absence of Adaptive FOSSA optimization yields the lowest accuracy (0.9785), whereas excluding GoogLeNet, VGG16, or Transformer also results in consistent performance degradation. In contrast, the full suggested model delivers the maximum precision (0.9891), precision (0.9878), Perceptiveness (0.9926), specificity (0.9896), as well as MCC (0.9906), along with the lowest FPR (0.0119) as well as FNR (0.0111), demonstrating superior predictive capability and robustness. While the computation time of the proposed model (65 s) is slightly higher than its ablated counterparts, the substantial performance improvement validates the critical role of each component in achieving optimal results.

Table 10 Evaluation of ablation study for Dataset 3

Metric	Without Adaptive FOSSA Optimization	Without GoogLeNet	Without VGG16	Without Transformer	Proposed
Accuracy	0.9788	0.9812	0.9799	0.9775	0.9899

Precision	0.9793	0.9818	0.9807	0.9786	0.9903
Sensitivity	0.9805	0.9825	0.9812	0.9798	0.9914
Specificity	0.9813	0.9837	0.982	0.9806	0.9929
F1-Score	0.9801	0.9823	0.9811	0.9793	0.9912
MCC	0.981	0.983	0.9818	0.9802	0.9912
NPV	0.9799	0.9816	0.9805	0.9789	0.9917
FPR	0.0187	0.0163	0.018	0.0194	0.0071
FNR	0.0195	0.0175	0.0188	0.0202	0.0062
Computation Time (s)	40.8	44.2	46.5	42.9	52

In Dataset 3, the ablation findings provide clear evidence the contribution of each component in the proposed architecture includes Adaptive FOSSA optimization, GoogLeNet, VGG16, and Transformer, which are important to the overall performance. If any of these modules are removed, decreases in important evaluation metrics, such as F1-score, MCC, sensitivity, specificity, accuracy, and precision and NPV will occur with an increase in FPR and FNR. For example, the lack of Adaptive Fossa optimization presents the lowest accuracy (0.9788) of the ablated models, whereas removing GoogLeNet, VGG16, or Transformer has approximately and equal performance declines. In comparison, the new model provides the best accuracy (0.9899), precision (0.9903), sensitivity (0.9914), particularity (0.9929), MCC (0.9912), and NPV (0.9917), while yielding the lowest FPR (0.0071) and FNR (0.0062). Therefore, The suggested model computation time (52s) is slightly more than the ablated models; the performance increase speaks toward the synergistic performance of all the parts together in providing the best classification performance.

5. CONCLUSIONS

The proposed LEDL Structure for identifying the type of IoT device that combines the strengths of GoogLeNet, VGG16, and Transformer models with advanced optimization techniques. The framework leverages pruning and quantization to achieve substantial reductions in model size and computational overhead without sacrificing classification accuracy. An adaptive confidence-based fusion strategy enables the ensemble to dynamically select the most reliable model predictions, while the integration of Few-Shot Learning effectively addresses the challenge of identifying rare or previously unseen IoT device types. Furthermore, Sparsemax Attention and Attribution Loss improve the model's focus on the most discriminative features, and SHAP-based interpretability ensures transparency in decision-making, which is critical for trustworthy IoT applications. Extensive experimental evaluations on benchmark datasets demonstrate that our framework outperforms conventional deep learning models, achieving an accuracy (0.9899), precision (0.9903), and remarkably low FNR (0.0062) and FPR (0.0071), highlighting its robustness and reliability for IoT deployments. The suggested method not only advances IoT device identification but also establishes A scalable, explainable, and resource-efficient solution that can be adapted to evolving IoT ecosystems. Future research will concentrate on expanding this framework to assist real-time edge

deployment, adaptive online learning, and integration with broader IoT security and anomaly detection systems.

REFERENCE

1. Kostas, K., Just, M. and Lones, M.A., 2022. IoTDevID: A behavior-based device identification method for the IoT. *IEEE Internet of Things Journal*, *9*(23), pp.23741-23749.
2. Kotak, J. and Elovici, Y., 2023. IoT device identification based on network communication analysis using deep learning. *Journal of Ambient Intelligence and Humanized Computing*, *14*(7), pp.9113-9129.
3. Thouti, S., Venu, N., Rinku, D.R., Arora, A. and Rajeswaran, N., 2022. Investigation on identify the multiple issues in IoT devices using Convolutional Neural Network. *Measurement: sensors*, *24*, p.100509.
4. Kostas, K., Kostas, R.Y., Just, M. and Lones, M.A., 2024. GeMID: Generalizable Models for IoT Device Identification. *arXiv preprint arXiv:2411.14441*.
5. Chowdhury, R.R., Idris, A.C. and Abas, P.E., 2022. Device identification using optimized digital footprints. *arXiv preprint arXiv:2212.04354*.
6. Zahid, H.M., Saleem, Y., Hayat, F., Khan, F.Z., Alroobaea, R., Almansour, F., Ahmad, M. and Ali, I., 2022. A framework for identification and classification of iot devices for security analysis in heterogeneous network. *Wireless Communications and Mobile Computing*, *2022*(1), p.8806184.
7. Almufti, S.M., Hani, A.A., Zeebaree, S.R., Asaad, R.R., Majeed, D.A., Sallow, A.B. and Ahmad, H.B., 2024. Intelligent home IoT devices: An exploration of machine learning-based networked traffic investigation. *Jurnal Ilmiah Ilmu Terapan Universitas Jambi*, *8*(1), pp.1-10.
8. Oyewobi, S.S., Djouani, K. and Kurien, A.M., 2022. Visible light communications for internet of things: Prospects and approaches, challenges, solutions and future directions. *Technologies*, *10*(1), p.28.\
9. Kabir, M.H., Hasan, M.A. and Shin, W., 2022. CSI-DeepNet: A lightweight deep convolutional neural network based hand gesture recognition system using Wi-Fi CSI signal. *IEEE Access*, *10*, pp.114787-114801.
10. Gao, J., Fan, H., Zhao, Y. and Shi, Y., 2023. Leveraging Deep Learning for IoT Transceiver Identification. *Entropy*, *25*(8), p.1191.
11. Madhu, B., Chari, M.V.G., Vankdothu, R., Silivery, A.K. and Aerranagula, V., 2023. Intrusion detection models for IOT networks via deep learning approaches. *Measurement: Sensors*, *25*, p.100641.
12. Lazzarini, R., Tianfield, H. and Charissis, V., 2023. A stacking ensemble of deep learning models for IoT intrusion detection. *Knowledge-Based Systems*, *279*, p.110941.
13. Wang, J., Zhong, J. and Li, J., 2023. Iot-portrait: Automatically identifying iot devices via transformer with incremental learning. *Future Internet*, *15*(3), p.102.
14. Sánchez, P.M.S., Celdrán, A.H., Bovet, G. and Pérez, G.M., 2024. Adversarial attacks and defenses on ML-and hardware-based IoT device fingerprinting and identification. *Future Generation Computer Systems*, *152*, pp.30-42.
15. Koball, C., Rimal, B.P., Wang, Y., Salmen, T. and Ford, C., 2023. IoT device identification using unsupervised machine learning. *Information*, *14*(6), p.320.
16. Chaganti, R., Ravi, V. and Pham, T.D., 2022. Deep learning based cross architecture internet of things malware detection and classification. *Computers & Security*, *120*, p.102779.

17. Awajan, A., 2023. A novel deep learning-based intrusion detection system for IOT networks. *Computers*, *12*(2), p.34.
18. Khan, N.W., Alshehri, M.S., Khan, M.A., Almakdi, S., Moradpoor, N., Alazeb, A., Ullah, S., Naz, N. and Ahmad, J., 2023. A hybrid deep learning-based intrusion detection system for IoT networks. Math. Biosci. Eng, 20(8), pp.13491-13520.
19. Shahin, M., Chen, F.F., Hosseinzadeh, A., Bouzary, H. and Rashidifar, R., 2022. A deep hybrid learning model for detection of cyber attacks in industrial IoT devices. The International Journal of Advanced Manufacturing Technology, 123(5), pp.1973-1983.
20. Keshk, M., Koroniotis, N., Pham, N., Moustafa, N., Turnbull, B. and Zomaya, A.Y., 2023. An explainable deep learning-enabled intrusion detection framework in IoT networks. Information Sciences, 639, p.119000.
21. Liu, X., Han, Y. and Du, Y., 2022. IoT device identification using directional packet length sequences and 1D-CNN. Sensors, 22(21), p.8337.
22. Jain, P., Rathour, A., Sharma, A. and Chhabra, G.S., 2025. Bridging Explainability and Security: An XAI-Enhanced Hybrid Deep Learning Framework for IoT Device Identification and Attack Detection. IEEE Access.
23. Yang, X., Peng, G., Zhang, D. and Lv, Y., 2022. An enhanced intrusion detection system for IoT networks based on deep learning and knowledge graph. Security and Communication Networks, 2022(1), p.4748528.
24. Li, Y., Wang, Y., Liu, X., Zuo, P., Li, H. and Jiang, H., 2023. Deep-reinforcement-learning-based wireless IoT device identification using channel state information. Symmetry, 15(7), p.1404.
25. Teymournezhad, K., Azgomi, H. and Asghari, A., 2022. Detection of counterfeit banknotes by security components based on image processing and GoogLeNet deep learning network. Signal, Image and Video Processing, 16(6), pp.1505-1513.
26. *Network Intrusion dataset(CIC-IDS- 2017)*. (2023, August
27. Kaggle. https://www.kaggle.com/datasets/chethuhn/network-intrusion-dataset
28. *Ton-IoT*. (2023, July 17).
29. Kaggle. https://www.kaggle.com/datasets/amaniabourida/ton-iot
30. *UNSW_NB15*. (2019, January 29).
31. Kaggle. https://www.kaggle.com/datasets/mrwellsdavid/unsw-nb15