FedP3E: Privacy-Preserving Prototype Exchange for Non-IID IoT Malware Detection in Cross-Silo Federated Learning

Rami Darwish, Mahmoud Abdelsalam, Sajad Khorsandroo, Kaushik Roy

Abstract—As IoT ecosystems continue to expand across critical sectors, they have become prominent targets for increasingly sophisticated and large-scale malware attacks. The evolving threat landscape, combined with the sensitive nature of IoTgenerated data, demands detection frameworks that are both privacy-preserving and resilient to data heterogeneity. Federated Learning (FL) offers a promising solution by enabling decentralized model training without exposing raw data. However, standard FL algorithms such as FedAvg and FedProx often fall short in real-world deployments characterized by class imbalance and non-IID data distributions-particularly in the presence of rare or disjoint malware classes. To address these challenges, we propose FedP3E (Privacy-Preserving Prototype Exchange), a novel FL framework that supports indirect cross-client representation sharing while maintaining data privacy. Each client constructs class-wise prototypes using Gaussian Mixture Models (GMMs), perturbs them with Gaussian noise, and transmits only these compact summaries to the server. The aggregated prototypes are then distributed back to clients and integrated into local training, supported by SMOTE-based augmentation to enhance representation of minority malware classes. Rather than relying solely on parameter averaging, our prototype-driven mechanism enables clients to enrich their local models with complementary structural patterns observed across the federation-without exchanging raw data or gradients. This targeted strategy reduces the adverse impact of statistical heterogeneity with minimal communication overhead. We evaluate FedP3E on the N-BaIoT dataset under realistic cross-silo scenarios with varying degrees of data imbalance. Results demonstrate that FedP3E consistently surpasses FedAvg and FedProx in malware detection performance, achieving accuracy ranging from 95.11% in severe non-IID conditions to 99.57% under light heterogeneity.

Index Terms—Federated Learning, IoT Malware Detection, Cross-Silo, Non-IID, Data Imbalance, Prototype Learning, Gaussian Mixture Models

I. INTRODUCTION

THE rapid advancement of 5G and the anticipated deployment of Beyond 5G (B5G) technologies are driving the proliferation of IoT devices at an unprecedented scale [1]. This hyper-connectivity is enabling new capabilities across healthcare, transportation, industry, and military sectors—collectively referred to as the Extended IoT (XIoT) [2]. However, the expansion of these networks has also introduced a significantly larger attack surface, accelerating both the frequency and severity of cyber threats targeting IoT ecosystems.

As XIoT domains continue to evolve, so do the associated security risks. Real-world incidents have underscored the scale and impact of these vulnerabilities. For instance, large-scale botnets such as Mirai and Gafgyt have been responsible for massive service disruptions and financial damage, with global losses estimated in the billions [3]. Additionally, the 2018 ransomware attack on the Taiwan Semiconductor Manufacturing Company—the world's largest chipmaker—exposed the fragility of Industrial IoT (IIoT) systems when operating under insufficiently secured infrastructures [4].

1

The diverse and heterogeneous nature of XIoT environments makes them especially difficult to protect. IoT devices monitor a wide range of data sources, including sensor readings, system logs, and control commands, often in real time. These data streams may contain highly sensitive or private information. Machine Learning (ML) and Deep Learning (DL) techniques have shown promise in detecting malicious activity across such data sources. However, conventional ML/DL approaches rely on centralized data aggregation, requiring raw data to be transmitted from edge devices to a central server [5]. This raises serious privacy concerns and regulatory challenges, especially in critical domains [6] like healthcare and smart infrastructure.

Federated Learning (FL) has emerged as a viable alternative to centralized training by enabling decentralized learning directly at the data source. In FL, devices collaboratively train a shared global model without exchanging raw data, thereby preserving user privacy and reducing communication costs. FL is commonly deployed in two paradigms: cross-device [7], where numerous clients (e.g., smartphones) contribute intermittently, and cross-silo [8], where a limited number of reliable organizations (e.g., hospitals, smart factories) participate in regular training rounds. Cross-silo FL is particularly well-suited to IoT malware detection, where devices are often grouped into domains with stable and trusted infrastructures.

Despite its advantages, FL faces several key challenges. A primary issue is the presence of non-independent and identically distributed (non-IID) data across clients. In realworld deployments, the distribution of malware types may vary significantly from one silo to another, leading to local models that converge poorly and global models that underperform on minority classes [9]. This issue is exacerbated when class imbalance is present—some malware families may be underrepresented or entirely missing on specific clients.

This work is driven by two pressing challenges in federated IoT malware detection: (1) enabling effective knowledge sharing among clients without exposing sensitive local data or requiring overlapping class distributions, and (2) improving generalization for rare or underrepresented malware families, which are frequently neglected by standard FL methods. Existing approaches like FedAvg and FedProx fall short in these scenarios, particularly under non-IID disjoint data conditions. To overcome these limitations, we introduce FedP3E—a novel prototype-based FL framework that transmits class-specific statistical summaries instead of raw data or model gradients. This design preserves data privacy, facilitates learning across heterogeneous silos, and enhances detection performance for minority malware classes, all while maintaining a low communication overhead.

The main contributions of this work are summarized as follows:

- Prototype-Based Representation Sharing: Instead of exchanging raw data or gradients, each client summarizes its local class-wise distribution using GMM-based proto-types. These representations are perturbed with Gaussian noise to preserve privacy while enabling implicit knowl-edge sharing across clients.
- **Disjoint-Aware Data Augmentation:** We integrate server-aggregated prototypes into local training and apply SMOTE-based oversampling to improve class balance. This addresses the limitations of disjoint and imbalanced data distributions commonly encountered in real-world federated IoT settings.
- Adaptive Communication Mechanism: Prototype exchange is triggered only when the global model's accuracy falls below a predefined threshold, ensuring minimal communication overhead while preserving the benefits of collaborative learning.
- Extensive Evaluation on N-BaIoT: We evaluate FedP3E on the N-BaIoT dataset under both IID and varying degrees of non-IID data imbalance, including disjoint class distributions. The results demonstrate consistent performance improvements over FedAvg and FedProx across heterogeneous federated scenarios.

The rest of this paper is organized as follows. Section II reviews state-of-the-art research on IoT malware detection using FL, including approaches for handling heterogeneous data and commonly used XIoT malware datasets. Section III introduces key terminologies relevant to the FL domain. Section IV details the proposed FedP3E framework, including its architecture and underlying techniques. Section V presents and analyzes the results of our experimental evaluation. Finally, Section VI concludes the paper and outlines potential directions for future research.

II. RELATED WORK

This section presents a comprehensive review of the current state-of-the-art in IoT malware detection using FL, examine advanced strategies designed to address data heterogeneity in FL environments, and summarize the most widely used public datasets that model cyberattacks across various XIoT domains.

A. IoT malware detection using Federated Learning

FL has recently gained traction as a privacy-preserving paradigm for IoT malware detection. A growing body of research has proposed diverse FL-based frameworks targeting various aspects of malware classification, including robustness, efficiency, class imbalance, and real-world deployment challenges.

Several studies have explored IoT malware detection using FL with real-world datasets. One framework leveraged both

supervised and unsupervised FL models-specifically a multilayer perceptron and autoencoder-trained on the N-BaIoT dataset to detect malware on both seen and unseen IoT devices. This approach demonstrated that federated models could achieve performance comparable to centralized models while maintaining data privacy. The study also evaluated aggregation robustness under adversarial conditions, highlighting vulnerabilities in standard FedAvg and the need for secure aggregation mechanisms [1]. Similarly, another study addressed the increasing prevalence of botnet attacks in IoT networks by evaluating the effectiveness of FL in detecting IoT malware traffic while preserving user privacy. Using the N-BaIoT dataset, the authors compared FL-based CNN, LSTM, and GRU models against a centralized baseline. The results showed that FL models, particularly CNN, achieved strong performance in identifying abnormal traffic, further stressing the value of FL in privacy-preserving IoT malware detection [10].

In the context of the Industrial Internet of Things (IIoT), FL has also been applied to mitigate privacy concerns and address the heterogeneity across industrial environments. A recent study proposed a federated botnet detection method where multiple industrial enterprises collaboratively train models without sharing raw data. The approach demonstrated high adaptability to local settings and robustness against poisoning attacks [11]. In another effort to enhance scalability and detection performance in FL-based IoT security systems, a study introduced a distributed optimization framework using the Siberian Tiger Optimization (STO) algorithm to fine-tune hyperparameters of a CNN model at the central server. These optimized parameters were then distributed to clients for local training [12].

To improve model robustness and label efficiency, FedMalDE introduced a semi-supervised FL framework employing knowledge transfer techniques and a subgraph aggregated capsule network to detect IoT malware [13]. Similarly, FEDroid targeted Android malware detection using a residual neural network combined with a genetic evolution strategy to simulate and detect malware variants, achieving superior performance across multiple Android datasets [14].

In another line of work, FL was integrated with Markov chains and associative rule learning to classify IoT malware across imbalanced and non-IID datasets. This hybrid approach achieved near-perfect accuracy (99%) while maintaining runtime performance comparable to centralized methods [15]. Complementing this, FED-MAL transformed malware binaries into image representations and used a compact CNN (AM-NET) with adversarial training to enhance generalizability on edge devices [16].

One framework focused on minimizing latency and model heterogeneity in FL for networked IoT systems, such as those using Raspberry Pi devices. The authors proposed a cloud-unification method to harmonize on-device models, achieving performance gains between 7% and 13%, with only minor increases in training time [17].

The SIM-FED model combined FL and a lightweight 1D CNN to deliver a robust and privacy-preserving malware detection framework. Evaluated on the IoT-23 dataset, SIM-FED achieved a 99.52% accuracy and showed resilience to whitebox and black-box adversarial attacks, while also reducing computational overhead [18].

Another study focused on ransomware detection through decentralized training across multiple clients, demonstrating strong results across all performance metrics. The integration of preprocessing, feature engineering, and collaborative learning led to a scalable and privacy-preserving detection solution [19].

A CNN-based FL model was also proposed to handle imbalanced datasets and intermittent client participation. In this approach, malware binaries were converted into color images to extract visual features, and data augmentation techniques were applied to balance training samples across clients. The system demonstrated strong results in handling both intermittent connectivity and class imbalance [20].

Graph-based FL frameworks have also emerged, including Fed-MalGAT, which outperformed its GCN-based counterpart by using multi-head attention for robust classification across federated rounds. Though it introduced additional computational cost, Fed-MalGAT consistently achieved high accuracy, precision, and F1 scores across multiple evaluations [6].

Recent work has also emphasized control-flow-based static analysis. One study introduced an FL framework for IoT binary classification using CFG-derived features under IID and non-IID scenarios. The models—FL-CNN and FL-DNN—demonstrated strong performance, with FL-CNN achieving 95.27% accuracy in the IID setting. The study further applied threshold tuning and class weighting to improve results under class imbalance [9].

Collectively, these works underscore the growing effectiveness of FL in detecting IoT malware under realistic and constrained conditions.

B. Heterogeneous data processing

Several advanced strategies have been proposed to address the challenges posed by non-IID data in FL. In [21], a novel algorithm named CSFedAvg was introduced to mitigate accuracy degradation caused by non-IID distributions. This method leverages weight divergence to estimate the degree of data heterogeneity at each client and selects those with more IID-like distributions for more frequent participation, thereby improving global model performance. In [22], a resourceefficient FL framework based on auction theory was proposed to minimize training costs while addressing non-IID effects. The approach jointly optimizes model utility, computational overhead, and data generation expenses, demonstrating that incorporating even a small fraction (less than 1%) of shared IID data significantly improves training efficiency and stakeholder profitability.

The FedSLD approach presented in [23] adjusts the contribution of each local data sample to the training objective based on the client's label distribution. By addressing label imbalance during optimization, the method improves training stability and convergence in heterogeneous data settings. In [24], FedNSE quantifies label distribution skewness and entropy to assess the non-IID characteristics of data across clients. Based on this assessment, it selects an optimal subset of clients for training. Results indicate a reduction of at least 10% in training loss and improved convergence speed compared to baseline methods.

To address non-IID data from a clustering perspective, [25] introduced K-FL, a Kalman filter-based FL method that groups clients with similar data distributions to produce low-variance, cluster-specific models. It operates without prior knowledge or initialization settings and achieves faster training and higher accuracy than FedAvg on MNIST, FMNIST, and CIFAR-10 datasets. Lastly, [26] proposed clustered Federated Multitask Learning (FMTL), which integrates model clustering and multitask learning within a dual-server privacy-preserving architecture. Secure two-party computation protocols are used to ensure data privacy while enhancing communication efficiency and overall model performance in non-IID environments.

C. Public datasets in XIoT malware detection

A wide array of public datasets has enabled significant progress in malware detection across the Extended Internet of Things (XIoT) domains, including IoT, Industrial Internet of Things (IIoT), Internet of Medical Things (IoMT), and Internet of Vehicles (IoV). These datasets vary in attack types, sources, and structural characteristics, supporting diverse detection approaches. Table I illustrates a summary of public datasets commonly used in XIoT-based malware detection, along with the federated splitting criterion.

To investigate Telnet-based attacks targeting IoT devices, IoTPoT [27] introduces a honeypot-driven sandbox that captures malware activity across architectures such as ARM, MIPS, and PPC. The dataset highlights the rapid evolution of DDoS malware families and their cross-platform propagation strategies. Bot-IoT [28], created within a cyber range, simulates realistic network traffic comprising both normal and malicious flows. Covering a broad spectrum of threats-DDoS, keylogging, scanning, and exfiltration-it provides millions of labeled flows for evaluating anomaly-based intrusion detection models. Focusing on protocol-specific traffic from both benign and infected IoT devices, IoT-23 [29] captures 23 real-world scenarios in pcap format. Its curated mix of malware and clean device captures supports studies on behavioral analysis and malware fingerprinting. With a static analysis approach tailored for Android, Drebin [30] is a well-known Android malware dataset introduced alongside a lightweight detection method designed to operate directly on smartphones. Given the rising volume and diversity of malicious Android applications, Drebin addresses the limitations of traditional defenses by employing broad static analysis. It extracts features from various sources-such as permissions, API calls, and network addresses-and embeds them in a joint vector space to capture malware-specific patterns. The dataset contains 123,453 Android applications and 5,560 labeled malware samples. Originating from a Microsoft-hosted classification challenge, BIG 2015 [31] features a massive collection of binaries representing nine malware families. Each sample includes both raw hexadecimal content and disassembled metadata, useful for signature- and behavior-based classification. Malimg [32] takes a visual approach by converting binary files

TABLE I Summary of Public Datasets Commonly Used in XIoT Malware Detection

Dataset	XIoT Domain	Federated Splitting Criterion
IoT-23 [37]	IoT	PCAP
N-BaIoT [36]	IoT	IoT device
IoTPoT [27]	IoT	Botnet sample / traffic stream
Bot-IoT [38]	IIoT	Attack type
BIG 2015 [31]	IoT/IIoT/IoMT	Family class
WUSTL-EHMS-2020 [35]	IoMT	Not applicable
NSL-KDD [34]	IoMT	Traffic category / attack type
Drebin [30]	IoT/IIoT/IoV	Android malware family
Malgenome [33]	IoT	Malware type
Malimg [32]	IIoT/IoV	Malware family

into grayscale images. With over 9,000 samples across 25 families, it allows malware to be classified based on visual texture patterns without requiring disassembly. The Malgenome dataset [33] systematizes over 1,200 Android malware samples collected over a year. It provides deep insights into mobile malware evolution, installation methods, and evasion tactics. Improving upon KDD'99, NSL-KDD [34] removes redundancy and balances class distribution. It remains a foundational benchmark for intrusion detection systems with well-defined categories like DoS, U2R, R2L, and probing. Designed for medical environments, WUSTL-EHMS-2020 [35] integrates biometric sensor data with network flow metrics. It captures spoofing and data injection attacks in a real-time healthcare testbed, reflecting IoMT-specific threats. Finally, N-BaIoT [36] offers traffic data from nine IoT devices infected with Mirai and BASHLITE. By using deep autoencoders to model benign behavior, the dataset facilitates precise detection of devicelevel anomalies in dynamic enterprise settings.

III. PRELIMINARIES

This section provides an overview of key terminologies commonly used in the FL domain, including FL deployment settings, data distribution strategies, and baseline aggregation algorithms.

A. Federated Learning deployment settings

1) Cross-Device: Cross-device FL refers to a setting in which numerous heterogeneous edge devices—such as smart-phones, sensors, and embedded systems—collaboratively train a shared model while keeping their data locally. This setting introduces several system-level challenges that must be addressed to ensure effective deployment [7]:

- Usability and Efficiency: Participating devices often vary significantly in hardware (e.g., x86, ARM) and software environments. These devices typically have limited computational power, storage capacity, and communication bandwidth, necessitating lightweight and efficient training, inference, and model management procedures.
- Scalability and Robustness: The system must support a vast number of devices and remain robust in the face of unreliable participants, including those that may frequently disconnect or respond slowly. As more resources are added, the system should scale efficiently to enhance training speed and model quality.

• Flexibility and Extensibility: Given the diversity of software stacks and APIs across devices, FL frameworks must allow for flexible algorithm customization and extension to optimize convergence and model performance across various application scenarios.

2) Cross-Silo: Cross-silo FL refers to collaborative model training among a small number of reliable and resourcerich organizations-such as hospitals, banks, and research institutions-where each participant holds large volumes of private data and remains involved throughout the training process [39]. Unlike cross-device FL, cross-silo FL faces unique challenges stemming from the need to balance high model performance, strict privacy requirements, and longterm cooperation between strategically motivated clients. A key technical challenge is statistical heterogeneity, where non-IID data distributions across clients degrade global model performance; this is addressed through data moderation, client clustering, and personalization techniques, which are more privacy-preserving. While system heterogeneity is less concerning due to clients' robust infrastructure, optimizing communication and computation through model compression and selective client participation remains important. Privacy and security are paramount, with defenses including differential privacy, homomorphic encryption, and secure multi-party computation-all offering tradeoffs between protection strength and computational cost. Moreover, cooperation and incentive mechanisms such as data valuation, profit allocation, and fairness-aware strategies are essential for sustaining client participation [39].

B. Federated Learning data distribution strategies

1) IID data in FL: In FL, data is considered independent and identically distributed (IID) when each client's local dataset follows the same underlying probability distribution and consists of statistically similar and independent samples. This assumption simplifies the collaborative training process by ensuring that all clients contribute uniformly representative data. Under IID conditions, local model updates are more aligned, reducing inter-client variability and enabling faster and more stable convergence of the global model. Consequently, performance discrepancies between local and global models are minimal, limiting the need for additional techniques such as personalization. Many foundational FL algorithms, such as FedAvg, were initially designed under this idealized setting, although real-world applications often deviate from these assumptions. The IID scenario is typically used as a performance baseline when evaluating FL algorithms and protocols in both simulations and empirical studies. It offers a controlled environment to understand the core behavior of FL frameworks before addressing more realistic data challenges such as statistical heterogeneity and privacy risks [40].

2) Non-IID data in FL: In practice, FL often operates under non-IID conditions, where data distributions vary significantly across clients. This heterogeneity stems from differences in user behavior, application usage, geographic location, sensor configurations, data collection frequency, and demographic characteristics, leading to inconsistencies in data quantity, class representation, and feature distributions [40]. These disparities introduce critical challenges such as increased communication costs, slower model convergence, reduced generalization performance, and heightened risks of privacy leakage. Non-uniform data distributions across clients require more frequent communication with the server to reach model consensus, resulting in higher latency and less efficient training. Class imbalance across clients can cause local models to underperform on underrepresented categories, which negatively affects the global model. Moreover, conflicting local gradients can hinder convergence, and feature-based discrepancies may reveal sensitive client information during aggregation [41].

C. Baseline optimization and aggregation strategies in Federated Learning

FedAvg (Federated Averaging) [42] and FedProx (Federated Proximal) [43] are foundational optimization strategies in FL, designed to train models across decentralized data sources. FedAvg operates by having each client perform a fixed number of local gradient descent steps on their private data. Once completed, clients send their updated model parameters to a central server, which averages these updates—weighted by the relative size of each client's dataset—to form a new global model. This method is simple and communication-efficient, but it does not explicitly restrict how far local updates can drift from the global model, which can become problematic in the presence of non-IID data [44].

In contrast, FedProx modifies the local objective function at each client by adding a proximal term that penalizes large deviations from the current global model. This means each client not only minimizes its local empirical loss but also includes a regularization term that discourages straying too far from the global model parameters received at the beginning of the round. As a result, FedProx helps stabilize the training process when client data distributions are different. While both algorithms use a similar aggregation approach-computing a weighted average of the client models-the key distinction lies in how local models are updated: FedAvg relies purely on local descent, while FedProx constrains the updates to remain close to the shared model. This difference makes FedProx particularly effective in handling heterogeneous data environments where client updates may diverge significantly [44].

IV. METHODOLOGICAL FRAMEWORK

This section presents the complete pipeline and techniques employed in the proposed FedP3E framework. The procedural steps are outlined in Algorithm 1.

A. Federated Learning setup

We employ a FL architecture comprising a central server and three client devices. Each client is a powerful and reliable machine capable of collecting traffic from IoT devices within the same network, for example, operating as a B5G base station—an arrangement that aligns with the cross-silo FL paradigm [1]. Each client (silo) acquires data from three



Fig. 1. Cross-silo federated learning architecture with three client devices collaborating under a central server.

distinct IoT devices, performs preprocessing, and conducts local model training, as illustrated in Fig. 1. The training process spans 20 communication rounds, with each client training locally for 15 epochs per round and transmitting updated model parameters to the server for aggregation.

At the beginning of each round, the server distributes the global model to all clients. If the average accuracy of the global model at round N falls below a predefined threshold (which may vary depending on the specific objectives of the application), clients are prompted to generate one-time, class-wise feature prototypes using Gaussian Mixture Models (GMMs). These prototypes, which capture intra-class variability, are transmitted to the server, aggregated, and redistributed to assist clients with missing or underrepresented classes.

To evaluate the proposed framework, we implement two widely adopted FL baselines: FedAvg and FedProx. FedAvg aggregates client updates through weighted averaging, while FedProx incorporates a proximal term into the local objective to mitigate update divergence in non-IID data settings.

B. Dataset description

To evaluate the proposed FedP3E framework, we utilize the N-BaIoT [36] dataset, which comprises network traffic captured from nine distinct IoT devices under both benign and malicious conditions, as summarized in Table II. Malicious activity was induced by infecting the devices with two malware families: Mirai and BASHLITE (also known as Gafgyt). Each data instance corresponds to a network packet recorded using Wireshark and is represented by 115 numerical features that reflect statistical properties—such as packet size, count, and jitter—computed over various time windows (ranging from 100 milliseconds to 1 minute).

The dataset is structured such that each device maintains a separate log, which facilitates natural partitioning for FL experiments. All devices include both benign and Gafgytinfected traffic, while seven out of the nine devices contain Mirai-related samples. Specifically, the Ennio doorbell and the Samsung SNH 1011 N webcam do not exhibit Mirai infections. Table III details the distribution of Gafgyt and Mirai variants across all IoT devices.

Why N-BaIoT ?: N-BaIoT dataset is particularly suited to our study for three reasons:

- 1) Cross-silo mapping The per-device log structure naturally maps to a cross-silo FL setting in which each silo aggregates traffic from one or more IoT devices managed by the same edge node.
- 2) Multi-class challenge The coexistence of benign traffic and multiple malware families yields a challenging, realistic multi-class detection task.
- 3) Configurable non-IID The dataset structure allows for simulating varying degrees of non-IID conditions, including: label distribution skew, where clients observe different class proportions; quantity skew, where the number of samples per client varies significantly; and disjoint distribution, where clients possess mutually exclusive sets of class labels.

This flexibility is essential for systematically validating the robustness of FedP3E under varying degrees of data heterogeneity.

C. Client data characteristics and data distribution

To emulate realistic cross-silo FL deployments, we consider two primary data distribution strategies across clients: IID and non-IID scenarios.

In the IID setting, the dataset is partitioned evenly among three clients based on device types. Specifically, all traffic data from IoT devices 1, 2, and 3 are assigned to Client 1; devices 4, 5, and 6 to Client 2; and the remaining devices (7, 8, and 9) to Client 3. This setup ensures that each client receives a balanced mix of benign samples, Gafgyt variants, and Mirai variants, thereby maintaining uniform class representation across the federation, as illustrated in Table IV.

In contrast, the non-IID distribution is explored through three escalating levels of statistical heterogeneity:

- Case 1: Light non-IID Certain attack variants are more prevalent in one client while sparsely present or entirely absent in others. For instance, the Gafgyt variant combo is abundant in Client 1, lightly represented in Client 3, and completely absent in Client 2, as shown in Table V.
- Case 2: Moderate non-IID Some attack types are entirely missing from one or more clients. For example, the Mirai variant *udp plain* appears exclusively in Client 2 and is absent from both Clients 1 and 3, as illustrated in Table VI.
- Case 3: Severe non-IID This most challenging configuration introduces extreme class imbalance where each client is assigned data from only one category. Specifically, Client 1 contains only benign samples, Client 2 hosts only Gafgyt samples, and Client 3 is limited to Mirai variants. This severe partitioning simulates a highly

heterogeneous federated environment, as illustrated in Table VII, and poses significant challenges for model convergence and generalization due to the complete absence of overlapping classes across clients.

These non-IID scenarios introduce practical challenges commonly encountered in distributed IoT environments. Differences in device capabilities, usage patterns, and threat exposure often result in skewed or incomplete local data distributions. As such, our design reflects real-world heterogeneity, where local models may struggle to learn from limited or biased representations of the overall data distribution.

Algorithm 1 FedP3E: Federated Privacy-Preserving Prototype Exchange

Require: Federated clients $\{C_1, C_2, ..., C_K\}$, global rounds T, prototype exchange threshold τ , evaluation round r^*

Ensure: Trained global model \mathcal{M}_G

- 1: Initialize global model \mathcal{M}_G^0
- 2: for each round t = 1 to T do

3:

- Server sends \mathcal{M}_G^{t-1} to all clients **for** each client C_k in parallel **do** 4:
- Perform local training on C_k for E epochs using local 5: data \mathcal{D}_k
- Send updated model \mathcal{M}_{k}^{t} to the server 6:
- 7: end for
- Server aggregates $\{\mathcal{M}_k^t\}_{k=1}^K$ to update \mathcal{M}_G^t 8:
- 9: if $t = r^*$ and Accuracy $(\mathcal{M}_G^t) < \tau$ then
- for each client C_k in parallel **do** 10:
- for each class $c \in C_k$ do 11:
- 12:
- 13:
- Fit a GMM to local features \mathcal{X}_c Extract component means $\mu_j^{(c)}$ as prototypes Perturb each prototype: $\tilde{\mu}_j^{(c)} = \mu_j^{(c)} + \epsilon$, $\epsilon \sim$ 14: $\mathcal{N}(0, \sigma^2 I)$

```
Send \{\tilde{\mu}_{i}^{(c)}\} to server
```

```
17:
         end for
```

15:

16:

- Server aggregates all prototypes by class using Mini-18: Batch K-Means
- Server sends merged global prototypes $\{\hat{\mu}_{l}^{(c)}\}$ to all 19: clients
- for each client C_k in parallel **do** 20:
- Apply SMOTE to $\{\hat{\mu}_l^{(c)}\}$ to generate synthetic 21: samples
- Add synthetic data to local dataset \mathcal{D}_k 22:
- 23: end for
- end if 24:
- 25: end for
- 26: **return** Final global model \mathcal{M}_G^T

D. Prototype generation via GMM

To realise privacy-preserving knowledge sharing while mitigating class imbalance, every client succinctly encodes its local data through *class-wise prototypes*. Instead of transmitting raw samples or model gradients, each client fits a GMM to the feature vectors of every local class and shares only the perturbed component means.

TABLE II SUMMARY OF BENIGN AND ATTACK INSTANCES FOR EACH IOT DEVICE IN THE N-BAIOT DATASET, INCLUDING INFECTION STATUS BY GAFGYT AND MIRAI MALWARE [36]

IoT Device	Gafgyt	Mirai	Benign	Attacks
Danmini Doorbell	1	1	49,548	968,750
Ennio Doorbell	1	X	39,100	316,400
Ecobee Thermostat	1	1	13,113	822,763
Philips B120N/10 Baby Monitor	1	1	175,240	923,437
Provision PT-737E Security	1	1	62,154	766,106
Cam				
Provision PT-838 Security Cam	1	1	98,514	738,377
SimpleHome XCS7-1002-WHT	1	1	46,585	816,471
Cam				
SimpleHome XCS7-1003-WHT	1	1	19,528	831,298
Cam				
Samsung SNH-1011N Webcam	1	X	52,150	323,072

Local GMM fitting: Let C_k denote the set of classes present on client k and $X_c = \{x_i \in \mathbb{R}^d \mid y_i = c\}$ the feature matrix of class $c \in C_k$. The class density is modelled as

$$p(x \mid c) = \sum_{j=1}^{K_c} \pi_j^{(c)} \, \mathcal{N}\!\left(x; \, \boldsymbol{\mu}_j^{(c)}, \boldsymbol{\Sigma}_j^{(c)}\right), \tag{1}$$

where

- K_c is the (data-driven) number of mixture components,
- selected via the Bayesian Information Criterion (BIC); $\pi_j^{(c)} > 0$ is the weight of component j with $\sum_{j=1}^{K_c} \pi_j^{(c)} = 1$; $\mu_j^{(c)} \in \mathbb{R}^d$ and $\Sigma_j^{(c)} \in \mathbb{R}^{d \times d}$ are, respectively, its mean (prototype) and covariance;
- $\mathcal{N}(\cdot; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes the *d*-variate Gaussian probability density function.

Prototype extraction and perturbation: After training the GMM, the set $\{\mu_j^{(c)}\}_{j=1}^{K_c}$ serves as the class prototypes. To protect privacy, each mean is obfuscated by additive noise:

$$\tilde{\boldsymbol{\mu}}_{j}^{(c)} = \boldsymbol{\mu}_{j}^{(c)} + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \, \sigma^{2} \mathbf{I}_{d}), \quad (2)$$

where

- σ controls the noise amplitude (tuned empirically to balance privacy and utility);
- \mathbf{I}_d is the $d \times d$ identity matrix;
- $\tilde{\mu}_{i}^{(c)}$ is the privacy-preserving prototype that will be up-

Conditional exchange: The prototype-sharing routine is one-off and adaptive: the server monitors the global accuracy during the first five rounds and triggers an exchange only if the mean accuracy drops below a preset threshold (97% in our experiments). However, these hyper-parameters are configurable in real-world deployments depending on the dataset characteristics, application requirements, and the convergence behavior of the federated system.

Rationale: Sharing perturbed GMM means fulfils three goals simultaneously: (1) it keeps raw data local, (2) it conveys essential class statistics to the federation, and (3) it injects privacy noise that thwarts sample reconstruction. Consequently, each client gains statistical cues about under-represented or unseen classes, enabling more balanced updates when local data are skewed.

E. Prototype aggregation and redistribution

To enable scalable and privacy-aware knowledge transfer, we introduce a centralized prototype aggregation mechanism that consolidates noisy client-side class representations. Once clients transmit their class-wise prototypes during the designated exchange round, the server initiates a centralized aggregation process. For each class c observed across the federation, the server collects the corresponding set of perturbed prototype vectors $\{\tilde{\mu}_i^{(c)}\}$ from all contributing clients. These vectors are then clustered using the MiniBatch K-Means algorithm, which is computationally efficient and well-suited for processing large volumes of prototype data with minimal overhead. The goal is to derive a compact yet representative set of global prototypes $\{\hat{\mu}_{l}^{(c)}\}_{l=1}^{L_{c}}$ for each class, where L_{c} denotes the number of clusters selected heuristically based on the number of received prototypes and the dimensionality of the feature space.

This aggregated set of global prototypes captures a more comprehensive view of class-specific distributions across the federated system. In the subsequent communication round, the server redistributes these refined prototypes to all participating clients. By integrating these global class representations into their local training pipelines, clients obtain statistical cues for classes that may not be present in their local datasets. This enhances the learning process and promotes better generalization across heterogeneous data.

Importantly, this redistribution strategy enables clients with skewed or disjoint data to benefit from the broader class diversity of the federation without violating data privacy constraints, thereby contributing to a more robust and balanced global model.

F. Data augmentation with SMOTE

To further mitigate local class imbalance and enrich the training corpus, clients apply the Synthetic Minority Oversampling Technique (SMOTE) to the received global prototypes. Upon receiving the set $\{\hat{\mu}_l^{(c)}\}\$ for underrepresented class c, each client generates additional synthetic samples by interpolating between randomly selected prototype pairs in the feature space. The number of synthetic instances is controlled to achieve approximately a 10% increase in the training set size relative to the number of received prototypes.

Formally, let $\hat{\mu}_a^{(c)}, \hat{\mu}_b^{(c)} \in \mathbb{R}^d$ be two randomly chosen prototype vectors corresponding to class c, where d is the feature space dimension. A synthetic sample $x_{syn} \in \mathbb{R}^d$ is then generated as follows:

$$x_{\rm syn} = \hat{\mu}_a^{(c)} + \lambda \cdot (\hat{\mu}_b^{(c)} - \hat{\mu}_a^{(c)}), \quad \lambda \sim \mathcal{U}(0, 1), \tag{3}$$

where λ is sampled from a continuous uniform distribution $\mathcal{U}(0,1)$. This linear interpolation preserves the geometric coherence of the class distribution while introducing controlled variability, thereby enhancing the representation of minority classes during local training.

The combination of prototype redistribution and SMOTEbased augmentation equips each client with a synthesized and more balanced perspective of the data space. This approach $\begin{tabular}{lll} TABLE III \\ DISTRIBUTION OF GAFGYT AND MIRAI ATTACK VARIANTS ACROSS IOT DEVICES IN THE N-BAIOT DATASET \\ \end{tabular}$

IoT device	Benign	Gafgyt variants					Mirai variants				
IoT1 (Danmini Doorbell)	Benign	combo	junk	scan	tcp	udp	ack	scan	syn	udp	udp plain
IoT2 (Ecobee Thermostat)	Benign	combo	junk	scan	tcp	udp	ack	scan	syn	udp	udp plain
IoT3 (Ennio Doorbell)	Benign	combo	junk	scan	tcp	udp	X	X	X	X	X
IoT4 (Philips B120N/10 Baby Monitor)	Benign	combo	junk	scan	tcp	udp	ack	scan	syn	udp	udp plain
IoT5 (Provision PT-737E Security Cam)	Benign	combo	junk	scan	tcp	udp	ack	scan	syn	udp	udp plain
IoT6 (Provision PT-838 Security Cam)	Benign	combo	junk	scan	tcp	udp	ack	scan	syn	udp	udp plain
IoT7 (Samsung SNH-1011N Webcam)	Benign	combo	junk	scan	tcp	udp	X	X	X	X	X
IoT8 (SimpleHome XCS7-1002-WHT Cam)	Benign	combo	junk	scan	tcp	udp	ack	scan	syn	udp	udp plain
IoT9 (SimpleHome XCS7-1003-WHT Cam)	Benign	combo	junk	scan	tcp	udp	ack	scan	syn	udp	udp plain

TABLE IV

Scenario 1: IID distribution of IoT traffic among clients. Each client receives a balanced, representative subset of benign packets and every Gafgyt / Mirai variant, emulating an ideal federated learning environment. The numbers reported in the table denote the exact instance counts for benign traffic and each malware family

Client	ІоТ	Benign		Gafgyt variants					Mirai variants			
			Combo	Junk	Scan	Тср	Udp	Ack	Scan	Syn	Udp	Udp
												plain
	IoT1	√ 49,548	√59,718	✓29,068	✓29,849	√ 92,141	✔105,874	✓102,195	√ 107,685	✓122,573	√ 237,665	√ 81,982
Client1	IoT2	√ 13,113	✓53,012	✓30,312	✓27,494	√95,021	√ 104,791	✓113,285	√ 43,192	✓116,807	√ 151,481	√ 87,368
	IoT3	√ 39,100	√53,014	√ 29,797	✓28,120	√ 101,536	√ 103,933	x	X	X	X	×
	IoT4	✓175,240	√58,152	✓28,349	✓27,859	√92,581	√105,782	✓91,123	√103,621	✔118,128	✓217,034	√ 80,808
Client2	IoT5	√62,154	✔61,380	√30,898	✓29,297	√ 104,510	√ 104,510	✔60,554	√ 96,781	√65,746	√ 156,248	√56,681
	IoT6	√ 98,514	√57,530	√ 29,068	√28,397	√ 89,387	√ 104,658	√57,997	√ 97,096	√61,851	√ 158,608	√53,785
	IoT7	√52,150	√58,669	✓28,305	√ 27,698	√ 97,783	√ 110,617	X	X	X	X	X
Client3	IoT8	√ 46,585	√54,283	√ 28,579	✓27,825	√ 88,816	✓103,720	√ 111,480	√ 45,930	✓125,715	√ 151,879	√78,244
	IoT9	√ 19,528	√59,398	√ 27,413	√ 28,572	√ 98,075	√ 102,980	✔107,187	√ 43,674	√ 122,479	√ 157,084	√ 84,436

TABLE V

Scenario 2 – Case 1: Light non-IID distribution of attack variants across clients. This setting introduces mild statistical heterogeneity, where certain malware variants are unevenly distributed among clients. The numbers shown in the table indicate the number of instances per client for both benign samples and specific malware variants

Client	IoT	Benign		Gafgyt variants					Mirai variants			
			Combo	Junk	Scan	Тср	Udp	Ack	Scan	Syn	Udp	Udp
												plain
	IoT1	√ 49,548	✓59,718	X	X	√ 92,141	X	X	√ 107,685	X	√ 237,665	√ 81,982
Client1	IoT2	√ 13,113	✓53,012	X	X	√95,021	X	X	√43,192	√ 116,807	X	√ 87,368
	IoT3	√ 39,100	✓53,014	X	X	√ 101,536	X	X	X	X	X	X
	IoT4	√ 175,240	X	✓28,349	√ 27,859	X	✔105,782	X	√ 103,621	√ 118,128	√ 217,034	X
Client2	IoT5	√ 62,154	X	√ 30,898	✓29,297	X	X	X	X	√65,746	X	X
	IoT6	√ 98,514	X	√ 29,068	√ 28,397	X	X	×	X	✔61,851	X	X
	IoT7	✓52,150	√58,669	X	X	√ 97,783	X	X	X	X	X	X
Client3	IoT8	√ 46,585	X	X	X	X	√ 103,720	√ 111,480	X	X	X	√ 78,244
	IoT9	√ 19,528	X	✔27,413	X	X	√ 102,980	√ 107,187	X	X	√ 157,084	√ 84,436

indirectly enhances model robustness—particularly in the presence of highly skewed or disjoint local datasets—without ever transmitting raw data samples or client-specific features.

Together, these components form the backbone of the proposed FedP3E framework, offering a scalable and privacypreserving mechanism for indirect knowledge transfer across non-IID and imbalanced IoT malware datasets within crosssilo FL environments.

V. EXPERIMENTS AND ANALYSIS

A. Experimental setup

This subsection outlines the data–preprocessing pipeline, the hardware/software stack, and the FL configuration. The full set of training hyper-parameters is listed in Table VIII.

1) Preprocessing: Each client loads its local CSV partition, scales all features to [0, 1] with Min–Max normalisation,

and performs an 80/20 stratified split to create training and test subsets. No further feature engineering or dimensionality reduction is applied, resulting in an input dimension of 115 features.

2) Hardware and software environment: Experiments run on a Linux workstation equipped with two NVIDIA GeForce RTX 4090 GPUs (24 GB each) and CUDA 12.2. Implementation uses Python 3.11.5, TensorFlow 2.15 (Keras API), and the Flower FL framework v1.15.2.

3) FL Configuration: We follow a cross-silo FL setup consisting of three high-capacity clients and a central server. All clients participate in every round, training the model locally for 15 epochs per round across 20 communication rounds. After the initial five rounds, if the global model's average accuracy falls below 97%, a one-time prototype exchange is triggered at round 6. In this step, each client transmits classwise GMM centroids with added Gaussian noise to preserve

TABLE VI

Scenario 2 – Case 2: Moderate non-IID distribution of attack variants across clients. This setting introduces a higher level of statistical heterogeneity compared to Case 1. Certain Malware variants are entirely absent from one or more clients, creating noticeable class imbalance. The numbers presented in the table reflect the instance counts of benign and malware samples distributed across clients

Client	ІоТ	Benign		Gafgyt variants					Mirai variants			
			Combo	Junk	Scan	Тср	Udp	Ack	Scan	Syn	Udp	Udp
												plain
	IoT1	√ 49,548	✓59,718	X	X	X	✔105,874	✓102,195	X	✓122,573	X	X
Client1	IoT2	✔13,113	✓53,012	×	X	X	X	√ 113,285	X	√ 116,807	X	X
	IoT3	√ 39,100	✓53,014	X	X	X	X	X	X	X	X	X
	IoT4	✔175,240	X	✓28,349	X	√ 92,581	X	X	√103,621	√ 118,128	X	√ 80,808
Client2	IoT5	X	X	√ 30,898	X	√ 104,510	√ 104,510	X	√ 96,781	√65,746	X	√56,681
	IoT6	X	X	√ 29,068	X	√ 89,387	X	X	√ 97,096	✔61,851	X	√53,785
	IoT7	X	X	×	√ 27,698	X	X	X	X	X	X	X
Client3	IoT8	X	X	×	✓27,825	X	X	√ 111,480	X	X	√ 151,879	X
	IoT9	✓19,528	X	X	✓28,572	X	✓102,980	✓107,187	X	X	√ 157,084	X

TABLE VII

Scenario 2 – Case 3: Severe non-IID distribution of attack variants across clients. In this setting, each client is exposed to a disjoint subset of the class space: Client 1 contains only benign samples, Client 2 observes exclusively Gafgyt malware, and Client 3 is limited to Mirai malware. The numbers reported in the table indicate the instance counts for each class type (benign, Gafgyt, and Mirai) assigned to the respective clients

Client	IoT	Benign		Gafgyt variants					Mirai variants			
			Combo	Junk	Scan	Тср	Udp	Ack	Scan	Syn	Udp	Udp
												plain
	IoT1	√ 49,548	X	X	X	X	X	X	X	X	X	X
Client1	IoT2	√ 13,113	X	X	X	X	X	X	X	X	X	X
	IoT3	√ 39,100	X	X	X	X	X	X	×	X	X	X
	IoT4	X	✓58,152	✓28,349	√ 27,859	√92,581	√ 105,782	X	X	X	X	X
Client2	IoT5	X	✔61,380	√ 30,898	√ 29,297	√ 104,510	✔104,510	X	X	X	X	X
	IoT6	X	√57,530	√ 29,068	√ 28,397	√ 89,387	√ 104,658	X	×	X	X	X
	IoT7	X	X	X	X	X	X	X	X	X	X	X
Client3	IoT8	X	X	X	X	X	X	√ 111,480	√ 45,930	✓125,715	√ 151,879	√78,244
	IoT9	X	X	X	X	X	X	✔107,187	√ 43,674	√ 122,479	√ 157,084	√ 84,436

 TABLE VIII

 P3E Federated-Learning Hyper-Parameters

Component / Setting	Configuration
Neural Network Architecture	
Input layer	115 numeric features
Dense layer 1	128 units, ReLU, $L_2 = 0.001$
Batch normalization	After Dense 1
Dropout	p = 0.5 after Dense 1
Dense layer 2	64 units, ReLU, $L_2 = 0.001$
Batch normalization	After Dense 2
Output layer	3 units, Softmax
Local Training Settings	
Optimizer	Adam, learning rate = 0.0001
Loss function	Sparse categorical cross-entropy
Epochs per round	15
Batch size	32
Federated Learning Setup	
FL strategy	FedAvg (full participation)
Number of rounds	20
Prototype exchange	Triggered if mean accuracy (rounds 1-5
	< 0.97
Gaussian noise	Added to prototypes, $\sigma = 0.01$
Total trainable parameters	23,683

data privacy, which the server reclusters the received centroids via MiniBatch K-Means and broadcasts the resulting global prototypes. Clients then apply SMOTE to augment minority classes by 10 % (relative to the original training size) and continue training.

B. Baseline methods

To benchmark the performance of the proposed P3E framework, we conduct comparative experiments under both IID and non-IID data distributions using two widely adopted FL algorithms: FedAvg and FedProx. These baseline methods are configured with the same training-level hyperparameters as FedP3E, excluding the prototype exchange mechanism and the incorporation of Gaussian noise. For FedProx, we evaluate its performance under varying values of the proximal term μ , specifically $\mu \in \{0.1, 0.3, 1.0\}$, to assess its sensitivity to different regularization strengths.

C. Results and analysis

This subsection presents a comprehensive evaluation of the proposed FedP3E framework across various data distribution scenarios. We compare its performance with FedAvg and FedProx using standard classification metrics, including accuracy, precision, recall, F1-score, and communication efficiency. Table IX summarizes the end-of-training performance of all FL methods under both IID and non-IID conditions.

1) IID scenario: In the IID scenario, both FedAvg and FedP3E rapidly exceed 99.5% accuracy within the first 10 communication rounds, with FedP3E attaining the lowest final loss (0.0168) and a slightly smoother learning curve. In contrast, FedProx with $\mu = 0.1$ trails slightly and plateaus at 98.9% accuracy. As the regularization strength increases

TABLE IX
END-OF-TRAINING PERFORMANCE OF ALL FEDERATED LEARNING METHODS UNDER IID AND NON-IID DATA-DISTRIBUTION SCENARIOS

Method	Accuracy	Precision	Recall	F1-score	Loss	Training time (s)		
Scenario 1: IID								
FedP3E	0.9971	0.997	0.997	0.997	0.0168	17,346.7		
FedAvg	0.9974	0.997	0.997	0.997	0.0214	17,171.2		
FedProx ($\mu = 0.1$)	0.9890	0.989	0.989	0.989	0.0423	17,186.4		
FedProx ($\mu = 0.3$)	0.9059	0.906	0.906	0.906	0.4927	17,414.6		
FedProx ($\mu = 1.0$)	0.1267	0.127	0.127	0.127	1.1750	16,937		
Scenario 2 - Case 1: Light non-IID								
FedP3E	0.9957	0.996	0.995	0.996	0.0199	12,338.3		
FedAvg	0.9379	0.940	0.935	0.938	0.3870	7,553		
FedProx ($\mu = 0.1$)	0.9278	0.930	0.925	0.928	0.3486	7,599.7		
FedProx ($\mu = 0.3$)	0.9610	0.961	0.960	0.961	0.1443	7,493.2		
FedProx ($\mu = 1.0$)	0.3328	0.330	0.330	0.331	1.1667	7,571.5		
	Scenar	io 2 - Case	2: Mod	erate non-	IID			
FedP3E	0.9940	0.994	0.994	0.994	0.0283	12,040.9		
FedAvg	0.9196	0.920	0.918	0.919	0.3921	8,052.6		
FedProx ($\mu = 0.1$)	0.9247	0.926	0.923	0.925	0.3407	7,949.9		
FedProx ($\mu = 0.3$)	0.9402	0.941	0.940	0.940	0.2167	8,103.2		
FedProx ($\mu = 1.0$)	0.5911	0.590	0.590	0.590	1.1379	7,971.4		
	Scena	ario 2 - Cas	se 3: Sev	ere non-II	D			
FedP3E	0.9511	0.951	0.950	0.951	0.7251	8 663.3		
FedAvg	0.4939	0.494	0.494	0.494	1.3466	5 788		
FedProx ($\mu = 0.1$)	0.4939	0.494	0.494	0.494	1.3556	5 785.5		
FedProx ($\mu = 0.3$)	0.4939	0.494	0.494	0.494	1.3637	5791.1		
FedProx ($\mu = 1.0$)	0.0485	0.049	0.048	0.048	1.2224	5 707.7		

to $\mu = 0.3$, convergence further degrades, and the final accuracy drops to 90.6%. With $\mu = 1.0$, FedProx struggles to learn altogether. These trends underscore that heavy proximal regularization is unnecessary in balanced data environments. The detailed accuracy and loss trends are shown in Fig. 2 and 3, respectively.

Takeaway: In balanced IID settings, FedP3E operates comparably to FedAvg without activating its prototype-exchange mechanism, which remains dormant unless a performance drop is observed. This allows the framework to maintain high performance with minimal communication overhead. In contrast, FedProx exhibits slower convergence and reduced final accuracy, indicating that proximal regularization offers limited value in such balanced environments.

2) Light non-IID: In the light non-IID setting, label distribution skew is mild: while all clients observe samples from both malware families, the frequency of individual variants differs across them. This setup reflects realistic deployments, where certain IoT environments experience uneven exposure to specific attack types. As shown in Fig. 4, FedP3E experiences a slight initial drop in accuracy—due to the deliberate class imbalance—but once the average global accuracy at round 5 falls below the 97% threshold, the one-time prototype exchange is triggered. This immediately improves performance,



Fig. 2. Comparison of accuracy across 20 rounds under the IID data distribution scenario for FedP3E, FedAvg, and FedProx with varying μ values.

with accuracy rising from 93.3% (round 6) to 99.5% (round 7), and remaining above 99% thereafter.

Fig. 5 further illustrates that this adjustment also leads to a significant drop in cross-entropy loss, reinforcing the stabilizing effect of prototype redistribution. In contrast, FedAvg and FedProx (with $\mu = 0.1$ and 0.3) converge more slowly and plateau between 93–96% accuracy, while FedProx with strong regularization ($\mu = 1.0$) fails to learn, stalling at just 33.28% accuracy. By round 20, FedP3E achieves an F1-score



Fig. 3. Evolution of training loss across 20 rounds under the IID data distribution scenario for FedP3E, FedAvg, and FedProx with varying μ values.



Fig. 4. Comparison of accuracy across 20 rounds under the light non-IID data distribution scenario for FedP3E, FedAvg, and FedProx with varying μ values.

of 99.6% and a loss of 0.020—improving by approximately 3.5 percentage points in F1-score and 86% in loss relative to the next-best baseline (FedProx, $\mu = 0.3$). The performance gain is most pronounced on minority variants, with recall improving by about 6 percentage points compared to FedAvg, indicating that prototype redistribution effectively compensates for mild class imbalance.

Takeaway: Under light heterogeneity, a one-time, privacy-preserving prototype exchange is sufficient to close the statistical gap between clients, enabling near-optimal global performance without data exposure or heavy communication. These results illustrate the practicality of FedP3E in scenarios where mild non-IID label imbalance exists.

3) Moderate non-IID: In the moderate non-IID setting, several malware variants are entirely absent from one or more clients. This heavier statistical skew (Table VI) emulates scenarios where clients operate in isolation or lack exposure to specific threat categories—for example, *udp plain* (Mirai) appears only at Client 2, while *combo* (Gafgyt) is limited to Client 1.

FedP3E demonstrates strong resilience to this form of



Fig. 5. Evolution of training loss across 20 rounds under the light non-IID data distribution scenario for FedP3E, FedAvg, and FedProx with varying μ values.



Fig. 6. Comparison of accuracy across 20 rounds under the moderate non-IID data distribution scenario for FedP3E, FedAvg, and FedProx with varying μ values.

heterogeneity. As illustrated in Fig.6, it steadily improves throughout the training process, achieving an accuracy and F1-score of 99.4% by round 20, along with the lowest final loss (0.028) as shown in Fig. 7. These gains are supported by a single round of prototype redistribution, which provides indirect exposure to unseen classes and facilitates inter-client knowledge transfer.

Among baseline methods, FedProx with $\mu = 0.3$ delivers the closest performance, plateauing near 96.1% accuracy, while FedAvg struggles with fluctuating accuracy and higher loss values. FedProx with $\mu = 1.0$ fails to converge entirely, indicating that aggressive regularization may be counterproductive in class imbalanced environments.

Importantly, FedP3E yields notable improvements in minority-class recall—achieving gains of approximately 7 percentage points over FedAvg and around 5 points over FedProx ($\mu = 0.3$). This highlights the effectiveness of targeted prototype exchange in moderating distribution skew while preserving both communication efficiency and client data privacy.



Fig. 7. Evolution of training loss across 20 rounds under the moderate non-IID data distribution scenario for FedP3E, FedAvg, and FedProx with varying μ values.

Takeaway: Under moderate non-IID conditions, FedP3E demonstrates strong generalization by using a one-time prototype redistribution to ensure all clients benefit from class information missing in their local datasets. Its consistent performance and improved recall on minority variants outperform all baselines, including the best FedProx variant ($\mu = 0.3$), highlighting its effectiveness in handling statistical heterogeneity without compromising privacy or communication efficiency.

4) Severe non-IID: The severe non-IID scenario represents the most challenging distribution setting, where each client holds data from a mutually exclusive class subset, resulting in complete label disjointness across the federation. Specifically, Client 1 contains only benign samples, Client 2 processes Gafgyt-infected traffic, and Client 3 receives only Mirai instances. This setting emulates highly siloed deployments—such as separate enterprise departments, segmented sensor networks, or distinct security zones—where operational roles and device types lead to drastically different threat exposures and data profiles.

Under this setting, baseline FL methods struggle to generalize due to the absence of overlapping label spaces across clients. As shown in Fig. 8, FedAvg fails to learn meaningful patterns, with global accuracy plateauing around 50% throughout. Similarly, FedProx across all tested μ values shows poor convergence, as clients lack exposure to any class diversity and the aggregation process struggles to align incompatible local models.

In contrast, FedP3E exhibits remarkable resilience. Despite the lack of overlapping label spaces, its one-time exchange of class-wise prototypes allows each client to gain semantically enriched representations of unseen categories. Following this exchange, SMOTE is applied locally to synthesize a small number (10%) of new training samples for missing classes based on the received prototypes. This enhancement empowers each client to update its model with representative features of all classes, enabling the server to aggregate a more coherent and generalizable global model.

As illustrated in Fig. 9, FedP3E significantly reduces loss after the prototype phase and maintains stable convergence thereafter. While the approach incurs a modest increase in communication due to prototype sharing, the performance



Fig. 8. Comparison of accuracy across 20 rounds under the severe non-IID data distribution scenario for FedP3E, FedAvg, and FedProx with varying μ values.



Fig. 9. Evolution of training loss across 20 rounds under the severe non-IID data distribution scenario for FedP3E, FedAvg, and FedProx with varying μ values.

gain—in both accuracy and convergence—is substantial, making FedP3E a practical solution for real-world federated deployments characterized by severe non-IID.

Takeaway: FedP3E effectively addresses the limitations of baseline FL algorithms in severe non-IID settings by introducing cross-client semantic context through prototype sharing and local SMOTE-based augmentation. This strategy enables robust convergence even in completely disjoint class distributions, where client datasets are inherently isolated and unbalanced. In contrast, baseline methods such as FedAvg and FedProx fail to generalize under such conditions, as they lack mechanisms to compensate for the absence of shared label space across clients.

5) FedAvg vs. FedProx: Comparative analysis across μ variants: FedAvg operates by averaging client model updates without regularization, while FedProx introduces a proximal term governed by the parameter μ to constrain local updates from drifting too far from the global model. This design aims to stabilize training in non-IID scenarios by aligning local objectives more closely with the global model.

However, our results reveal significant limitations in Fed-Prox's performance across different levels of data heterogeneity. Even under balanced IID conditions-where clients share identical class distributions—FedProx with $\mu = 1.0$ performs poorly, reaching only 12% accuracy. This drastic underperformance results from overly restrictive regularization, which severely limits local model updates and prevents the model from learning even in well-aligned data settings. Such behavior suggests that strong proximal constraints can be counterproductive in scenarios where client data is already consistent and does not require additional stabilization. In the light non-IID scenario, FedProx with $\mu = 0.3$ achieves slightly better final accuracy than both FedAvg and FedProx with $\mu = 0.1$, while maintaining a comparable convergence rate. In the moderate non-IID setting, FedProx with $\mu = 0.1$ and 0.3 outperform FedAvg in terms of accuracy, although all three approaches experience instability across rounds. This is primarily due to the absence of specific variant-level on certain clients. Without a mechanism to compensate for these missing variants, the locally-trained models struggle with consistent predictions on under-represented categories. In the severe non-IID case, where each client has access to only a single class, all FedProx variants exhibit a dramatic collapse in performance. The proximal term, which is intended to keep local models aligned with the global model, ends up restricting their ability to learn from the limited data available. At the same time, it provides no mechanism for handling unseen classes, which are entirely absent from local datasets.

Takeaway: FedProx's effectiveness depends heavily on the data distribution and the choice of μ . Its performance drops significantly in both balanced and skewed settings when regularization is too aggressive. In federated environments with high class imbalance or disjoint label spaces, stronger strategies like FedP3E-which actively enhance each client's view of the global class distribution-are required to ensure stable training and generalization.

6) Communication cost: To evaluate the communication efficiency of FedP3E, we quantify the transmission overhead associated with its prototype exchange mechanism relative to standard FL protocols. Let the communication payloads be defined as:

$$\mathbf{W} \in \mathbb{R}^{d_w}, \qquad \mathbf{P}_i = \left\{ \boldsymbol{\mu}_{k,j} \right\}_{\substack{k=1,\dots,C\\j=1,\dots,m_k}}, \quad \boldsymbol{\mu}_{k,j} \in \mathbb{R}^{d_x},$$

where:

- $d_w = 23,683$ total number of trainable model parameters.
- C number of classes observed locally by client i,
- m_k number of GMM prototypes per class k,
- $d_x = 115$ dimensionality of the input feature space,
- |W| = d_w size of the model update in floats,
 |P_i| = ∑^C_{k=1} m_k ⋅ d_x total size of prototype vectors uploaded by client *i*.

Regular FL Communication: In each FL round, clients upload their local model and receive the global model:

Upload: \mathbf{W} , Download: $\mathbf{W}^{(global)}$ \Rightarrow Total: $2 \cdot |\mathbf{W}|$ floats.

Prototype Exchange Overhead: If the global model fails to reach a predefined accuracy threshold (97%) after five communication rounds, a one-time prototype exchange is initiated. Each client uploads class-wise perturbed GMM centroids \mathbf{P}_i and receives a global prototype set \mathbf{P} aggregated and reclustered by the server:

Upload:
$$\mathbf{P}_i$$
, Download: \mathbf{P} .

Upload Size: Assuming each client observes C = 3classes and generates $m_k = 3$ prototypes per class, the total prototype upload size becomes:

 $|\mathbf{P}_i| = C \cdot m_k \cdot d_x = 3 \cdot 3 \cdot 115 = 1035$ floats.

Relative to the full model, this represents an upload cost of:

$$\frac{|\mathbf{P}_i|}{|\mathbf{W}|} = \frac{1035}{23,683} \approx 4.37\%.$$

Download Size: Following aggregation, the server sends back $\widehat{\mathbf{P}}$ consisting of $m'_k = 4$ prototypes per class for $C_{\text{global}} =$ 3 global classes:

 $|\widehat{\mathbf{P}}| = C_{\text{global}} \cdot m'_k \cdot d_x = 3 \cdot 4 \cdot 115 = 1380 \text{ floats},$

yielding a relative download cost of:

$$\frac{|\mathbf{P}|}{|\mathbf{W}|} = \frac{1380}{23,683} \approx 5.83\%.$$

Total Prototype Communication .: The total cost of the one-time prototype exchange, accounting for both upload and download, is:

$$\frac{|\mathbf{P}_i| + |\mathbf{\hat{P}}|}{|\mathbf{W}|} = \frac{1035 + 1380}{23,683} \approx 9.58\%.$$

Takeaway: FedP3E introduces a one-time prototype exchange mechanism that incurs less than 10% of the communication volume of a full model round. This lightweight overhead enables statistically enriched updates under non-IID conditions while preserving the communication efficiency of standard FL protocols in all other rounds.

7) Training overhead: In addition to communication efficiency and model accuracy, we evaluate the training overhead of FedP3E across varying degrees of data heterogeneity. All methods were executed on identical hardware to ensure fair comparison, and Fig. 10 presents the total training time under IID, light, moderate, and severe non-IID scenarios.

Under the IID scenario, all approaches yield nearly identical training durations, indicating that FedP3E introduces no computational penalty when prototype exchange remains inactive. This confirms its ability to remain lightweight under balanced conditions.

However, as data heterogeneity increases, FedP3E incurs a higher training cost compared to FedAvg and FedProx. This



Fig. 10. Training time comparison across IID and non-IID scenarios.

is attributed to the additional steps of prototype generation, centralized aggregation, and SMOTE-based data augmentation triggered by non-IID conditions. While this overhead is non-negligible, it reflects a deliberate trade-off: enhanced generalization and robustness in challenging settings, achieved without any repeated exchange or raw data sharing.

Takeaway: FedP3E remains lightweight in IID settings and activates its additional components only when performance falls below a threshold. Although this introduces moderate computational overhead in skewed scenarios, the improved robustness justifies the cost in real-world deployments.

D. Comparison with existing federated learning methods on N-BaIoT

To further highlight the performance of FedP3E, Table X provides a comparative analysis against existing FL-based methods evaluated on the N-BaIoT dataset. Most prior studies focus on standard IID or mildly skewed non-IID scenarios and do not explicitly account for fully disjoint class distributions. As a result, their effectiveness diminishes as data heterogeneity increases. In contrast, FedP3E is explicitly designed to operate across the full spectrum of data distributions, including extreme cases where each client possesses samples from only one class. It achieves 99.71% accuracy in the IID setting, 99.57% under light non-IID, 99.40% in moderate non-IID, and maintains a strong 95.11% accuracy under severe non-IID conditions. This consistent performance across increasing degrees of heterogeneity underscores FedP3E's adaptability and robustness, making it a practical and scalable solution for federated IoT malware detection in highly diverse and siloed environments.

VI. CONCLUSION AND FUTURE WORK

This study proposed FedP3E, a prototype-based FL framework tailored to address the challenges of non-IID data distributions in IoT malware detection. Unlike conventional aggregation methods such as FedAvg and FedProx, FedP3E leverages class-wise prototype exchange to bridge the data

TABLE X Comparison of Proposed FedP3E with Existing Federated Learning-based Methods on N-BaIoT Dataset

Ref.	IID OR Non-IID?	Disjoint Class Considered?	Accuracy (%)
[1]	Non-IID	X	Supervised: 99.42–99.92 Unsupervised: ≈97.4
[11]	IID	X	99.63
[10]	Non-IID	X	90.00
[12]	Non-IID	X	89.90
FedP3E	Both	1	IID: 99.71
			Non-IID: 95.11–99.57

heterogeneity gap across clients without compromising data privacy.

Extensive experiments were conducted across four data distribution scenarios—IID, light, moderate, and severe non-IID. The results demonstrate that FedP3E consistently achieves faster convergence, higher accuracy, and improved robustness compared to existing baselines, particularly under severe non-IID conditions. Notably, while FedProx is designed for non-IID settings, our experiments revealed its limitations in more skewed scenarios, where it failed to recover performance.

The findings underscore the effectiveness and scalability of prototype-based communication in real-world cross-silo FL settings where data imbalance and heterogeneity are prevalent. FedP3E offers a lightweight and privacy-preserving mechanism that enhances collaborative learning among distributed IoT devices, contributing to more secure and adaptive malware detection.

Looking ahead, we plan to deploy FedP3E in real-world ondevice FL settings to assess its efficiency on lightweight IoT devices. This will enable us to evaluate the framework under more stringent constraints, including limited computational resources, constrained memory and communication bandwidth, and a vast number of dynamically selected heterogeneous devices with varying hardware capabilities. In parallel, we intend to explore more complex forms of data heterogeneity, particularly feature-space skew, where clients have access to different subsets of input features. This presents new challenges for collaborative learning, requiring novel strategies for model alignment and robust generalization across diverse feature representations.

ACKNOWLEDGMENTS

This should be a simple paragraph before the References to thank those individuals and institutions who have supported your work on this article.

REFERENCES

- V. Rey, P. M. S. Sánchez, A. H. Celdrán, and G. Bovet, "Federated learning for malware detection in IoT devices," *Computer Networks*, vol. 204, p. 108693, 2022.
- [2] R. Darwish, M. Abdelsalam, and S. Khorsandroo, "Deep learning based xiot malware analysis: A comprehensive survey, taxonomy, and research challenges," *Journal of Network and Computer Applications*, p. 104258, 2025.
- [3] DailyHostNews, "IoT malware surges by 400% in 2023 with US being the most targeted country – Zscaler," 2023, Accessed: 14 April 2024. [Online]. Available: https://www.linkedin.com/pulse/ iot-malware-surges-400-2023-us-being-most-targeted-country-m9m1f

- [4] D. Smith, S. Khorsandroo, and K. Roy, "Machine learning algorithms and frameworks in ransomware detection," *IEEE Access*, vol. 10, pp. 117 597–117 610, 2022.
- [5] X. Wu, Y. Zhang, M. Shi, P. Li, R. Li, and N. N. Xiong, "An adaptive federated learning scheme with differential privacy preserving," *Future Generation Computer Systems*, vol. 127, pp. 362–372, 2022.
- [6] M. Amjath, S. Henna, and U. Rathnayake, "Graph representation federated learning for malware detection in internet of health things," *Results* in Engineering, vol. 25, p. 103651, 2025.
- [7] D. Chen, D. Gao, Y. Xie, X. Pan, Z. Li, Y. Li, B. Ding, and J. Zhou, "Fsreal: Towards real-world cross-device federated learning," in *Proceed*ings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2023, pp. 3829–3841.
- [8] E. Lomurno, A. Archetti, L. Cazzella, S. Samele, L. Di Perna, and M. Matteucci, "Sgde: Secure generative data exchange for cross-silo federated learning," in *Proceedings of the 2022 5th International Conference on Artificial Intelligence and Pattern Recognition*, 2022, pp. 205– 214.
- [9] R. Darwish and K. Roy, "Comparative analysis of federated learning, deep learning, and traditional machine learning techniques for iot malware detection," in 2025 IEEE 4th International Conference on AI in Cybersecurity (ICAIC). IEEE, 2025, pp. 1–10.
- [10] P. H. Do, T. D. Le, V. Vishnevsky, A. Berezkin, and R. Kirichek, "A horizontal federated learning approach to iot malware traffic detection: An empirical evaluation with n-baiot dataset," in 2024 26th International Conference on Advanced Communications Technology (ICACT). IEEE, 2024, pp. 1494–1506.
- [11] H. Zhou and Z. Sheng, "A federated learning based botnet detection method for industrial internet of things," in *Proceedings of the 8th International Conference on Cyber Security and Information Engineering*, 2023, pp. 282–288.
- [12] B. B. Gupta, A. Gaurav, W. Alhalabi, V. Arya, E. Alharbi, and K. T. Chui, "Distributed optimization for iot attack detection using federated learning and siberian tiger optimizer," *ICT Express*, 2025.
- [13] X. Pei, X. Deng, S. Tian, L. Zhang, and K. Xue, "A knowledge transferbased semi-supervised federated learning for iot malware detection," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 3, pp. 2127–2143, 2022.
- [14] W. Fang, J. He, W. Li, X. Lan, Y. Chen, T. Li, J. Huang, and L. Zhang, "Comprehensive android malware detection based on federated learning architecture," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 3977–3990, 2023.
- [15] G. D'Angelo, E. Farsimadan, M. Ficco, F. Palmieri, and A. Robustelli, "Privacy-preserving malware detection in android-based iot devices through federated markov chains," *Future Generation Computer Systems*, vol. 148, pp. 93–105, 2023.
- [16] M. Abdel-Basset, H. Hawash, K. M. Sallam, I. Elgendi, K. Munasinghe, and A. Jamalipour, "Efficient and lightweight convolutional networks for iot malware detection: A federated learning approach," *IEEE Internet of Things Journal*, vol. 10, no. 8, pp. 7164–7173, 2022.
- [17] S. Shukla, S. Rafatirad, H. Homayoun, and S. M. P. Dinakarrao, "Federated learning with heterogeneous models for on-device malware detection in iot networks," in 2023 Design, Automation & Test in Europe Conference & Exhibition (DATE). IEEE, 2023, pp. 1–6.
- [18] M. Nobakht, R. Javidan, and A. Pourebrahimi, "Sim-fed: Secure iot malware detection model with federated learning," *Computers and Electrical Engineering*, vol. 116, p. 109139, 2024.
- [19] S. Koike, H. Tanaka, and M. Maeda, "Federated learning-based ransomware detection via indicators of compromise," 2024.
- [20] F. Ullah, G. Srivastava, S. Ullah, and L. Mostarda, "Privacy-preserving federated learning approach for distributed malware attacks with intermittent clients and image representation," *IEEE Transactions on Consumer Electronics*, vol. 70, no. 1, pp. 4585–4596, 2023.
- [21] W. Zhang, X. Wang, P. Zhou, W. Wu, and X. Zhang, "Client selection for federated learning with non-iid data in mobile edge computing," *IEEE Access*, vol. 9, pp. 24462–24474, 2021.
- [22] E. Seo, D. Niyato, and E. Elmroth, "Resource-efficient federated learning with non-iid data: An auction theoretic approach," *IEEE Internet of Things Journal*, vol. 9, no. 24, pp. 25 506–25 524, 2022.
- [23] J. Luo and S. Wu, "Fedsld: Federated learning with shared label distribution for medical image classification," in 2022 IEEE 19th International Symposium on Biomedical Imaging (ISBI). IEEE, 2022, pp. 1–5.
- [24] S. Bansal, M. Bansal, R. Verma, R. Shorey, and H. Saran, "Fednse: Optimal node selection for federated learning with non-iid data," in 2023 15th International Conference on COMmunication Systems & NETworkS (COMSNETS). IEEE, 2023, pp. 713–721.

- [25] H. Kim, B. Kim, Y. Kim, C. You, and H. Park, "K-fl: Kalman filterbased clustering federated learning method," *IEEE Access*, vol. 11, pp. 36 097–36 105, 2023.
- [26] J. Shu, T. Yang, X. Liao, F. Chen, Y. Xiao, K. Yang, and X. Jia, "Clustered federated multitask learning on non-iid data with enhanced privacy," *IEEE Internet of Things Journal*, vol. 10, no. 4, pp. 3453–3467, 2022.
- [27] Y. M. P. Pa, S. Suzuki, K. Yoshioka, T. Matsumoto, T. Kasama, and C. Rossow, "{IoTPOT}: analysing the rise of {IoT} compromises," in 9th USENIX Workshop on Offensive Technologies (WOOT 15), 2015.
- [28] U. of New South Wales (UNSW), "Bot-iot dataset unsw research," https://research.unsw.edu.au/projects/bot-iot-dataset, accessed: 03 Jan 2025.
- [29] S. Laboratory, "Iot-23 dataset: A labeled dataset for iot malware and benign traffic," https://www.stratosphereips.org/datasets-iot23, accessed: 03 Jan 2025.
- [30] D. Arp, M. Spreitzenbarth, M. Hubner, H. Gascon, K. Rieck, and C. Siemens, "Drebin: Effective and explainable detection of android malware in your pocket." in *Ndss*, vol. 14, 2014, pp. 23–26.
- [31] R. Ronen, M. Radu, C. Feuerstein, E. Yom-Tov, and M. Ahmadi, "Microsoft malware classification challenge," arXiv preprint arXiv:1802.10135, 2018.
- [32] L. Nataraj, S. Karthikeyan, G. Jacob, and B. S. Manjunath, "Malware images: visualization and automatic classification," in *Proceedings of the* 8th international symposium on visualization for cyber security, 2011, pp. 1–7.
- [33] Y. Zhou and X. Jiang, "Dissecting android malware: Characterization and evolution," in 2012 IEEE symposium on security and privacy. IEEE, 2012, pp. 95–109.
- [34] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the kdd cup 99 data set," in 2009 IEEE symposium on computational intelligence for security and defense applications. Ieee, 2009, pp. 1–6.
- [35] A. A. Hady, A. Ghubaish, T. Salman, D. Unal, and R. Jain, "Intrusion detection system for healthcare systems using medical and network data: A comparison study," *IEEE Access*, vol. 8, pp. 106 576–106 584, 2020.
- [36] Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, A. Shabtai, D. Breitenbacher, and Y. Elovici, "N-baiot—network-based detection of iot botnet attacks using deep autoencoders," *IEEE Pervasive Computing*, vol. 17, no. 3, pp. 12–22, 2018.
- [37] S. Garcia, A. Parmisano, and M. J. Erquiaga, "Iot-23: A labeled dataset with malicious and benign iot network traffic," (*No Title*), 2020.
- [38] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset," *Future Generation Computer Systems*, vol. 100, pp. 779–796, 2019.
- [39] C. Huang, J. Huang, and X. Liu, "Cross-silo federated learning: Challenges and opportunities," arXiv preprint arXiv:2206.12949, 2022.
- [40] Z. Lu, H. Pan, Y. Dai, X. Si, and Y. Zhang, "Federated learning with non-iid data: A survey," *IEEE Internet of Things Journal*, 2024.
- [41] V. Altomare, D. Thakur, A. Guzzo, and F. Piccialli, "Client specific dynamic aggregation for non-iid federated learning," in 2024 IEEE International Conference on Big Data (BigData). IEEE, 2024, pp. 7622–7631.
- [42] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273– 1282.
- [43] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proceedings of Machine learning and systems*, vol. 2, pp. 429–450, 2020.
- [44] L. Su, J. Xu, and P. Yang, "A non-parametric view of fedavg and fedprox: Beyond stationary points," *arXiv preprint arXiv:2106.15216*, 2021.