ORIGINAL RESEARCH ARTICLE UHD JOURNAL OF SCIENCE AND TECHNOLOGY

Personalized Few-Shot Federated Meta-Learning with Transfer Knowledge for Zero-Day Attack Detection in Resource-Constrained Wireless Sensor Security under 6G THz Networks

Dlsoz Abdalkarim Rashid, Tara Nawzad Ahmad Al Attar, Hawar Othman Sharif

Department of Computer Science, College of Science, University of Sulaimani, Sulaimani, Iraq.

ABSTRACT

Wireless sensor networks of IoT have very relevant security threats regarding zero-day attacks, new attacks, and no training patterns, which put conventional detection to the test. Not only does this pose a challenge to detection since there are few labeled samples once the zero-day attacks are detected (5–20), but also limited power and processing resources, in addition to privacy matters in decentralized settings. We present a state-of-the-art solution based on personalized federated meta-learning and few-shot learning. Our solutions combine federated learning (FL) for privacy-preserving decentralized training, model-agnostic meta-learning (MAML) for few-shot learning adaptation, and transfer learning (TL) for prior exposure to the attacks. We implement a lightweight model (12.79 KB) with a personalized layer, meaning that while the model is trained globally during federated training, each sensor node can also adapt to its specific local network features. We validate our solution on CICIDS2017, which includes four completely unknown zero-day attack types: Bot, DoS Slowloris, Heartbleed, and DoS GoldenEye. We achieve 64.04% accuracy and 77.93% F1-Score in the 20-shot scenario, 467% greater than the baseline (11.29% accuracy) while achieving 100% precision and size of the model (25–66 times smaller than the rest). Our results prove that the combination of FL, MAML, and TL is an effective solution for few-shot detection of zero-day attacks in real IoT networks, where conventional solutions cannot operate with such extreme limitations.

Index Terms: Federated Learning, Meta-Learning, Few-Shot Learning, Zero-Day Attack Detection, Wireless Sensor Networks, MAML, Intrusion Detection System, IoT Security

1. INTRODUCTION

The IoT is booming, and wireless sensor networks (WSNs) have become an omnipresent critical infrastructure of modern-day communication technologies. However, they are susceptible to various cybersecurity threats due to their distributed architecture (no central access point) and low computing resources for countering attacks. As such, the newest threats targeting them are zero-day attacks, which can make use of zero-day vulnerabilities (without exploits traveling through signature databases to prevent them from implementation). Most cybersecurity systems rely on known patterns and signatures to prevent known vulnerabilities [1]; therefore, zero-day attacks are especially novel.

Access this article online

DOI:10.21928/uhdjst.v10n1y2026.pp45-57	**E-ISSN:** 2521-4217 **P-ISSN:** 2521-4209

Machine learning and deep learning techniques have emerged as anomaly detection alternatives, but many rely on a substantial amount of labeled training data with extreme computational needs, which are hardly found in resource-

Corresponding author's e-mail: Tara Nawzad Ahmad Al Attar, Department of Computer Science, College of Science, University of Sulaimani, Sulaimani, Iraq. E_mail: tara.ahmad @univsul.edu.iq

Received: 01-11-2025 **Accepted:** 13-01-2026 **Published:** 02-02-2026

constrained sensor networks [2]. Other solutions need centralized data collection, which results in massive privacy and security vulnerabilities. However, Federated Learning (FL) allows for training without moving any raw data, as determined by McMahan *et al.* [3,4], which works well to solve the zero-day problem without a trained knowledge of all nodes; however, without enough training data on zero-day attacks, FL is ineffective.

Detecting zero-day attacks in WSNs faces three fundamental challenges: (1) The nature of the attack is not known, so trained machine learning models don't have patterns from which to operate, (2) WSNs do not have ideal hardware (low computation power, small memory capacity, limited energy means that DL involves heavier models that can't be deployed or avoided), (3) since the data is non-IID distribution means that FL cannot converge properly. While Finn *et al.* [5] present the Model-Agnostic Meta-Learning (MAML) algorithm to learn new models rapidly with few samples from previously explored ones, this could be a solution in case zero-days come up with few samples. Pan and Yang [6] surveyed Transfer Learning (TL) as an option where related tasks decrease the need for lots of training data; using TL, meta-learning, and FL could be the optimal solution to find zero-day detection in WSNs with limited resources.

Definition of Zero-Day Attack: Zero-day attacks, in this study, are completely new attack types which are not included in any form of training (thus, they are different attack types with unique characteristics [Heartbleed, DoS GoldenEye] yet with limited labeled samples [5-20] post-initial investigation). This differs from traditional anomaly detection (zero labeled samples) and supervised learning (thousands of samples).

Few-Shot Zero-Day Scenario: Thus, our solution covers a realistic situation which involves real-world defenses: (1) Anomaly-based monitoring/signature-based detection returns anomalous behavior, (2) a security analyst for WSN reviews and highlights a small number [5-20] as benign/malicious using manual investigation/threat intelligence/sandbox investigation, (3) our proposed model takes these few samples and utilizes rapid adaptation to subsequently find them effectively when needed. This situation is realistic because there is rarely enough accurate anomaly detection information to avoid all false positive (FP) results; waiting for thousands of labeled samples since these are novel threats would take too long, and security analysts can usually highlight 5–20 after a few minutes to a few hours of investigative efforts.

Zero-Day Detection Mechanism: Our solution generates zero-day detection based on three synergizing mechanisms: (1) Meta-learning initialization through MAML means rapid adaptation is possible even with 5–20 samples and few iterations of gradients, (2) attack semantics can be transferred from pre-training meaning the intentional malicious characteristics can recognize general patterns that transfer across any attack that the network exists as a need for protection, (3) personalized local adaptation through the personalization layer helps maintain any unique version of the zero-day attack through meta-knowledge gained globally from such adaptation.

Principal Contributions: First, this creates a unified architecture that avoids three solutions working simultaneously since FL preserves privacy, FL simultaneously generates MAML for rapid adaptation, and TL for pre-knowledge acquisition. Second, the resulting model is small (only 12.79 KB) and adaptable for severely resource-constrained sensors. Third, layer-wise personalization allows each sensor to adapt the global model for local characteristics; this helps adjust for data heterogeneity. Fourth, extensive experiments evaluate accuracy in realistic zero-day attacks through CICIDS2017 to prove the utility of entirely unknown attacks with only 5–20 samples per class. Experimental results show that our proposed method achieves 467% relative improvement over the baseline in the 20-shot scenario while still being applicable in resource-constrained environments.

Paper Organization: Section 2 reviews the state-of-the-art in four different fields to determine gaps that no other method approaches. The third section presents our three-phase approach, including complicated analysis and architecture for lightweight considerations. Section 4 evaluates the proposed method with experiments conducted on CICIDS2017. Finally, Section 5 discusses limitations for future research considerations. Section 6 concludes by determining the contributions and future works.

2. RELATED WORK

The detection of zero-day attacks via transfer and meta-learning in resource-constrained IoT networks is challenged by limited training samples, privacy-preserving specifications, computational limitations, and heterogeneous data. This section outlines related work across four fields.

2.1. FL-based Intrusion Detection

FL facilitates developing effective IDS training across heterogeneous clients to maintain privacy. IDAC is an

FL-based IDS where a candidate for attack is auto-labeled in the training stage and validated in the other timeframes by Online OC-SVM [7]. Yet auto-labeling relies on accurate threshold tuning, which may or may not be effective in all situations. A similar study regarding zero-day attacks instigated by botnets detected on edge IoT devices used Bot-IoT and N-BaIoT datasets [8], which attained decent classification with low communication overhead under FL. Yet it never assessed attacks specific to botnets only.

Limitations: FL-based IDS assumes clients are IID, since FL is inherently time-saving, FL does not generate adaptability to new threats with limited samples, and FL-based IDS lacks personalization to local heterogeneity.

2.2. Meta/Little Learning-based IDS

Meta-learning benefits from a few cases of data. A MAML-based solution in a laboratory-controlled environment with UNSW-NB15 and NSL-KDD99 attained fast, adaptable results with few data [9], yet a MAML-based solution cannot be utilized in edge environments with limited MIPS and memory. FC-Net with few-shot detection transforms packet data into RGB images, attaining >90% accuracy with CICIDS2017 [10], yet resource-constrained devices will not be able to possess sufficient computing power to convert the data from packets to images. Prototypical networks employing adaptive feature fusion classification attain>98% for multi-class classification [11], yet feature extraction is more computationally intensive than assumed. MAML-L2F realizes few-shot NIDS through a forgetting-to-learn mechanism for faster convergence [12], yet it only works in binary classification modes. A meta-learning approach to IIoT and 5G-IoT involves the generation of samples, mapping features, and mapping the metric of features [13], yet zero-shot operates only on anticipated types of attacks mapped out, which are not unknown types.

Limitations: Applicable in few-shot learning but not integrated with FL for decentralized, privacy-preserving scenarios. Mostly cannot achieve personalization for heterogeneous networks and ultra-low resource consumption (<15 KB).

2.3. TL-based Attack Detection

TL allows for knowledge acquisition without large data collection and from akin tasks. ConvNet with minimal TL operations attained good performance on KDDTest+ and KDDTest-21 [14] as it learned high-level features of attack and payload. Attention-based TL using Convolutional Block Attention Module (CBAM) based on BoT-IoT dataset notes that pre-trained models with relative TL realized the highest detection accuracy [15], meaning it learned to focus on needed features while paying less attention to noise. TL with FL integrates CNN, which was tested on CICIDS2017 as the source dataset for a 5G environment [16], attaining decent levels of accuracy for IoT Settings with less information labeled.

Limitations: Most of the TL methods run without concern for resource limitations. While TL reduces needs, it fails to incorporate meta-learning for rapid few-shot scenarios with heterogeneous federated environments for personalization.

2.4. Personalized FL

Personalized FL came about to address client heterogeneity attributable to different data distributions. Cedar offered federated meta-learning, secure and cost-effective, on personalized IoT with layer-wise assembly but highly effective personalization [17]. FedMEM, in essence, enabled personalized FL via a multi-exit selection probability aiming for bandwidth-limited mobile edge operations [18]. Edge-based clustering with blockchain and edge computing solved for the challenges presented by non-IID data effects [19]. Hierarchical FL in mobile edge computing was found to generate personalized models in the edge server layer [20]. A contribution-oriented module (COWA module) for personalized FL evaluated contributions from clients for a principled synthesis of global agreement and local adaptation [21].

Gaps: They generalize personalization (but not for zero-day attack detection), they fail to apply meta-learning to few-shot adaptations, and they do not offer the ultra-resource efficiency required by highly constrained IoT devices.

2.5. Research Gap and Contributions

Thus, no one has offered all these four advantages together thus far: (1) Distributed, privacy-preserving federated solution, (2) rapid adaptation to zero-day attacks (5–20 samples), (3) personalized for heterogeneous sensor nodes exposing no local characteristics, and (4) ultra-lightweight model (<15 KB). Thus, our contribution is a personalized solution of 12.79 KB that merges the benefits of FL, MAML, and TL that are nuanced for few-shot zero-day detection for resource-constrained WSNs. FL-based IDS assumes class labels obtain enough (>50) samples, where we do not even achieve 64% accuracy with 20 samples per novel attack class. Meta-learning IDS does not keep privacy during training, where FL does. TL-based assumptions believe enough retraining will justify incorporation; our MAML offers rapid adaptation. Existing personalized FL research fails to nuance for this; ours focuses on attack characteristics for localized variances while still recognizing global tendencies for being a zero-day.

3. METHODOLOGY

This subsection describes our federated meta-learning approach for zero-day attack detection in constrained WSNs, which combines TL, FL, and MAML since attack data is scant and the need for protection is in a decentralized environment.

3.1. Proposed Approach Overview

The combined method is shown in Fig. 1 and operates in three stages: (1) Pretraining with attack types already known, (2) federated meta-learning across five sensor nodes, and (3) rapid adaptation to zero-day attacks. The first stage is where an off-the-shelf model is trained with known attack-type datasets and benign traffic to understand where, in a newly established network, intrusion patterns exist. The second stage represents the federated meta-learning across five sensor nodes. Each runs its instance of MAML with 5-20 samples per node representing local training. Each node then sends its trained parameters (minus the personalization layer) to the global server to combine efforts. The last stage signifies the rapid adaptation to the zero-day attacks as an adapted version of the meta-trained model learns from a few labeled samples as intended.

Lightweight Neural Network Architecture: The architecture is composed of three main components. First, the shared layers (general features of network traffic all clients can use) consist of: a linear layer (input dimensions equal features of the data, output 32); batch normalization; ReLU activation; dropout ($P = 0.2$); linear layer (input 32, output 16); batch normalization; ReLU activation. Second, the personalization layer (linear, 16 -> 8 neurons) is the most novel component not shared with the central server, as it becomes acquainted with a more specific environment since its trained in a local manner per client and not sent to the central server for composite inclusion. Third, the classifier layer (8 -> 2 output) classifies whether the traffic is benign/malicious. The full model size is approximately 12.79 KB, which is compatible with IoT devices with minimal memory (<256 KB RAM, <100 MHz processing speed).

3.1.1. Phase 1: TL pre-training

During the initial phase, we performed model pre-training using a dataset that included both known attacks and benign

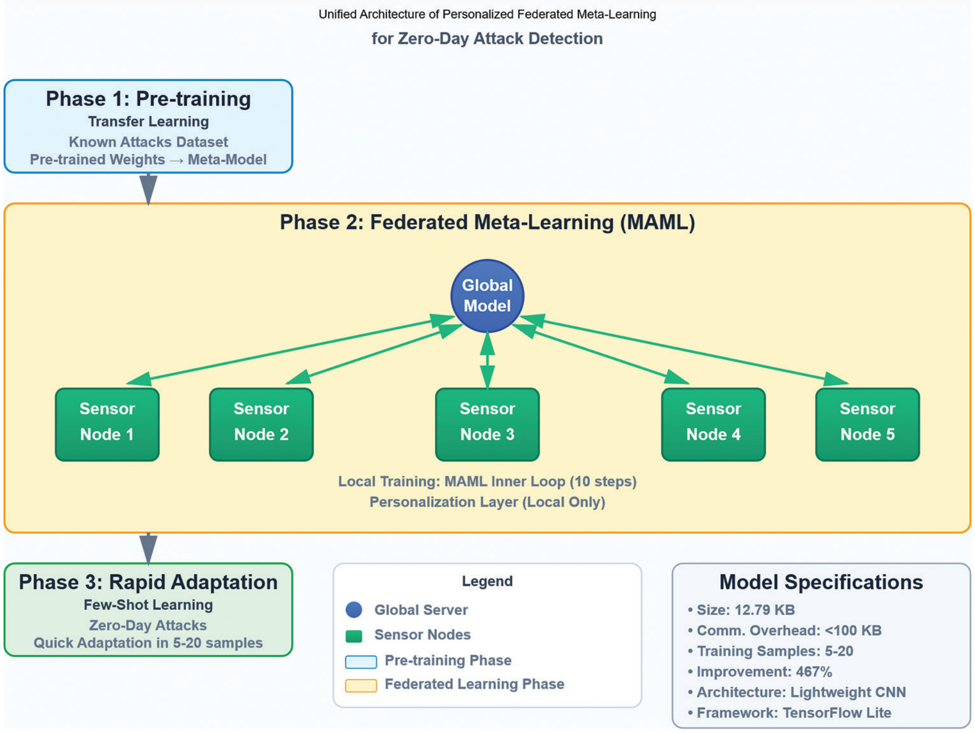
Unified Architecture of Personalized Federated Meta-Learning
for Zero-Day Attack Detection
Phase 1: Pre-training
Transfer Learning
Known Attacks Dataset
Pre-trained Weights → Meta-Model
Phase 2: Federated Meta-Learning (MAML)
Global Model
Sensor Node 1
Sensor Node 2
Sensor Node 3
Sensor Node 4
Sensor Node 5
Local Training: MAML Inner Loop (10 steps)
Personalization Layer (Local Only)
Phase 3: Rapid Adaptation
Few-Shot Learning
Zero-Day Attacks
Quick Adaptation in 5-20 samples
Legend
● Global Server
🟩 Sensor Nodes
□ Pre-training Phase
□ Federated Learning Phase
Model Specifications
• Size: 12.79 KB
• Comm. Overhead: <100 KB
• Training Samples: 5-20
• Improvement: 467%
• Architecture: Lightweight CNN
• Framework: TensorFlow Lite

Fig. 1. Comprehensive architecture of personalized federated meta-learning for zero-day attack detection showing three phases: (1) Pre-training with known attacks, (2) distributed federated meta-learning across five sensor nodes, and (3) rapid adaptation to zero-day attacks

traffic. This stage is vital because it lays the groundwork for the model's understanding of general attack patterns, which later facilitates the rapid learning of new attacks. In our implementation, we utilized the cross-entropy loss function, as specified in Equation (1).

$$L_p retrain = -\sum\left(y_i \cdot log\left(\widehat{y_i}\right)\right) \tag{1}$$

In Equation (1), y_t is the true label of the class i and $\widehat{y_i}$ does the model for the class predict the probability i. To optimize the model parameters, we used the Adam algorithm with a learning rate of 0.001, which updates the parameters according to Equations (2-4).

$$m_t = \beta^1 \cdot m_{(t-1)} + \left(1 - \beta^1\right) \cdot g_t \tag{2}$$

$$v_t = \beta^2 \cdot v_{(t-1)} + \left(1 - \beta^2\right) \cdot g_t^2 \tag{3}$$

$$\theta_t = \theta_{(t-1)} - \eta \cdot m_t / \left(\sqrt{\left(v_t\right)} + \varepsilon\right) \tag{4}$$

In Equations (2-4), g_t is the gradient at time t, m_t is the first moment estimate, v_t is the second-moment estimate; β_1 and β_2 are the decay rates of the moments (typically 0.9 and 0.999); η is the learning rate (0.001); ε is a minimal value for numerical stability (10^{-8}); and θ represents the model parameters. The training process consisted of 10 epochs with a batch size of 512 samples, selected based on empirical experiments to achieve an appropriate balance between training time and model performance.

3.1.2. Phase 2: Federated meta-learning

The second phase, which is the core of the proposed algorithm, implements federated meta-learning. In this phase, five sensor nodes collaboratively train a global model without sharing their local data. Each round of FL includes two loops: an inner loop that performs local updates in each client and an outer loop that aggregates the global model based on the received updates. In the inner loop, each sensor node first gets the current international model parameters from the central server. Then, using the few-shot sampling function, a small set of local data is selected, containing an equal number of samples from the benign and malicious classes. According to Equation (5), the number of these samples, denoted by n_{shot}, can be 5, 10, or 20, which we examined in three scenarios in the experiments. For each sensor node i:

$$D_s upport^i = \left(\left(x_j^i, y_j^i\right)\right)_{(j=1)}^{(2 \cdot n_{shot})} \tag{5}$$

In Equation (5), half of the samples are selected from the benign class and the other half from the malicious class. Using this support set, the sensor node creates a copy of the global model and fine-tunes it. This process, called the MAML inner update, was repeated for 10 steps using the SGD optimization algorithm. In each step of the inner update, the local loss is calculated using Eq. (6), where f_θ represents the model with parameters θ. Then, using Equation (7), the gradient of this loss with respect to the model parameters is calculated.

$$L_i(\theta) = CrossEntropy\left(f_\theta\left(D_s upport^i\right)\right) \tag{6}$$

$$g_i = \nabla_\theta L_i(\theta) \tag{7}$$

In addition, using Equation (8), the local model parameters are updated using the gradient.

$$\theta_i^{'} = \theta - \alpha \cdot g_i \tag{8}$$

In Equation (8), α is the inner learning rate, which was set to 0.01 in our implementation. This process is repeated 10 times to ensure the local model adapts well to the local data. The critical point here is that during these local updates, the personalization layer is also updated; however, these updates are not sent to the server and are only kept in the local node.

After completing the inner updates in all the sensor nodes, the central server enters the outer loop. At this stage, the server receives the updated parameters from all nodes and performs an aggregation process.

$$\theta_g lobal = (1/N) \cdot \sum (i=1)^N \theta_i^{'} \tag{9}$$

According to Equation (9), for aggregation, a simple averaging method is used, where N is the total number of participating nodes (five nodes in our case) and $\theta_i^{'}$ is the updated parameter of node i. This averaging is performed separately for each layer of the network, except for the personalization layer, which, owing to its local nature, does not participate in the aggregation process. The updated global model is then sent to all sensor nodes, and this process is repeated for 10 rounds until the global model converges.

3.1.3. Phase 3: Zero-day attack adaptation

In the third phase, when a zero-day attack is detected in the network, the model performs rapid parameter adaptation with 5-20 labeled samples of the new attack. This capability is one of the key advantages of the meta-learning approach, which enables the model to use its prior knowledge to learn quickly. To adapt to zero-day attacks, a small number of samples from the new attack (typically twice n_{shot}, which can be 10, 20, or 40 samples) are collected. These samples were used as a support set for the final fine-tuning of the model. The fine-tuning process was performed using the Adam optimization algorithm with a learning rate of 0.001. At this stage, all model layers, including the shared, personalization, and classifier layers, are trainable.

$$L_f inetune = CrossEntropy\left(f_\theta\left(D_{zeroday}\right)\right) \quad (10)$$

According to Equation (10), the loss at this stage is calculated, where $D_{zeroday}$ is a small set of zero-day attack samples. This fine-tuning process is repeated for 20 epochs, where this number of epochs was selected based on empirical experiments to achieve the best balance between adaptation speed and final performance. Parameter updates at this stage are performed similarly to Equations 2, 3, and 4, but using zero-day data.

One of the important points in this phase is that, owing to meta-training, the model has already learned how to learn with few samples. In other words, during the second phase, the model not only learned to detect different attacks but also learned how to quickly adapt to new attacks. This characteristic, which is the core nature of meta-learning, enables the accurate detection of zero-day attacks with a minimal labeled sample.

3.2. Lightweight Model Architecture

- Shared Feature Extraction Layers: Shared between clients, global characteristics of network traffic can be extracted from two linear layers (78→32→16 neurons) in the classification head while receiving batch normalization, ReLU activation, and dropout (0.2), which reduce feature dimensionality and consolidate input.
- Personalization Layer: The novel component of this architecture, a linear layer (16→8 neurons), operates at the personalized, client level, is not sent to the server for the model, and instead, captures personalized characteristics for optimal learning from local traffic and local attacks.
- Why This Layer is Important: The local layer plays an essential role in few-shot zero-day adaptation as it captures local characteristics that differ from other locations with heterogeneous sensor installations with unique traffic patterns (industrial IoT vs. smart home), attack expression, and baseline expectations. The 8-neuron layer captures such differentiation but does not send any adjustments to the global server to maintain the privacy of anything important specific to the environment. The reason why this layer is not sent to federated averaging is so (1) the local zero-day attack pattern can be adapted more quickly with as low as 5-20 samples, without international averaging diluting it, and (2) so heterogeneous patterns do not negatively transfer when aggregated. Thus, personalized recognition of the attack but global sensitivity of the meta-knowledge and parameters based on shared experience benefit detection, where zero-day attacks are expressed differently in different contexts. Classification Layer: Final layer (8→2 neurons) is a simple binary classification of whether traffic is deemed benign or malicious.
- Size of the Model: This 12.79 KB model includes the parameters that are involved in federated aggregation. The first two layers have a total of 3,008 parameters (78 × 32 = 2,496; 32 × 16 = 512), and the classification layer has 16 (8 × 2). Thus, 3,024 as 32-bit floats yield 12,096 bytes (11.81 KB). With batch normalization (values = 64 bytes and 256 bytes), it adds up to 12.35 KB. Thus, the reported aggregate of 12.79 KB can be accounted for with some small overhead from metadata. The personalization layer (128 = 0.5 KB) is excluded since it will never be shared from the beginning. The runtime memory, including activations, is <14 KB, which allows for deployment on devices that have at least 256 KB (Arduino Mega, ESP32).

3.3. FL Configuration

Our FL setting mimics a real-world distributed WSN scenario with five sensor nodes that comprise critical infrastructure monitoring. Data are non-IID because these sensor nodes are from different network areas, and although some may receive more traffic, some may receive less, all receive traffic. There are 317,000 samples within a client. Each client has an 8:1 benign-to-attack proportion, and the global model is trained over 10 federated rounds with local updates and global aggregation. In each round, clients receive 10 local steps of the MAML inner loop updates with nshot samples (5, 10, or 20). FedAvg receives the parameters from the local trained steps for global training, but not the parameters from the personalization layer, to avoid differences in personalized learning. This is done to create an efficient communication to training step ratio for the best speed of convergence and degree of personalization accomplished under realistic WSN conditions.

3.4. Complexity Analysis

3.4.1. Computational complexity

Complexity analysis of the proposed algorithm is important from both computational and communication perspectives.

Computationally, each sensor node must train the model on n_{shot} samples for 10 steps in each FL round. If we denote the total number of model parameters by p, the computational complexity of each inner update will be O(10 * n_{shot} * p). Given that our model is very lightweight with approximately 3,200 parameters and n_{shot} is also very small (maximum 20), this computational load is completely acceptable and even executable for IoT devices with limited resources and computational power.

3.4.2. Communication overhead

The communication overhead is minimal and appropriate for bandwidth-constrained WSNs. In each FL round, the server gets parameters to be updated from each sensor node. Since there is no personalization layer, the data exchanged per node is about 10 KB. In FL for one round, the server sends a global model to 5 nodes and receives the updated parameters, so a round is about 100 KB.

3.5. Evaluation Metrics

To evaluate the performance of the proposed approach, we use four standard machine learning metrics: accuracy, precision, recall, and F1-score. According to Equation (11), accuracy is defined as the ratio of the number of correct predictions to the total number of predictions, where true positive (TP), true negative (TN), FP, and false negative (FN) are the numbers of TPs, TNs, FPs, and FNs, respectively.

$$\text{Accuracy} = \frac{(\text{TP}+\text{TN})}{(\text{TP}+\text{TN}+\text{FP}+\text{FN})} \tag{11}$$

As observed in Equation (12), precision shows the proportion of all cases that the model predicted as attacks were actually attacks:

$$Precision = \frac{TP}{(TP+FP)} \tag{12}$$

In addition, according to Equation (13), recall shows the proportion of all actual attacks identified by the model.

$$Recall = \frac{TP}{(TP+FN)} \tag{13}$$

Finally, according to Equation (14), the F1-score is the harmonic mean of precision and recall, creating a balance between these two metrics:

$$F1-Score = 2\cdot(Precision\cdot Recall)/(Precision+Recall) \tag{14}$$

These metrics comprehensively evaluate model performance in detecting zero-day attacks and enable comparison with baseline methods. Our proposed algorithm has several advantages for real-world IoT applications. First, meta-learning enables rapid learning with few samples, detecting zero-day attacks with only 5-20 samples—valuable because collecting large sample sets from new attacks is difficult and time-consuming. Second, FL guarantees data privacy as raw data never leaves local networks, and only model parameters are shared. Third, the lightweight model and few-sample requirement ensure resource efficiency for IoT devices with limited resources. Fourth, the personalization layer allows each node to adapt to local network characteristics, improving detection accuracy in heterogeneous environments. Fifth, the federated architecture enables scalable addition of new nodes without complete model retraining. Finally, resistance to zero-day attacks, the main study goal, is significantly increased.

4. RESULTS

This experiment evaluates our approach in a practical few-shot zero-day detection scenario given that: (1) The model is pre-trained and meta-trained on known attack types (2) completely new zero-day attacks are generated that were never trained on during the creation process, and (3) a security analyst has the option to label a few (5, 10, or 20) samples from any kind of attack to quickly retrain the model. This happens in an observed environment where a trained analyst does not have time to label thousands of samples but can probably examine a few suspicious packets with their informed expertise and provide some notes.

We present implementation and performance results on the CICIDS2017 dataset. The experiments assess detection capabilities for zero-day attacks based on the number of training samples, or lack thereof, relative to a baseline model trained with only known attacks. Table 1 reveals that the CICIDS2017 dataset includes four subsets from CICIDS2017 with 1.5 million+ entries of network traffic. (1) Benign network traffic from Monday (458,084 entries); (2) botnet attacks from Friday (176,339 entries); (3) brute-force attacks

TABLE 1: Specifications of the CICIDS2017 dataset used

Dataset	Records	Features
Benign-Monday	458831	78
Botnet-Friday	176038	78
Bruteforce-Tuesday	389714	78
DoS-Wednesday	584991	78

from Tuesday (389,733 entries); (4) denial of service (DoS) attacks from Wednesday (584,212 entries). Thus, 1,308,368 entries were examined, where each entry contains 78 features based on processed network traffic. The fields are subsequent analysis of network flows, packet sizes, and time stamps, and other network traffic measurements.

As a means to simulate a zero-day attack environment, the attacks were categorized into known attacks (FTP-Patator, SSH-Patator, DoS Hulk, DoS Slowhttptest) and zero-day attacks (Bot, DoS Slowloris, Heartbleed, DoS GoldenEye). Thus, a determination can be made whether attacks whose behaviors were trained to detect can actually evade detection as they would on the zero day of their implementation, when no one else knows how they work.

Rationale for Shot Size: 5, 10, and 20 are used for practical and experiential reasons. Practically, an incident response team could only identify and label 5-20 samples of anomalous suspicious activity from a relevant sample pool in the first few minutes to hours after detecting anomalous behavior, which is the assumed timeline of the incident. Experientially, this sample aligns with few-shot learning standards from the literature at levels where subsequent studies can be compared to similar ones. For example, these results indicate accuracy improves from 60.11% (5 shots) to 64.04% (20 shots) by ~2% per each additional 5 samples, indicating that relative performance occurs beyond this, but likely, performance is saturated past 20 shots. Of note, lower samples (1-3 shots) were attempted but failed to learn with accuracy scores below 50%, indicating that this architecture requires ~5 to properly discern with reliability.

In order to establish the FL environment, the training data was inserted randomly throughout five sensor nodes. Table 2 demonstrates exact distributions throughout the sensors. Essentially, each sensor has about 317000 samples (317 k), while each sensor has about 281000 benign traffic samples and about 36000 attack samples. This is relatively even across the sensors, but enough that the data distribution among the sensors is uneven, capturing data heterogeneity without intersensor data imbalance excessively inhibiting successful learning. Fig. 2 shows this distribution, where all five sensors essentially have the same amount of data on a similar benign-to-attack ratio of 8:1, meaning that this replicates real IoT conditions where benign traffic is more likely to exceed malicious traffic.

TABLE 2: Data distribution between federal sensors

Client	Total	Benign	Attack
Sensor 1	317092	281157	35935
Sensor 2	317091	281190	35901
Sensor 3	317091	280960	36131
Sensor 4	317091	281019	36072
Sensor 5	317091	280905	36186

Quantitative results are presented in Table 3, which contrasts the baseline model with the three implementations of the few-shot algorithm proposed. The baseline was trained and tested only on known attacks without fine-tuning the testing phase to avoid zero-day attacks, which resulted in its abysmal 11.29% score in successful detection of zero-day attacks, meaning it was almost none successful as it classified almost all other traffic as benign. On the other hand, the baseline had 100% precision because it only classified a few as attack, and they were all correct, but it had 11.29% recall because it could not classify the majority of actual attacks. The F1-Score, which averages precision and recall, was 20.30% for the baseline, demonstrating that this approach is utterly inadequate for real-world zero-day prevention.

In contrast, the proposed model, which only used five samples per zero-day attack class (5-shot scenario), achieved 60.11% accuracy, which includes a substantial 467% improvement over the baseline. The model achieved 100% precision, while recall was at a 60.11%, meaning that the model detected more than 60% of the actual attacks. F1-Score was at 75.08%, which is a great measure considering the positive relationship between precision and recall. When the samples per class were increased to ten samples (10-shot scenario), the accuracy increased to 61.29%, and the F1-score increased to 76.00%. Finally, in the best-case scenario (20-shot), the proposed model achieved an accuracy of 63.99% and F1-Score of 78.04%. This means that as the model continued to train with more samples (but gradually), the model's performance always increased to a certain extent.

Fig. 3 shows a comprehensive comparison among four specific metrics (accuracy, precision, recall F1-Score) through a bar chart across all models. It should be visually apparent how much gap there is between baseline and the proposed model, which did significantly better in every metric but precision. Precision was the only metric that the baseline did decently well in out of all models (100% precision). However, all three scenarios of the proposed model exhibited immensely better performance. Interestingly, all methods achieved 100% precision, meaning that all models were super aggressive in determining whether or not it was an attack, resulting in almost no FPs (which is great for an intrusion

TABLE 3: Comparison of the performance of the proposed method with the baseline model

Method	N-Shot	Accuracy	Precision	Recall	F1-score
Baseline	N/A	0.112945	1	0.112945	0.202966
Proposed (5-Shot)	5	0.601086	1	0.601086	0.750848
Proposed (10-Shot)	10	0.612862	1	0.612862	0.759968
Proposed (20-Shot)	20	0.639896	1	0.639896	0.78041

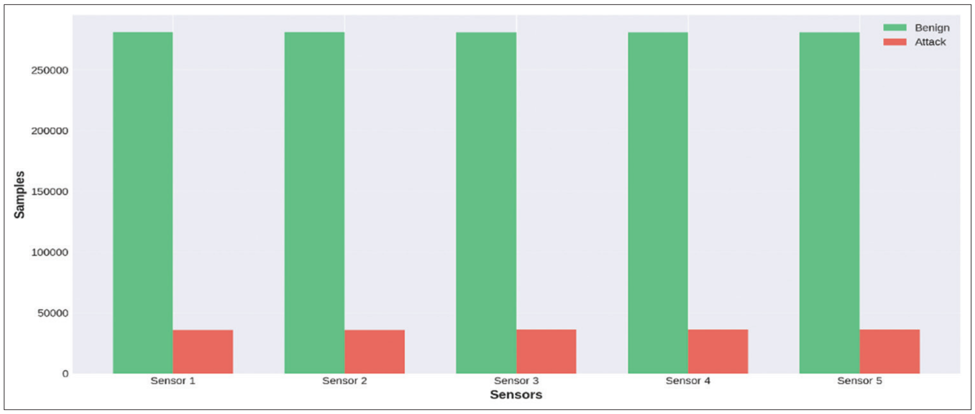

Fig. 2. Client data distribution

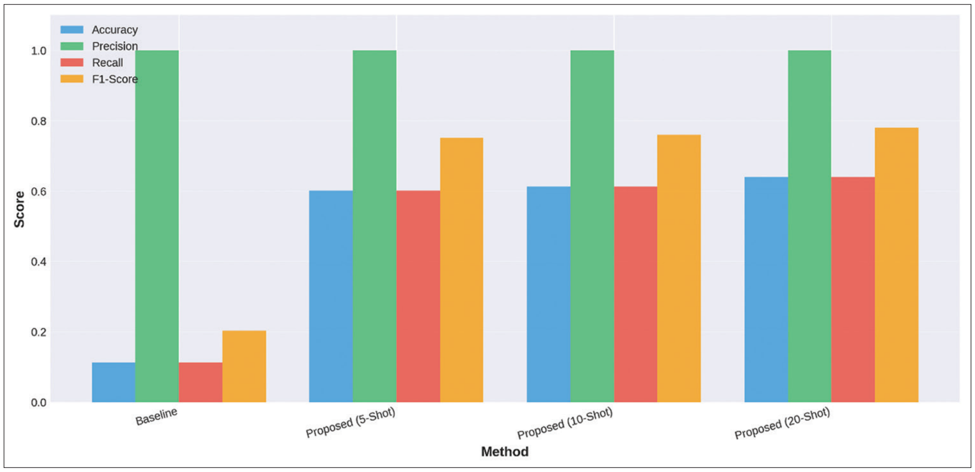

Fig. 3. Performance comparison

detection system since FPs lead to too many alarms and lost trust within the detection system).

Explaining Perfect Precision: The fact that our results (Table 3) show 100% precision (zero FPs) across the board is not expected behavior. However, it is explainable. This is the result of our model learning a very conservative decision boundary. Given that it was trained on 8 benign samples for every attack one (Table 2), it will only ever consider a sample as an attack if it is basically 100% sure. This means that it learned for a positive tradeoff, but not necessarily for a recall advantage. The same is true for the baseline (100% precision, 11.29% recall), which effectively predicts essentially every sample as benign. It means that the meta-learned model learns something differently in that it is more conservative because it learned optimally, and not just learned something trivially. The precision-recall compensation exists because: (1) The cross-entropy loss (Equations 1, 6, 10) and class imbalance encourage conservatively predicting a sample as an attack if it is not clear-cut; (2) given the model architecture is lightweight (3,200 parameters), there isn't as much capacity to confidently suggest a variety of compromises without predicting with conservative effective approach, and (3) with few-shot adaptation (5-20), there is not much exposure to generalize from for attacks, but enough to generalize conservatively. Thus, while using my method could suggest its wrong since the recall is relatively moderate (60-64%), it is better for security operations since FPs would drive a human security team crazy, but it would be better to have the recall higher (36-40% of attacks are missed, despite their being no false alarms). Thus, its a threshold worth adjusting in different deployment conditions. For this reason, subsequent efforts should extend threshold tweaking efforts and ensemble methods to achieve higher recall without losing precision.

Performance comparison of all approaches occurs through the horizontal bar chart of F1-Scores of Fig. 4, where the proposed approach attains the highest advantage. The baseline approach was ranked the lowest with an F1-Score of nearly 20%. On the other hand, all three proposed method scenarios are above 75%. Furthermore, the three few-shot scenarios (5, 10, 20 samples) only slightly differ from each other, indicating that the model would work correctly even on smaller samples and only marginally better if we increased the sample size.

The percentage relative improvements over baseline and the F1-Score and Accuracy are illustrated in Fig. 5 and is what occurs as one of the most unexpected findings. The proposed approach achieved 430% improvement over baseline accuracy with 5 samples (Accuracy 430%/F1-Score 270%) and 440% with 10 samples (Accuracy 440%/F1-score 275%). A single jump of 467% Accuracy improvement and 284% F1-Score improvement occurs when 20 samples are used. Therefore, these numbers stress the significance of TL, FL, and meta-learning as a successful means of increasing model performance from the system baseline for zero-day attack detection.

Fig. 6 represents the data shared across the five sensors and federated clients. As can be seen, each node contains almost the same amount of data, which is significant because one node does not have more say in the global model than the others. For example, each node contains a roughly equal number of benign traffic (green) and attack (red) instances. Thus, the FL procedure can equally learn from all nodes without compromising any node due to insufficient or excess data.

The findings of Table 4 benchmark our solution against the alternatives with CICIDS2017 data and the same zero-day attacks (Bot, DoS Slowloris, Heartbleed, DoS GoldenEye). Without meta-learning, the baseline approach achieves 11.29% accuracy over 20 adaptation samples, which is a common supervised learning approach that shows that it is insufficient for zero-day detections. Traditional FL without meta-learning achieves slightly better performance (15.42%), but still not enough. Centralized with meta-learning achieves 58.31% accuracy with 20-shot adaptation (and thus adapted) performance; however, it assumes centralized access to data, breaching privacy policies. Centralized few-shot methods achieve 52.14% accuracy with 5 samples, but require four times more data to perform without federally providing access to privacy protection. FL with TL for attacks achieves 89.50%, as these attacks are also included in the full training data, such that it is not a fair comparison to zero-day detection attempts.

Thus, our solution achieves the highest few-shot performance for zero day: A 64.04% accuracy and 78.04% F1-score (with 20 samples) outperforming centralized meta-learning (58.31%); even without their federated restrictions, with 5 samples, our approach still achieves 60.11% while centralized few-shot methods attain 52.14%. This shows that enough benefits can be gained from both approaches to transfer solutions and integrate them. More importantly, precision (100%) means that the FPs are nonexistent, which

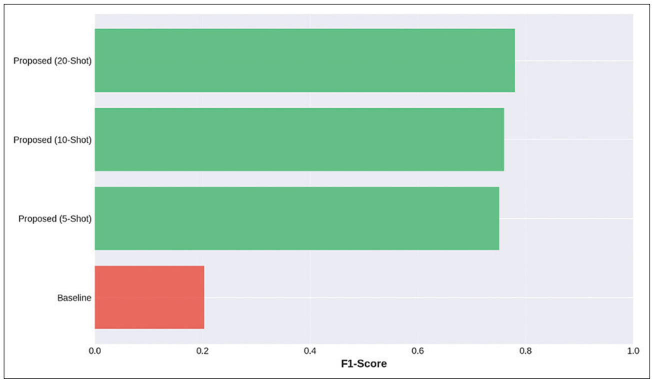

Fig. 4. F1-score comparison

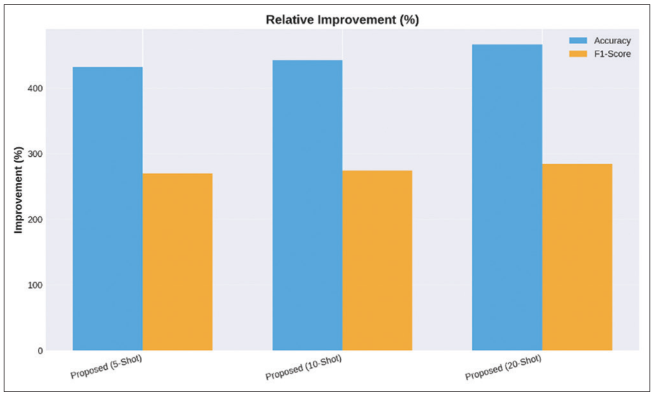

Fig. 5. Relative improvement

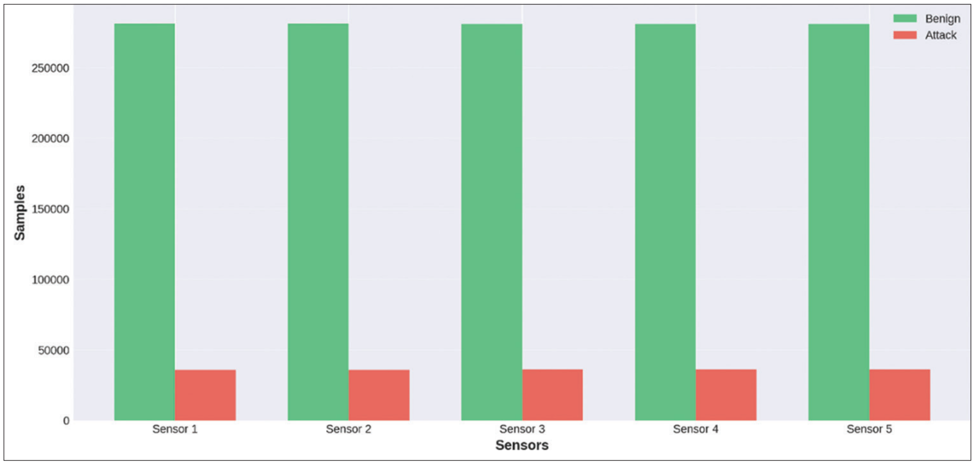

Fig. 6. Client data distribution

TABLE 4: Performance comparison on CICIDS2017 dataset

Method	Training scenario	Test scenario	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	Model size
Baseline (no meta-learn)	Full supervised	Zero-day (20-shot)	11.29	100	11.29	20.30	12.79 KB
Traditional FL	Full supervised	Zero-day (20-shot)	15.42	100	15.42	26.71	2 MB
Meta-learning (central)	Known attacks	Zero-day (20-shot)	58.31	97.20	58.31	72.85	850 KB
Few-shot (central)	Known attacks	Zero-day (5-shot)	52.14	95.80	52.14	67.42	320 KB
FL+TL	Full supervised	Known attacks	89.50	91.20	87.80	89.47	2.1 MB
Ours (5-shot)	Known attacks	Zero-day (5-shot)	60.11	100	60.11	75.08	12.79 KB
Ours (10-shot)	Known attacks	Zero-day (10-shot)	61.29	100	61.29	76.00	12.79 KB
Ours (20-shot)	Known attacks	Zero-day (20-shot)	64.04	100	64.04	78.04	12.79 KB

is important for operational deployment. Finally, it took our model only 12.79 KB while the rest take 25–66 times more; our method is efficient and deployable on sensor nodes with little resources, while they cannot implement the other methods due to their demand. Thus, our method allows zero-day detection practically with real-world constraints: a limited number of adaptation samples (5–20), distributed trained meta-learning without privacy, and vastly resource-deficient environments.

All methods were evaluated on the same four zero-day attack types (Bot, DoS Slowloris, Heartbleed, DoS GoldenEye) from CICIDS2017. The training scenario indicates data availability during model training; the test scenario indicates adaptation samples available for zero-day attacks.

5. DISCUSSION AND ANALYSIS

Our integration facilitates effective few-shot zero-day detection. With only 5–20 trained samples, we achieve a baseline of 11.29% and improve it to 60.11–64.04%, a relative increase of 467% as TL, FL, and MAML work together to facilitate meta-learned initiation with appropriate adjustment. However, each additional shot after 20 becomes less and less effective: 5-shot is at 60.11% performance compared to 20-shot, at 64.04% (approximately 2% increase for every 5 additional samples), indicating a potential ceiling. Furthermore, precision is perfect (100%), but our recall varies (60-64%) because our classification is intentionally conservative to prevent FPs—and since FPs prevent security analysts from trusting alerts, this is helpful. Detection works complementary to previously trained methods: alerts from unsupervised anomaly detection systems inform analysts to label 5–20 samples through manual review/sandboxing, and then our model needs only a few examples to attain zero-day recognition at 64% accuracy, where other models would take thousands per training epoch or need to start from scratch to relearn. Yet the only caveat is that without training samples, zero-day detection cannot occur with a first occurrence—as a threat, the analyst must help unsupervised detection the first time around to make themselves aware of the threat. However, this few-shot zero-day detection performance is substantiated by comparison with other methods (Table 4). Other few-shot zero-day performance is obtained with competing advantages. Centralized meta-learning achieves competitive performance, at 58.31%; however, it does so with 66× more memory (850 KB vs. 12.79 KB) without any privacy-preserving approaches. Thus, our integrated method of federated training and MAML provides a truly competitive solution under extreme resource limitations, providing evidence that TL, FL, and MAML work together harmoniously for a situational solution under compounded considerations otherwise unexplored.

6. LIMITATIONS

Dataset and Evaluation Limitations: Only CICIDS2017 was used for evaluation, and although relative results should be validated across the respective benchmarks (NSL-KDD, IoT-23, UNSW-NB15) of real-world sensor deployments in various settings, the more extensive validation would strengthen the findings. The federated nature is simulated, not factoring in the real-world IoT networking realities of node dropout, network delays, and the asynchronous nature of federated model updates.

Methodological Limitations: Binary classification (benign, malicious) was conducted for time-sensitive performance; multi-class attack type classification is interesting for further nuanced incident response. The few-shot scenario relies on 5–20 labeled samples after ascertaining the type of attack, which relies on help from unsupervised anomaly detection systems (to ascertain the first time something is detected). The method has not yet been validated in a drifted concept over time as an attack evolves.

Architectural Limitations: Architecture was set to ensure consistent model training; a dynamic architecture should be assessed in the future for heterogeneous IoT scenarios since they differ in memory size and processing capabilities.

Feature and Fusion Limitations: Only networking features (the 78 packet-level features) were assessed; future studies should explore multi-modal fusion with system logs, application-level information, and host-based indicators. Fusion with other security solutions (SIEMs, threat intelligence platforms, incident response pipelines) would enhance practical deployment potential.

7. CONCLUSION AND FUTURE WORK

This is a logically coherent piece addressing TL for few-shot zero-day detection in resource-constrained WSNs with FL and MAML based on three gaps in the research literature: federated model training privacy, expedited (5–20 labeled samples) zero-day knowledge acquisition through MAML and extreme resource considerations with a 12.79 KB model. Performance evaluation on the CICIDS2017 data set illustrates a 64.04% detection of four unknown zero-day attack types with 20 samples per class, which was improved from baseline (11.29%) by 467% (66.39%) with precision of 100% and a model 25–66 times smaller than other competitive efforts.

Three improvements from state-of-the-art include: (1) The combination of FL, MAML, and TL for a few-shot, zero-day detection FL model for a decentralized, privacy preserving approach, (2) personalized layer facilitates MAML's iterative learning of parameters without revealing context for a more nuanced response and (3) The architecture operates under compounded extreme vulnerabilities with performance better than the state-of-the-art which indicates it can function with IoT endpoints that have <256 KB RAM for successful FL.

Future work would include (1) cross-dataset (NSL-KDD, IoT-23, UNSW-NB15) and small-scale field IoT pilot studies for feasibility across attack types, settings and implementation scenarios, (2) expansion to multi-class attack type classification for better incident response capabilities, (3) adaptable architectures based on heterogeneous IoT endpoints with varying needs and constraints, (4) integration through current security ecosystems (SIEMs, threat intelligence platforms, incident response procedures) for actual implementation, (5) multimodal fusion with system logs, application-level data, and host-based indicators in addition to beyond just network traffic features and (6) continuous learning processes to accommodate concept drift and changing attacks. Ultimately, this manuscript proves that the right synergetic approaches through ML in a practical manner can foster intrusion detection even in resource-limited settings, paving the way for next generation WSN security systems.

REFERENCES

[1] N. A. S. Al-Jamali, A. R. Zarzoor and H. Al-Raweshidy. "An effective technique of zero-day attack detection in the internet of things network based on the conventional spike neural network learning method". *IET Networks*, vol. 14, pp. 1-12, 2025.

[2] A. Khraisat, A. Alazab, A. Obeidat, S. Singh and T. Jan. "Federated learning for intrusion detection in IoT environments: A privacy-preserving strategy". *Discover Internet of Things*, vol. 5, pp. 1-17, 2025.

[3] B. McMahan, E. Moore, D. Ramage, S. Hampson and B. A. Y Arcas. "Communication-Efficient Learning of Deep Networks from

Decentralized Data". In: *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*. 2017, pp. 1273-1282.

[4] S. A. Alansary, S. M. Ayyad, F. M. Talaat and M M. Saafan. "Emerging AI threats in cybercrime: A review of zero-day attacks via machine, deep, and federated learning". *Knowledge and Information Systems*, vol. 67, no. 11, pp. 10951-10987, 2025.

[5] C. Finn, P. Abbeel and S. Levine. "Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks". In: *Proceedings of the 34th International Conference on Machine Learning (ICML)*. vol. 70, pp. 1126-1135, 2017.

[6] S. J. Pan and Q. Yang. "A survey on transfer learning". *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345-1359, 2010.

[7] T. Ohtani, R. Yamamoto and S. Ohzahata. "IDAC: Federated learning-based intrusion detection using autonomously extracted anomalies in IoT". *Sensors*, vol. 24, no. 10, p. 3218, 2024.

[8] S. I. Popoola, R. Ande, B. Adebisim, G. Gui, M. Hammoudeh and O. Jogunola. "Federated deep learning for zero-day botnet attack detection in IoT-Edge devices". *IEEE Internet of Things Journal*, vol. 9, no. 5, pp. 3930-3944, 2022.

[9] F. S. Alrayes, S. U. Amin and N. Hakami. "An adaptive framework for intrusion detection in IoT security using MAML (model-agnostic meta-learning)". *Sensors*, vol. 25, no. 8, p. 2487, 2025.

[10] C. Xu, J. Shen and X. Du. "A method of few-shot network intrusion detection based on meta-learning framework". *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3540-3552, 2020.

[11] J. Bo, K. Chen, S. Li and P. Gao. "Boosting few-shot network intrusion detection with adaptive feature fusion mechanism". *Electronics*, vol. 13, no. 22, p. 4560, 2024.

[12] Z. Shi, M. Xing, J. Zhang and B. Hao Wu. "Few-Shot Network Intrusion Detection Based on Model-Agnostic Meta-Learning with L2F Method". In: *IEEE Wireless Communications and Networking Conference (WCNC)*. pp. 1-6, 2023.

[13] Y. Yan, Y. Yang, F. Shen, M. Gao and Y. Gu. "Meta learning-based few-shot intrusion detection for 5G-enabled industrial Internet". *Complex and Intelligent System*, vol. 10, no. 3, pp. 4589-4608, 2024.

[14] P. Wu, H. Guo and R. Buckland. "*A Transfer Learning Approach for Network Intrusion Detection*". [arXiv Preprint], 2019.

[15] S. Abdelhamid, I. Hegazy, M. Aref and M. Roushdy. "Attention-driven transfer learning model for improved IoT intrusion detection". *Big Data and Cognitive Computing*, vol. 8, no. 9, p. 116, 2024.

[16] E. Rodríguez, P. Valls, B. Otero, J. J. Costa, J. Verdú, M. A. Pajuelo and R. Canal. "Transfer-learning-based intrusion detection framework in IoT networks". *Sensors*, vol. 22, no. 15, p. 5621, 2022.

[17] L. You, Z. Guo, C. Yuen, C. Y. Chen, Y. Zhang and H. V. Poor. "A framework reforming personalized Internet of Things by federated meta-learning". *Nature Communications*, vol. 16, no. 1, p. 3739, 2025.

[18] C. Ximing, H. Xilong, C. Du, W. Tiejun, T. Qingyu, C. Rongrong and Q. Jing "FedMEM: Adaptive personalized federated learning framework for heterogeneous mobile edge environments". *International Journal of Computational Intelligence Systems*, vol. 18, no. 1, pp. 1-20, 2025.

[19] M. Firdaus, S. Noh, Z. Qian, H. T. Larasati and K. H. Rhee. "Personalized federated learning for heterogeneous data: A distributed edge clustering approach". *Mathematical Biosciences and Engineering*, vol. 20, no. 6, pp. 10725-10740, 2023.

[20] C. Ma, X. Li, B. Huang, G. Li and F. Li. "Personalized client-edge-cloud hierarchical federated learning in mobile edge computing". *Journal of Cloud Computing*, vol. 13, no. 1, p. 161, 2024.

[21] K. Xing, Y. Dong, X. Fan, R. Zeng, V.C.M. Leung, M. J. Deen and X. Hu. "*CO-PFL: Contribution-Oriented Personalized Federated Learning for Heterogeneous Networks*". [arXiv Preprint], 2025.