

A Lightweight Federated-Learning Intrusion Detection System for Agricultural IoT: Random-Forest-Guided Feature Selection with a Temporal Convolutional Classifier

Mohd Abdul Rahim Khan, Yasir Hashim Naif

Department of Electrical Engineering & Computer Science, College of Engineering, A'sharqiyah University, Ibra 400, Oman

Abstract—Internet of Things (IoT) devices used in precision agriculture, such as soil and climate sensors, irrigation controllers, livestock trackers, and autonomous machinery, form a large, heterogeneous, and often poorly secured attack surface, referred to here as Agricultural IoT (Ag-IoT). Centralized intrusion detection is difficult to apply in this setting because raw traffic often cannot leave the farm gateway for privacy, bandwidth, and regulatory reasons, and because edge devices have limited computing and memory budgets. This study proposes a lightweight federated intrusion detection framework for Ag-IoT that decouples a one-time, centralized Random Forest (RF) feature-selection stage from repeated federated training of a compact Temporal Convolutional Network (TCN). TCN is trained with Federated Averaging (FedAvg) across ten simulated clients whose label distributions follow a Dirichlet distribution ($\alpha = 0.5$), producing non-independent and identically distributed (non-IID) data. The RF reduces the UNSW-NB15 flow features to 20 high-importance attributes plus one RF attack-probability feature. On two hold-out protocols drawn from the official UNSW-NB15 training and testing partitions, the federated TCN reaches 95.84% accuracy (Matthews correlation coefficient, MCC = 0.904; area under the receiver operating characteristic curve, AUC = 0.988) and 97.72% accuracy (MCC = 0.954), respectively, with only 8,834 trainable parameters, a 0.034 MB model, and 0.034 ms per-sample inference latency, and it converges within three to five communication rounds. A per-client loss-spread diagnostic reveals an approximately 50-fold disparity between the best- and worst-served clients that global accuracy conceals. The study is limited to binary detection on one benchmark; same-split centralized and published baselines, alternative aggregators, and additional IoT datasets are identified as future work.

Keywords—Federated learning; intrusion detection system; agricultural internet of things; temporal convolutional network; random forest; feature selection; federated averaging; non-independent and identically distributed data; edge computing; lightweight deep learning

I. INTRODUCTION

Precision agriculture increasingly relies on networked sensing and actuation: soil-moisture and nutrient sensors, weather stations, variable-rate irrigation controllers, livestock wearables, drones, and autonomous tractors. These Agricultural IoT (Ag-IoT) devices are typically resource-constrained, deployed over wide geographic areas with

intermittent connectivity, and managed by operators with limited security expertise, making them attractive targets for botnets, reconnaissance, denial-of-service, and injection-style attacks [1], [2], [3], [4], [5]. Unlike enterprise networks, Ag-IoT deployments cannot always transmit raw packet captures to a central server for analysis: farm connectivity is frequently metered or unreliable, and consolidating raw traffic raises privacy and competitive-sensitivity concerns for growers [3], [6].

Federated Learning (FL) offers a natural fit for this setting: each farm or gateway trains a local model on its own traffic and shares only model updates with a coordinating server, which aggregates them into a global model [7], [8], [9]. However, two practical obstacles remain. First, Ag-IoT edge hardware (microcontrollers, low-power gateways, single-board computers) cannot host large deep models, so the classifier itself must be small in parameter count and inference latency [9], [10]. Second, traffic distributions differ sharply across farms and device types. A non-IID setting, which is known to slow convergence and destabilize naive federated averaging [11], [12], [10].

This study addresses both obstacles with a two-stage pipeline. A Random Forest (RF) [13] is first trained centrally on a labeled reference sample purely to rank feature importance, which reduces the engineered flow-feature space (packet-header statistics, inter-arrival times, flag counts, protocol indicators, and connection counters) to a compact top-k subset; the RF's own predicted attack probability is appended as one additional hybrid input. The reduced input is then used to train a compact Temporal Convolutional Network (TCN) [14] with Federated Averaging (FedAvg) [7] across simulated edge clients whose label distributions are drawn from a Dirichlet distribution, so that the reported results reflect statistically heterogeneous, non-independent and identically distributed (non-IID) conditions rather than an idealized uniform split [12].

Despite a growing body of work on federated intrusion detection [6], [15], [16], [17], [18], [10], three gaps remain for agricultural deployments. First, FELIDS [6], a representative agriculture-specific federated intrusion detection system, trains deep classifiers on the full feature space, and its evaluation emphasizes detection accuracy and data privacy rather than a quantified on-device footprint, so its suitability for low-power

farm gateways is not established. Second, most federated IoT detectors report a single global accuracy, which can hide clients that are poorly served by the shared model under non-IID data [8], [10]. Third, prior lightweight and optimization-driven IoT detectors, including our own [19], [4], [20], are trained centrally and therefore do not address the data-locality constraint of multi-farm deployments. The novelty of this work lies in combining 1) a one-time centralized feature-selection stage that never participates in federated communication, 2) a federated TCN of fewer than 9,000 parameters whose deployment and communication cost are measured explicitly, and 3) a per-client fairness diagnostic reported alongside detection metrics, all within an explicit Ag-IoT system and threat model.

The contributions of this work are:

- A decoupled feature-selection-then-federated-classification pipeline in which an RF trained once centrally selects 20 flow features (plus an RF attack-probability feature) for a compact TCN trained in a federated manner, so that the comparatively expensive feature-ranking model never adds to the per-round federated communication.
- An explicit Ag-IoT system and threat model that defines the participating entities, trust assumptions, in-scope and out-of-scope adversaries, and design goals, and that maps the attack categories of the evaluation benchmark to plausible farm-level scenarios.
- An empirical evaluation of FedAvg with the TCN on UNSW-NB15 under Dirichlet non-IID partitioning, using two hold-out protocols drawn from the official training and testing partitions, reported with accuracy, weighted precision, recall, F1, AUC, the Matthews correlation coefficient, per-class error rates, and a per-client loss-spread fairness diagnostic.
- A measured lightweight-deployment profile (8,834 trainable parameters, 0.034 MB, 0.034 ms per-sample inference, and approximately 0.7 MB of communication per round for ten clients) that quantifies the on-device and communication cost of the federated classifier.

The remainder of this study is organized as follows. Section II reviews related work in IoT intrusion detection and federated learning. Section III defines the system and threat model. Section IV describes the proposed methodology in detail. Section V describes the experimental setup. Section VI reports and discusses results. Section VII discusses limitations, and Section VIII concludes.

II. RELATED WORK

A. IoT and Network Intrusion Detection

Machine-learning-based network intrusion detection has been studied extensively on benchmark datasets such as UNSW-NB15 [21], which augments modern attack behaviors over older corpora such as KDD-99/NSL-KDD, and more recent IoT-specific datasets such as Edge-IIoTset [22] and CIC-IoT2023 [23], which capture traffic from realistic IoT and

industrial-IoT device populations under a wide range of attack families (DDoS, DoS, reconnaissance, brute force, spoofing, web-based, and Mirai-style botnet traffic). Classical tree ensembles such as Random Forest remain competitive baselines on these tabular flow-feature datasets and are additionally attractive for their built-in feature-importance ranking, which this work exploits purely as a feature-selection mechanism rather than as the final classifier. Closer to the specific application domain, metaheuristic-optimized detectors have recently been proposed for IoT traffic, including a self-adaptive IoT intrusion detection framework combining Addax optimization with echo state networks [19], a DenseNet-based botnet detector using residual dense blocks with cheetah/harmony-search-optimized (CE-HSOA) hyperparameter tuning [4], and a hybrid metaheuristic cluster-head optimization approach for improving IoT network performance metrics relevant to intrusion-detection deployment [20]. Related work on device- and application-level threat detection has also targeted Android malware directly, using deep-learning ensembles with cheetah-optimized feature selection [24] and image-based classification of decompiled Android application features [25], as well as obfuscation-attack detection via grayscale/RGB image classification of APK content [26] — all illustrating the broader pattern, adopted in this study, of pairing a feature-selection or optimization stage with a compact downstream classifier. Most of these studies, however, train and evaluate centrally on a single pooled dataset, so their reported accuracy does not reflect the data-locality and client-heterogeneity constraints considered here.

B. Sequence and Temporal Models for Traffic Classification

Because network flows are naturally ordered in time, sequence models — recurrent networks (LSTM, GRU) and, more recently, Temporal Convolutional Networks (TCNs) — have been applied to intrusion detection. TCNs [14] use causal, dilated convolutions to obtain a large receptive field with fewer parameters and better training parallelism than recurrent alternatives, which is attractive for resource-constrained deployment. Squeeze-and-excitation (SE) blocks [27], originally proposed for image classification, provide a lightweight channel-attention mechanism that this work adapts to re-weight temporal-convolutional feature channels. A closely related lightweight architecture — a multi-octave dilated TCN combined with explainable AI and meta-learning for digital financial fraud detection [28] — demonstrates that compact, dilated-convolutional temporal models of the kind used here generalize well beyond network intrusion detection to other high-stakes, resource-sensitive anomaly-detection tasks. Existing TCN-based detectors are, however, typically trained centrally, and their parameter budgets are seldom reported against the memory limits of edge gateways; this work evaluates a TCN of fewer than 9,000 parameters in a federated setting.

C. Federated Learning: Foundations and IDS Applications

Federated learning has matured into a broad research area spanning systems, optimization, and privacy. Comprehensive surveys catalog open problems across communication efficiency, systems heterogeneity, robustness to adversarial or unreliable clients, and privacy [8], and formalize the distinction

between horizontal, vertical, and transfer federated learning depending on how data and features are partitioned across participating organizations [29] — this study operates in the horizontal setting, where clients share the same feature schema (the RF-selected flow features) but hold disjoint samples. Within IoT specifically, federated learning has been surveyed as an enabling technology across a wide range of applications — data sharing, offloading and caching, localization, mobile crowdsensing, and network attack detection — under communication, device-heterogeneity, and resource constraints closely analogous to those motivating the lightweight design adopted here [9].

Federated Averaging (FedAvg) [7] established the basic client-local-training-then-server-averaging protocol used throughout this work. Subsequent work identified that naive averaging degrades under statistically heterogeneous (non-IID) client data; FedProx [11] addresses this by adding a proximal term that penalizes local models for drifting far from the global model, while server-side momentum methods such as FedAvgM [12] apply momentum to the aggregated update to stabilize convergence. The present study adopts FedAvg as the reference aggregator under a controlled Dirichlet non-IID partition, which is a standard way to interpolate between IID and highly skewed client distributions [12]; FedProx and FedAvgM are retained as planned extensions (Section IV-G). The general strategy of pairing federated training with a dedicated feature-selection stage has also been explored outside the network-security domain — for example, a federated epilepsy prediction and management system that combines a hybrid harmony-search and mutual-information (HSA-MI) feature-selection approach with federated model training [30] — lending independent support to the feature-selection-then-federation design pattern adopted in this study.

D. Class Imbalance and Privacy

Intrusion-detection datasets are inherently imbalanced, with benign traffic dominating most attack classes. SMOTE [31] is a widely used synthetic oversampling technique for tabular minority classes, and focal loss [32] down-weights easy (typically majority-class) examples during training to focus learning capacity on hard, often minority-class, examples. On the privacy side, differentially private training via gradient clipping and calibrated noise addition, following the general approach of DP-SGD [33], is a natural mechanism for bounding the privacy leakage of client updates beyond what federation alone provides; it is not enabled in the experiments reported here and is discussed as a planned extension.

E. Agricultural IoT Security

Compared with the broader IoT and industrial-IoT security literature, Ag-IoT-specific studies remain relatively scarce [1], [2], and existing Ag-IoT security surveys tend to focus on device-level authentication, key management, and secure firmware updates [2], [3] rather than network-traffic-based intrusion detection. This gap motivates treating general-purpose IoT intrusion-detection benchmarks as proxies, as

done in this work, while explicitly designing the system around constraints — intermittent connectivity, geographically distributed clients, limited operator security expertise, and heterogeneous low-cost hardware — that are characteristic of farm deployments specifically, rather than of IoT deployments in general (e.g., smart-home or smart-factory settings, which typically have denser connectivity and more uniform device populations). A machine-learning framework for attack detection on general IoT devices [5] and studies of insider-threat detection using NLP word embeddings [34] further illustrate that IoT and organizational-network security research has, to date, largely treated agriculture as a special case of generic IoT rather than as a setting with its own distinct connectivity and fairness constraints — the gap this study aims to address.

F. Federated Intrusion Detection for IoT and Agriculture

Several federated intrusion detection systems have been proposed for IoT and cyber-physical settings. D²IoT [17] was among the first to aggregate device-type-specific anomaly-detection models with federated learning, detecting Mirai-infected devices without sharing raw traffic. DeepFed [18] combines convolutional and gated recurrent layers with a privacy-preserving communication protocol for industrial cyber-physical systems, and Mothukuri et al. [15] train gated recurrent unit models with federated averaging for IoT anomaly detection. Popoola et al. [16] apply federated deep learning to zero-day botnet detection on IoT edge devices, and Campos et al. [10] evaluate federated intrusion detection under several non-IID partitions and show that the partitioning scheme substantially affects detection performance. In the agricultural domain, FELIDS [6] trains deep neural, convolutional, and recurrent classifiers with federated learning for agricultural IoT and reports that federated training can outperform its centralized counterparts while keeping traffic local.

Table I contrasts these approaches with the present work along the dimensions most relevant to Ag-IoT deployment. Three observations emerge. First, among the IoT intrusion detectors in Table I, none separates feature ranking from federated training, so each communication round carries a model built on the full input space; the closest precedent is our federated epilepsy-prediction system [30], which pairs HSA-MI feature selection with federated training in a medical domain. Second, accuracy is usually reported as a single global figure, whereas the per-client distribution of model quality, which determines whether small or atypical farms are protected, is rarely examined. Third, on-device cost is reported inconsistently, which makes it difficult to judge whether a detector can run on low-power gateways. The centrally trained detectors from our earlier work [19], [4] achieve high accuracy with optimization-driven architectures, but they assume that all data can be pooled. The present work targets these gaps directly, while acknowledging that it does not yet match the breadth of datasets used by some of the federated studies.

TABLE I. CRITICAL COMPARISON OF REPRESENTATIVE RELATED STUDIES

Study	Domain/data	Training	Classifier	Separate feature-selection stage	Main limitation with respect to Ag-IoT
D3IoT [17], 2019	IoT devices (Mirai testbed)	Federated	Recurrent (per device type)	No	Device-type anomaly models; not agriculture-specific
DeepFed [18], 2021	Industrial CPS	Federated	CNN + GRU	No	Heavier hybrid model; industrial rather than farm setting
Mothukuri et al. [15], 2022	General IoT	Federated (FedAvg)	GRU	No	Not agriculture-specific; on-device cost not emphasized
Popoola et al. [16], 2022	IoT edge, zero-day botnets	Federated	Deep neural network	No	Botnet-specific; not agriculture-specific
Campos et al. [10], 2022	IoT (evaluation study)	Federated	Various	No	Analysis of non-IID effects; no deployment-oriented model
FELIDS [6], 2022	Agricultural IoT	Federated	DNN, CNN, RNN	No	Full-feature deep models; footprint not emphasized
Khan [19], 2026	IoT intrusion (NSL-KDD)	Centralized	Echo state network	Metaheuristic optimization	Assumes data pooling
Khan [4], 2025	IoT botnets	Centralized	DenseNet169 variant	Metaheuristic tuning	Assumes data pooling; large model
Khan et al. [30], 2025	EEG seizure detection	Federated	LSTM / DenseNet-121	Yes (HSA-MI)	Medical domain; demonstrates the FS + FL pattern
This work	Ag-IoT proxy (UNSW-NB15)	Federated (FedAvg)	TCN, 8,834 parameters	Yes (one-time RF)	Single benchmark; binary task

G. Positioning of this Work

Relative to prior FL-IDS work, this study's main departure is the explicit separation of feature selection from federated classifier training, so that a comparatively expensive, high-capacity feature-ranking model (RF) does not need to be retrained or re-communicated every federated round; only the compact TCN classifier participates in repeated federated communication. This is complementary to, rather than a replacement for, aggregator-level innovations such as FedProx and FedAvgM, either of which can be substituted for FedAvg within the same pipeline. It is also complementary to metaheuristic-optimization-based IoT detectors [19], [4], [20]: those approaches optimize the classifier's architecture or hyperparameters directly, whereas this study optimizes the input representation (via RF-guided feature selection) ahead of a fixed, lightweight federated classifier, and the two directions could in principle be combined in future work.

III. SYSTEM AND THREAT MODEL

This section formalizes the deployment setting, the trust assumptions, and the adversaries considered, so that the scope of the detection and privacy claims made in later sections is explicit.

A. System Entities and Data Flow

The system comprises three types of entities. 1) Field devices, such as soil-moisture, nutrient, and weather sensors, irrigation and fertigation controllers, livestock wearables, drones, and autonomous machinery, generate telemetry and control traffic but are too constrained to host a classifier. 2) Farm edge gateways, which act as the federated clients, aggregate the traffic of one farm or field zone, extract flow-level features, run the deployed classifier, and perform local training; we assume single-board-computer-class hardware without a dedicated accelerator. 3) A coordinating aggregation server, operated, for example, by a farming cooperative, an agronomy service provider, or a regional extension agency, orchestrates training rounds and aggregates model updates but never receives raw traffic. In each round, the server broadcasts

the global TCN parameters, each participating gateway trains locally on its own flows, and only the updated parameters are returned. The RF used for feature selection is trained once, before federation, on a labeled reference corpus held by the coordinator; the resulting list of selected features and the trained RF, which is needed to compute the hybrid RF-probability feature, are distributed to the gateways once.

B. Operating Assumptions

We assume that 1) gateways have intermittent but recurring connectivity sufficient to exchange a model of a few tens of kilobytes per round; 2) training rounds are synchronous, and a gateway that holds no samples does not participate; 3) all gateways use the same feature-extraction pipeline, so that every client produces the same 21-dimensional input schema; and 4) heterogeneity across farms (differences in crop cycle, device population, and exposure to attacks) manifests primarily as label skew, which we model with a Dirichlet distribution with concentration parameter α , where smaller α yields more skewed partitions [12].

C. Adversary Model

Two adversaries are considered. The network adversary is an external attacker who can inject malicious traffic into the network of one or more farms, for example, denial-of-service floods against irrigation controllers, reconnaissance scans of gateway services, exploitation of unpatched device firmware, delivery of backdoors and shellcode to compromise a gateway, or worm-like propagation among field devices. These scenarios correspond to the nine attack categories of UNSW-NB15 (Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms) [21], [35], which are merged into a single attack class in this study. No single client is assumed to observe all attack types, which motivates collaborative training. The privacy adversary is an honest-but-curious aggregation server that follows the protocol but may attempt to infer properties of a farm's traffic from its model updates. In the present implementation, privacy protection is limited to data locality; that is, raw flows never leave the

gateway; formal protection of the updates themselves, such as differential privacy [33], is a planned extension (Section IV-G).

D. Out-of-Scope Threats

Malicious or compromised clients that submit poisoned updates (Byzantine or backdoor attacks on federated learning), adversarial evasion attacks crafted against the deployed classifier, physical tampering with field devices, and compromise of the aggregation server itself are outside the scope of this study. These threats are recognized in the federated learning literature [8] and require complementary defenses, such as robust aggregation and update validation, which are left to future work.

E. Design Goals

Within this model, the framework pursues four goals: G1 (detection), a high recall of attack traffic with a false-positive rate low enough for operators with limited security expertise; G2 (data locality), no raw traffic leaves the farm; G3 (resource efficiency), the deployed classifier and its per-round communication fit the memory, compute, and bandwidth budget of a farm gateway; and G4 (client fairness), the global

model does not serve some farms markedly worse than others. Section VI evaluates G1, G3, and G4 quantitatively, whereas G2 holds by construction.

IV. PROPOSED METHODOLOGY

A. Pipeline Overview

The pipeline has two decoupled stages, illustrated conceptually as: raw flow features → [central RF feature selection] → reduced feature set → [federated TCN training across edge clients] → global lightweight classifier. Decoupling the two stages means the (relatively expensive, one-time) feature-selection step does not need to be repeated every federated round, while the repeated federated step trains only the compact downstream classifier that is actually deployed on-device. The rationale is that feature ranking benefits from a high-capacity model that is comparatively costly to train, whereas the deployed detector must be small; separating the two stages lets each use the model best suited to its role [14], [13]. Fig. 1 summarizes the complete methodology, including the data exchanged between the three stages and the values used in this study.

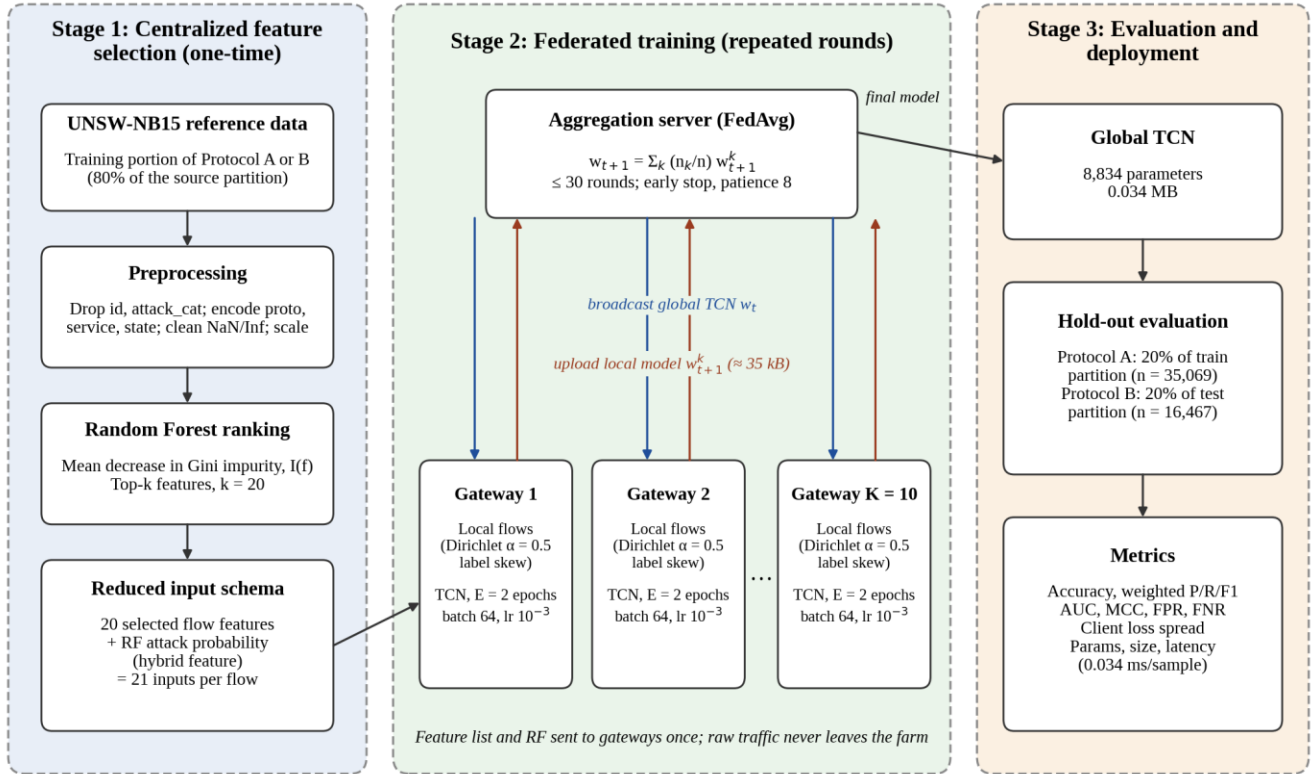


Fig. 1. Methodology of the proposed framework: one-time centralized RF feature selection (Stage 1), federated TCN training with FedAvg across K = 10 non-IID farm gateways (Stage 2), and hold-out evaluation and deployment profiling (Stage 3).

B. Random-Forest-Guided Feature Selection

An RF classifier [13] is trained on a centralized reference copy of the labeled training data (the training portion of each evaluation protocol, Section V-B). An RF was chosen for ranking because it handles mixed categorical and continuous flow features without distributional assumptions, captures non-linear feature interactions, provides an importance measure as a by-product of training, and tree ensembles are consistently

competitive on tabular intrusion benchmarks [13], [1]. Feature importances from the trained forest are ranked using the mean Gini-impurity decrease attributable to each feature, averaged over all T trees in the forest:

$$I(f) = (1/T) \sum_{t=1}^T \sum_{n: v(n)=f} p(n) \cdot \Delta i(n), \quad (1)$$

where, t indexes the T trees, n ranges over internal nodes whose split variable $v(n)$ equals feature f, $p(n)$ is the proportion

of training samples reaching node n , and $\Delta_i(n)$ is the impurity decrease produced by that split. The top- k features by $I(f)$ ($k = 20$ in this study) are retained. An optional correlation-pruning step removes features that are highly correlated with an already-selected, higher-importance feature, reducing redundancy without discarding independently informative signals. The RF's own predicted attack probability is appended as an additional hybrid input feature to the downstream classifier, giving the TCN a compressed summary of the tree ensemble's view of each sample in addition to the raw selected features. The value $k = 20$ was fixed a priori as a compromise between retaining most of the importance mass and keeping the input of the deployed classifier small; a sensitivity analysis over k is left to future work. Because the hybrid feature is computed by the RF, the RF must also be available at each gateway at inference time (Section IV-I).

C. Lightweight Temporal Convolutional Classifier

The primary classifier is a compact TCN [14] operating on the RF-selected feature vector (treated as a short pseudo-sequence). Causal, dilated 1-D convolutions give the network a controllable receptive field with a small parameter count relative to an equivalently deep recurrent network. For a filter f of size K and dilation factor d , the output at time step t is

$$y_t = K^{-1} \sum_{i=0}^{K-1} f(i) \cdot x_{t-d \cdot i}, \quad (2)$$

with causality enforced because y_t depends only on $x_t, x_{t-d}, \dots, x_{t-d(K-1)}$, that is, on the present and past but never the future. Stacking L such layers with exponentially increasing dilation ($d = 1, 2, 4, \dots, 2^{L-1}$) grows the receptive field exponentially in depth while keeping the parameter count linear in L , which is the source of the TCN's parameter efficiency relative to an equivalently receptive recurrent network. An optional squeeze-and-excitation (SE) block [27] adds negligible parameter overhead while allowing the network to re-weight temporal-convolutional channels adaptively through a learned per-channel gate. The TCN was preferred over recurrent alternatives such as the GRU [36] because its convolutions parallelize across the sequence and its parameter count grows only linearly with depth [14]. In the evaluated configuration, the 21-dimensional input (20 selected features plus the hybrid feature) is treated as a length-21 sequence with one channel and processed by $[L]$ residual temporal blocks with $[C]$ channels and kernel size $[K]$, followed by global average pooling and a two-unit output layer, for a total of 8,834 trainable parameters; the SE block is [enabled/disabled].

D. Federated Training Protocol

Training proceeds in synchronous rounds. In each round, the server broadcasts the current global model to all clients that hold data; each client performs E local epochs of mini-batch training on its own partition, and the server aggregates the returned models with FedAvg [7]. FedAvg was selected as the reference aggregator because it is the de facto baseline in the federated intrusion detection literature [15], [10], so that the results can be interpreted against prior work and serve as the reference point for the planned aggregators of Section IV-G. In FedAvg, the client models are averaged, weighted by each client's number of local samples n_k out of the total $n = \sum_k n_k$ across the K participating clients:

$$w_{t+1} = \sum_{k=1}^K n_k / n \cdot w_{t+1}^k \quad (3)$$

A cosine learning-rate schedule is applied across rounds, and training stops early once validation F1 fails to improve for eight consecutive rounds, avoiding wasted federated communication once the global model has converged.

E. Handling Non-IID Data and Class Imbalance

Client partitions are constructed by drawing, for each class c , a proportion vector across the K clients from a symmetric Dirichlet distribution [12]:

$$p_c \sim \text{Dir}(\alpha), p_c \in \Delta^{K-1} \quad (4)$$

where, α is a concentration parameter, and Δ^{K-1} is the $(K-1)$ -simplex; smaller α produces more skewed, less-IID client label distributions, and $\alpha \rightarrow \infty$ recovers an approximately IID split. This is a standard mechanism for simulating realistic, non-uniform label skew across Ag-IoT sites without requiring naturally pre-partitioned field data. To counter the resulting local class imbalance, the implementation provides three complementary options; their individual effects have not yet been ablated (Section VII):

- Class-weighted cross-entropy loss, where per-class weights w_c are computed from the (local or global) label distribution:

$$L_{CE} = -\sum_{c=1}^C w_c y_c \log(\hat{y}_c) \quad (5)$$

- Focal loss [32], which down-weights the contribution of well-classified examples via a tunable focusing parameter γ , concentrating gradient signal on hard/minority examples:

$$FL(p_i) = -\alpha_i (1 - p_i)^\gamma \log(p_i) \quad (6)$$

- Client-local SMOTE [31], which synthesizes minority-class examples within each client's local partition before training, without ever sharing raw or synthetic samples across clients.

F. Fairness Across Clients

Because Ag-IoT clients vary widely in traffic volume and attack exposure, we track a simple fairness diagnostic alongside global accuracy: the spread (minimum, maximum, and standard deviation) of average local training loss across active clients within each round. A large spread indicates that the global model is converging well for some clients while performing poorly for others — a signal that raw accuracy alone would hide.

G. Planned Extensions (Not Evaluated in This Study)

Several components were designed but are not implemented or evaluated in the present study, and no results are reported for them: 1) the FedProx [11] and FedAvgM [12] aggregators, which add a proximal term to the local objective and server-side momentum, respectively, to counter client drift under non-IID data; 2) a GRU [36] baseline of comparable size; 3) differentially private update protection that clips each client update to a maximum L2 norm and adds calibrated Gaussian noise, following DP-SGD [33]; and 4) a post-hoc ensemble that blends RF and TCN class probabilities at

evaluation time. They are listed here only to delimit the scope of the present results and are prioritized in Section VIII.

H. Computational Complexity

Let d denote the number of raw engineered flow features, $k \leq d$ the number of RF-selected features, C the number of TCN channels, L the number of temporal-convolutional layers, and T the pseudo-sequence length induced by the chosen windowing of the k selected features. The one-time centralized RF training cost scales with the standard Random Forest training complexity in the number of trees, samples, and d ; this cost is amortized across all subsequent federated rounds since it is not repeated per round. Per-round federated cost is dominated by local TCN training at each active client, which scales linearly in the number of local epochs, local batch count, and the TCN's own per-example cost, $O(L \cdot C^2 \cdot T)$ for the convolutional layers plus a small constant overhead for the optional SE block; because k is chosen to be much smaller than d , and C and L are kept small by design, this per-round cost is intended to remain tractable on constrained edge hardware, which is verified empirically via the parameter-count and model-size profile reported in Section VI.

I. Communication Cost

Federated communication cost per round is proportional to the trainable parameter count of the deployed classifier (the TCN), transmitted twice per participating client per round (broadcast of the global model, upload of the local update), independent of the RF's own size, since the RF is trained once centrally and never transmitted as part of the repeated federated protocol. Keeping the deployed classifier small therefore directly bounds both the per-round communication cost and the on-device memory/inference footprint, which is the motivating design goal behind pairing RF-based feature selection with a compact TCN rather than federating a larger model directly on the full feature space. For the evaluated model, 8,834 parameters stored as 32-bit floating-point values amount to approximately 35 kB per transfer, or about 0.7 MB per round for ten clients (one download and one upload each), which is compatible with metered rural links. Because the hybrid RF-probability feature is used, the RF must also be distributed to each gateway once before deployment; this one-time cost is not repeated per round, but it adds to the on-device memory footprint and is not included in the model size reported in Section VI.

V. EXPERIMENTAL SETUP

A. Dataset and Preprocessing

UNSW-NB15 [21], [35] is used as the evaluation benchmark and as a proxy for Ag-IoT traffic. It was chosen because it is a widely used, labeled flow-level dataset with nine modern attack categories plus benign traffic; because it is distributed with fixed official training (175,341 records, of which 56,000 are benign) and testing (82,332 records, of which 37,000 are benign) partitions that support reproducible comparison [35]; and because its attack categories map naturally to the farm-level scenarios of Section III. All attack categories are merged into a single attack class (binary task: class 0 = benign, class 1 = attack). Preprocessing drops the identifier and label-leaking columns (id and attack_cat),

encodes the categorical fields (proto, service, and state), replaces missing and infinite values, and scales the numerical features before RF-based selection. Two further IoT benchmarks, Edge-IIoTset [22] and CIC-IoT2023 [23], are supported by the same preprocessing pipeline but were not evaluated in this study; they are planned for future work (Section VIII).

B. Evaluation Protocols

Two hold-out protocols were used, and their results are reported separately because they are not directly comparable. In Protocol A, a stratified 20% hold-out of the official training partition ($n = 35,069$; 31.9% benign) serves as the test set. In Protocol B, a stratified 20% hold-out of the official testing partition ($n = 16,467$; 44.9% benign) serves as the test set. The test-set sizes and class proportions of both protocols coincide with these fractions of the respective partitions ($0.2 \times 175,341 \approx 35,069$ and $0.2 \times 82,332 \approx 16,467$; $56,000/175,341 = 31.9\%$ and $37,000/82,332 = 44.9\%$ benign). In each protocol, the remaining 80% of the same partition is used for RF fitting, client partitioning, and validation-based early stopping, so that test records are never used for feature selection or training.

C. Client Simulation

Following feature selection, the training split is partitioned across a configurable number of simulated clients (10 by default) using Dirichlet(α) label-skewed sampling, with $\alpha = 0.5$, a moderately skewed setting widely used in the federated learning literature [12], [10]; ten clients approximate a small cooperative of farms. Clients with zero assigned samples in a given partition are excluded from that round's active-client count.

D. Model and Training Hyperparameters

TABLE II. EXPERIMENTAL CONFIGURATION USED IN BOTH PROTOCOLS

Item	Setting
Dataset/task	UNSW-NB15; binary (benign vs. attack)
Test protocols	A: 20% of official training partition ($n = 35,069$); B: 20% of official testing partition ($n = 16,467$)
Feature selection	RF, mean decrease in Gini impurity; $k = 20 + \text{RF attack probability (21 inputs)}$
RF hyperparameters	[number of trees; maximum depth; random seed]
Classifier	TCN, 8,834 trainable parameters; [blocks; channels; kernel size; SE on/off]
Clients/partition	10 clients; Dirichlet $\alpha = 0.5$ label skew
Aggregator	FedAvg (sample-size weighted)
Rounds / local epochs	≤ 30 rounds (early stopping, patience 8); 2 local epochs
Batch size/learning rate	64; 10^{-3} with cosine decay
Optimizer/loss	[optimizer; loss function and class weighting]
Software/hardware	[framework and version; CPU/GPU used for training and latency measurement]

Table II lists the complete configuration used in both protocols. The RF retains $k = 20$ features; federated training runs for at most 30 rounds with 2 local epochs per round, a local batch size of 64, an initial learning rate of 10^{-3} with cosine decay, and early stopping with a patience of eight rounds on validation F1. Two local epochs limit client drift

under non-IID data while keeping on-device computation small [7], [11]; a batch size of 64 fits comfortably in gateway memory; and the 30-round cap with a patience of eight is generous relative to the observed convergence, which plateaus within the first five rounds in both protocols (Fig. 3 and 5).

E. Evaluation Metrics

Detection quality is reported on the hold-out test set of each protocol with accuracy, weighted precision, weighted recall, weighted F1, and the area under the receiver operating characteristic curve (AUC), together with per-class precision and recall, the false-positive rate (FPR, benign traffic flagged as attack), the false-negative rate (FNR, attacks missed), and the confusion matrix. Because the two protocols have different class ratios, the Matthews correlation coefficient, $MCC = (TP \cdot TN - FP \cdot FN) / \sqrt{((TP + FP)(TP + FN)(TN + FP)(TN + FN))}$, is also reported; MCC uses all four cells of the confusion matrix and is more informative than accuracy or F1 under class imbalance [37]. FPR is reported explicitly because, for operators with limited security expertise, false alarms translate directly into alert fatigue (goal G1). Deployment feasibility (goal G3) is characterized by the trainable parameter count, the serialized model size, and the mean per-sample inference latency. Federated-process quality and fairness (goal G4) are characterized by per-round validation curves, total wall-clock training time, and the spread (minimum, maximum, and standard deviation) of average local training loss across clients (Section IV-F).

VI. RESULTS AND DISCUSSION

This section reports the results of the implemented configuration, the RF-selected TCN trained with FedAvg on UNSW-NB15, under the two evaluation protocols defined in Section V-B. Both runs use the same 21 input features, the same aggregator, and the same TCN architecture, which is the only classifier implemented in the reference code; they differ only in the official partition from which the training and test records were drawn and, consequently, in their class balance. Protocol A (Fig. 2–4) stopped early after 15 rounds, and Protocol B (Fig. 5–7) after 17 rounds. Results for the planned

extensions of Section IV-G are not reported because those components were not run.

A. Protocol A: Hold-Out from the Official Training Partition

Fig. 2 shows the average local training loss across active clients per communication round for Protocol A. The loss falls sharply from 0.312 in round 1 to 0.043 by round 3 and settles into a low, stable band around 0.008–0.015 from round 5 through round 15, at which point training stopped, consistent with the early-stopping patience of eight rounds described in Section V-D being triggered after validation F1 stopped improving.

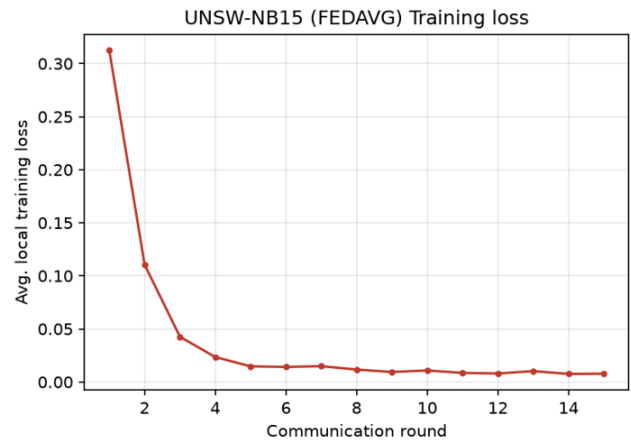


Fig. 2. Average local training loss per communication round for Protocol A (TCN with FedAvg on UNSW-NB15, 10 clients).

Fig. 3 shows the corresponding per-round validation accuracy and F1. Both metrics rise from 90.5%/90.0% (accuracy/F1) in round 1 to 95.8%/95.8% by round 3, peak at round 7 (96.01% accuracy, 96.00% F1), and then oscillate in a narrow 95.87–96.01% band through round 15 rather than monotonically improving, a mild non-monotonicity plausibly attributable to the heterogeneity across the 10 non-IID clients discussed below, since each round's global model reflects a different weighted mix of client updates.

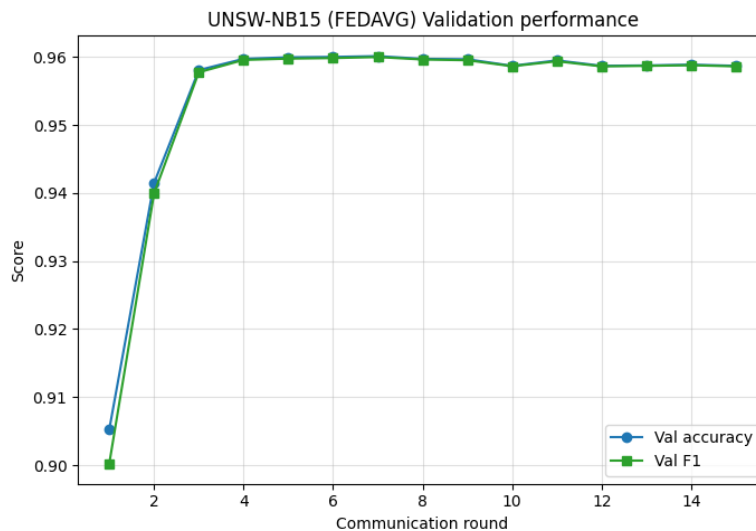


Fig. 3. Validation accuracy and F1 per communication round for Protocol A.

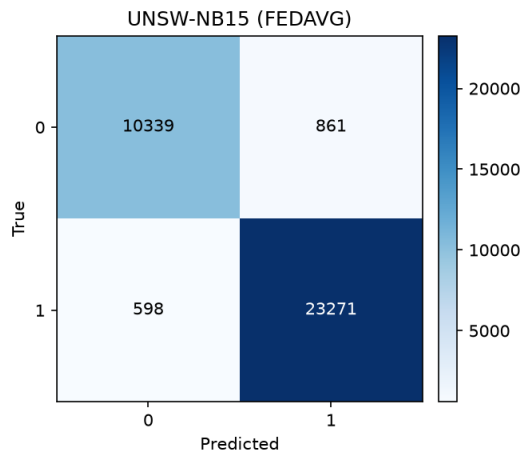


Fig. 4. Test-set confusion matrix for Protocol A; $n = 35,069$.

Fig. 4 shows the test-set confusion matrix for Protocol A (10,339 true negatives, 861 false positives, 598 false negatives, and 23,271 true positives; $n = 35,069$). These counts give an accuracy of 95.84%, a weighted precision of 95.83%, a weighted recall of 95.84%, a weighted F1 of 95.83%, an AUC of 0.988, and an MCC of 0.904 (Table III). Attack traffic is detected with 96.4% precision and 97.5% recall, whereas benign traffic is recognized with 94.5% precision and 92.3% recall. The resulting FNR (2.5%) is low, but the FPR (7.7%) is about three times larger: in this protocol, benign traffic is the minority class (31.9%), and the model errs toward flagging benign flows as attacks. For an IDS, this is the safer failure mode, but at operational traffic volumes an FPR of 7.7% would generate a substantial alert load, which motivates the class-imbalance mechanisms discussed in Section IV-E.

The client-fairness diagnostic introduced in Section IV-F is summarized in Table IV: across the 10 clients, average local training loss ranges from 0.00255 (client 7) to 0.126 (client 4), a spread with standard deviation 0.0313; client 4's loss is roughly 4 $\times$ the across-client median (≈ 0.032), illustrating

exactly the kind of per-client heterogeneity this metric is designed to surface even though the global test accuracy (95.84%) looks unremarkable on its own. Per-round wall-clock time was also uneven, ranging from 41.2 s (round 13) to 217.6 s (round 11), for a total training time of 1,697.2 s (≈ 28.3 minutes) across 15 rounds; this variability likely reflects fluctuation in active-client count or local compute load per round rather than a fixed per-round cost, and is worth stabilizing in future runs if wall-clock predictability matters for deployment scheduling.

Table IV also summarizes the deployment profile: 8,834 trainable parameters, a serialized size of 0.0337 MB, and a mean inference latency of 0.0336 ms per sample. Because both protocols use the same input dimension and TCN configuration, the parameter count and model size apply to both runs. These figures are several orders of magnitude below the memory available on single-board-computer gateways, which supports goal G3; however, the latency was measured on the training host rather than on gateway hardware, and the one-time cost of distributing the RF required for the hybrid feature is not included.

B. Protocol B: Hold-Out from the Official Testing Partition

Fig. 5 shows the corresponding average local training loss, which falls from 0.44 in round 1 to 0.086 in round 2 and below 0.03 from round 4 onward, remaining flat through round 17. The steeper initial drop than in Fig. 2 is consistent with the different training partition used in Protocol B.

Fig. 6 shows validation accuracy and validation F1 per communication round for Protocol B. Both curves jump from about 0.932 (accuracy) at round 1 to above 0.972 by round 2, then plateau in a narrow 0.973–0.975 band through round 17, evidence that, consistent with the fast-convergence pattern also seen in Fig. 2–3 above, the great majority of achievable validation performance is reached within the first two or three communication rounds, with later rounds contributing only marginal refinement.

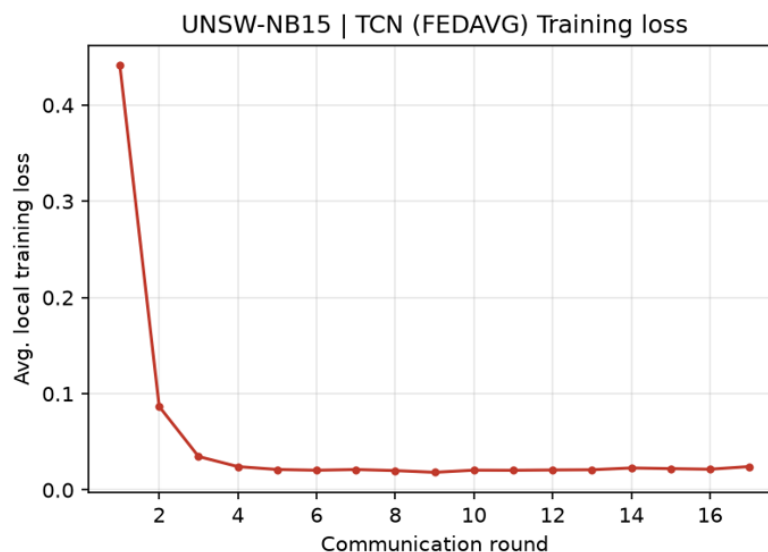


Fig. 5. Average local training loss per communication round for Protocol B.

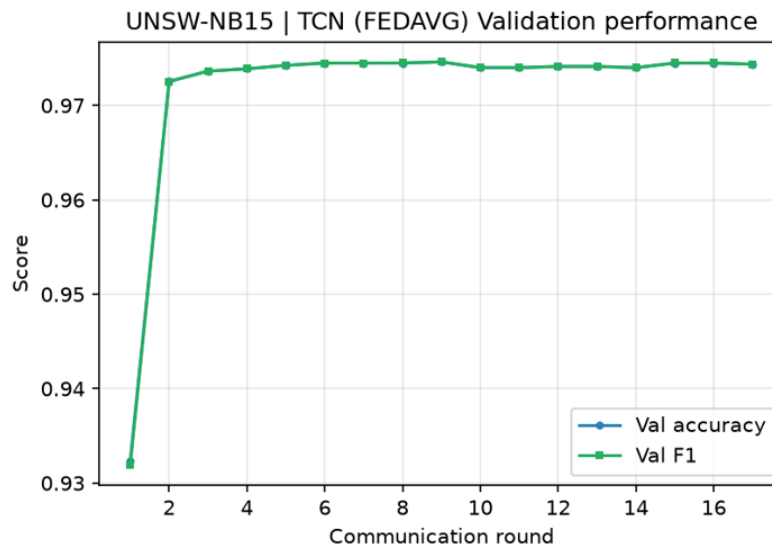


Fig. 6. Validation accuracy and F1 per communication round for Protocol B (TCN with FedAvg on UNSW-NB15).

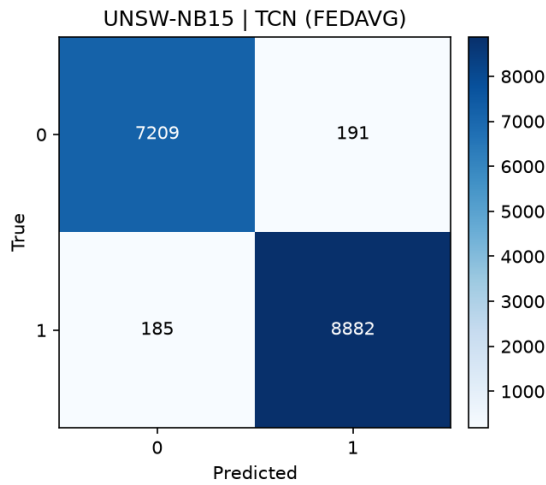


Fig. 7. Test-set confusion matrix for Protocol B; $n = 16,467$.

Fig. 7 shows the test-set confusion matrix for Protocol B (7,209 true negatives, 191 false positives, 185 false negatives,

and 8,882 true positives; $n = 16,467$). This gives an accuracy, weighted precision, weighted recall, and weighted F1 of 97.72% and an MCC of 0.954 (Table III). Benign precision and recall are 97.5% and 97.4%, and attack precision and recall are 97.9% and 98.0%, so the FPR (2.6%) and FNR (2.0%) are nearly balanced. Compared with Protocol A, the FPR is about three times lower, whereas the FNR is similar.

C. Consolidated Results and Interpretation

Table III consolidates both protocols. The accuracy gap of 1.9 percentage points and the larger MCC gap (0.904 vs. 0.954) should not be read as run-to-run instability of an otherwise identical experiment: the two protocols draw their training and test records from different official partitions whose class composition differs (31.9% vs. 44.9% benign records) [21], [35]. The gap is concentrated in the benign class (benign recall of 92.3% vs. 97.4%), that is, in the class that is the minority in Protocol A, which is consistent with a class-imbalance effect rather than an architectural one. Because each protocol was run with a single seed, the variance attributable to the seed cannot yet be separated from the protocol effect; repeated runs are listed as a limitation in Section VII.

TABLE III. DETECTION PERFORMANCE OF THE FEDERATED TCN (FEDAVG, UNSW-NB15, $K = 20 +$ HYBRID FEATURE) UNDER THE TWO EVALUATION PROTOCOLS

Metric	Protocol A	Protocol B
Test set (n ; benign share)	35,069; 31.9%	16,467; 44.9%
Rounds until early stopping	15	17
Accuracy	95.84%	97.72%
Weighted precision/recall	95.83% / 95.84%	97.72% / 97.72%
Weighted F1	95.83%	97.72%
AUC	0.988	Not recorded
MCC	0.904	0.954
Benign precision/recall	94.53% / 92.31%	97.50% / 97.42%
Attack precision/recall	96.43% / 97.49%	97.89% / 97.96%
Benign F1 / attack F1	93.41% / 96.96%	97.46% / 97.93%
FPR / FNR	7.69% / 2.51%	2.58% / 2.04%

TABLE IV. DEPLOYMENT PROFILE AND FEDERATED-PROCESS STATISTICS (PROTOCOL A; MODEL FIGURES APPLY TO BOTH PROTOCOLS)

Statistic	Value
Trainable parameters	8,834
Model size	0.0337 MB
Avg. inference latency	0.0336 ms/sample
Total training time (15 rounds)	1,697.2 s ($\approx$ 28.3 min)
Round time, min/max	41.2 s / 217.6 s
Client loss std/min/max	0.0313 / 0.0025 / 0.1264
Communication per round (10 clients)	$\approx$ 0.7 MB ($\approx$ 35 kB per model transfer)

TABLE V. RF-SELECTED FEATURES FOR UNSW-NB15 (K = 20 + RF ATTACK-PROBABILITY HYBRID FEATURE)

#	Feature	#	Feature
1	sttl	12	smean
2	ct_state_ttl	13	dbytes
3	dttl	14	dinpkt
4	rate	15	dur
5	sload	16	djit
6	dload	17	state
7	dmean	18	sinpkt
8	synack	19	sjit
9	ct_srv_dst	20	ct_dst_sport_ltm
10	ackdat	21	rf_attack_proba (hybrid)
11	sbytes		

TABLE VI. COMPARISON WITH BASELINES ON IDENTICAL SPLITS (UNSW-NB15, BINARY TASK)

Model	Training	Protocol A accuracy / MCC	Protocol B accuracy / MCC	Parameters
RF, all features	Centralized	[] / []	[] / []	—
RF, k = 20 features	Centralized	[] / []	[] / []	—
TCN, 21 inputs	Centralized	[] / []	[] / []	8,834
Published federated IDS (re-implemented)	Federated	[] / []	[] / []	[]
TCN, 21 inputs (proposed)	Federated (FedAvg)	95.84% / 0.904	97.72% / 0.954	8,834

Three findings emerge from Tables III and IV. First, a federated TCN with fewer than 9,000 parameters reaches 95.8–97.7% accuracy and an MCC of 0.90–0.95 on UNSW-NB15 under non-IID partitioning, converging within three to five communication rounds in both protocols (Fig. 3 and 5); for bandwidth-limited farms, rapid convergence matters as much as final accuracy because every round consumes connectivity. Second, the evaluation protocol, and not only the model, materially affects the reported numbers: the same architecture and aggregator yield a threefold difference in FPR depending on which official partition supplies the data, so results reported on a single, unspecified split should be compared with caution. Third, global accuracy hides substantial client-level disparity: in Protocol A, the average local loss of the worst-served client is about 50 times that of the best-served client (Table IV), so a farm with an atypical traffic mix could receive a markedly weaker detector than the headline accuracy suggests.

Table V lists the selected features. The top-ranked attributes are dominated by time-to-live values (sttl, dttl, and ct_state_ttl), traffic rate and load (rate, sload, and dload), and TCP handshake timing (synack and ackdat), which is consistent with attack categories that alter flow volume and

connection setup; payload-content fields are absent, which suggests that the detector relies on header-level behavior that remains observable when payloads are encrypted. At the same time, time-to-live attributes in UNSW-NB15 partly reflect the configuration of the testbed that generated the data and may transfer imperfectly to real farm networks, which reinforces the need for validation on farm-collected traffic.

The main advantages of the approach are therefore its small and measurable deployment footprint, its fast convergence, and the fact that raw traffic never leaves the farm, which together address goals G1–G3. Its main weaknesses are the elevated FPR under the imbalanced Protocol A, the client-level disparity noted above (goal G4), and the dependence on a centrally trained RF. Aggregators designed to counteract client drift, such as FedProx [11] and FedAvgM [12], and the class-imbalance options of Section IV-E are natural next steps for reducing the disparity and the FPR; their benefit, however, has not been measured and is not claimed here. Table VI is intended to provide the centralized and published reference points needed to quantify the accuracy cost of federation on identical splits.

D. Validation of Results

Beyond the accuracy/precision/recall/F1/AUC already reported, Table III also reports the Matthews Correlation Coefficient (MCC) — computed directly from each run's confusion matrix (Fig. 4 and 7) as defined in Section V-E, because accuracy alone can be misleading under the class imbalance present in both UNSW-NB15 runs [37] (support ratios of roughly 45:55 and 32:68 for benign: attack in Protocols B and A, respectively). This follows the validation practice used in our own prior work on digital financial fraud detection [28], which reported a Matthews Correlation Coefficient of 0.9945 alongside accuracy, precision, recall, and F1 specifically to demonstrate robustness on a highly imbalanced dataset, rather than relying on accuracy alone. By the same measure, Protocol B attains MCC = 0.954 and Protocol A attains MCC = 0.904; both indicate strong agreement between predicted and true labels well above the MCC = 0 chance baseline, corroborating the accuracy-based conclusions in Sections VI-A and VI-B rather than contradicting them, and the gap between the MCC values of the two protocols (0.954 vs. 0.904) is consistent in direction and rough magnitude with the accuracy gap already discussed above (97.72% vs. 95.84%).

A second validation practice from our own prior work is component-wise ablation: our federated epilepsy-prediction study [30] validated each part of its pipeline (the HSA-MI

feature-selection stage and the federated-learning stage) via a dedicated ablation study rather than reporting only the full system's headline accuracy and reported extensive experimental evaluation across multiple runs as part of establishing that result. The results reported in this study do not yet include an equivalent ablation — for example, RF-guided feature selection (Section IV-B) with vs. without the `rf_attack_proba` hybrid feature, or the squeeze-and-excitation block (Section IV-C) enabled vs. disabled — nor repeated runs across multiple random seeds to establish variance around the 97.72%/95.84% accuracy figures reported here. Following the precedent set in [30], [28], a seed-averaged ablation study (reporting mean $\pm$ standard deviation across at least 3–5 seeds per configuration and isolating each proposed component's individual contribution) is identified as the next concrete validation step for this codebase, additional to — not a replacement for — implementing the missing aggregators and baselines noted in Section VII.

E. Comparison with Related Work

Table VII positions this study's implemented results (Section VI-A–C) alongside the headline accuracy figures reported by four of our own closely related prior studies, each applying a comparable optimization-plus-deep-learning or feature-selection-plus-federated-learning strategy to a different detection domain.

TABLE VII. POSITIONING AGAINST OUR PRIOR WORK (REPORTED ACCURACY; DIFFERENT DOMAINS AND DATASETS)

Study	Domain / Dataset	Approach	Federated?	Accuracy
Khan [19], 2026	IoT intrusion detection (NSL-KDD)	Addax optimization + echo state network	No	98.45%
Khan [4], 2025	IoT botnet detection	DenseNet169 + RSE/ASPP, CE-HSOA-optimized	No	99.00%
Khan et al. [30], 2025	Epileptic seizure detection (EEG)	HSA-MI feature selection + LSTM/DenseNet-121	Yes	99.41%
Naif, Khan & Almuğhairi [28], 2026	Digital financial fraud detection	Multi-octave dilated TCN, XAI + meta-learning	No	99.13% (MCC 0.9945)
This work, Protocol B	Ag-IoT proxy intrusion detection (UNSW-NB15)	RF feature selection + TCN, FedAvg	Yes	97.72% (MCC 0.954)
This work, Protocol A	Ag-IoT proxy intrusion detection (UNSW-NB15)	RF feature selection + TCN, FedAvg	Yes	95.84% (MCC 0.904)

This table is a positioning summary, not a controlled benchmark: each row's accuracy was measured by its own authors on its own dataset and task, so no cell-to-cell ranking should be read as one method outperforming another on shared ground. Two systematic differences are worth noting when interpreting the gap between this work (95.84–97.72%) and [19], [4], [28] (98.45–99.13%): first, [19], [4], and [28] all train centrally rather than in a federated manner, so their figures do not pay the accuracy cost, if any, of federated averaging over non-IID clients that this study's TCN/FedAvg runs do incur; and second, EEG seizure detection and financial-transaction fraud detection are established, comparatively well-separated classification tasks in the literature, whereas network intrusion detection on UNSW-NB15 mixes many attack categories with subtler flow-level signatures. Of the four prior studies, only the epilepsy-prediction system [30] is, like this study, federated, and its 99.41% accuracy on a federated medical task suggests the accuracy levels reported here for FedAvg on UNSW-NB15 are not an inherent ceiling for federated approaches in general, although a direct comparison would require running both systems on a shared dataset.

Published federated detectors [6], [15], [16], [17] are not added to Table VII because their results were obtained on different datasets (for example, CSE-CIC-IDS2018, MQTTset, and InSDN for FELIDS [6]) or on device-specific testbeds [17], with different class definitions and splits, so a cell-by-cell comparison would be misleading. The appropriate comparison is to re-run a representative published federated detector, together with centralized RF and TCN baselines, on exactly the Protocol A and B splits, as organized in Table VI.

VII. LIMITATIONS

- Scope of the evaluation. Only the FedAvg aggregator with the TCN classifier is evaluated, on one benchmark (UNSW-NB15) and one task (binary detection). FedProx, FedAvgM, the GRU baseline, differential privacy, and the RF+TCN ensemble are described only as planned extensions (Section IV-G), and no results are claimed for them.
- Baselines and repeated runs. Centralized RF and TCN baselines and a published federated detector have not

yet been evaluated on the Protocol A and B splits (Table VI), and each protocol was run with a single random seed, so neither the accuracy cost of federation nor seed-to-seed variance is yet quantified.

- Hybrid-feature dependency. The RF attack-probability feature requires the RF at each gateway at inference time, and if it is computed on the samples the RF was fitted on, it may be optimistically biased; an ablation without this feature and an out-of-fold computation are needed to confirm its contribution.
- Simulated federation. Clients are simulated by partitioning existing centralized datasets rather than collected from physically distinct farm deployments; real Ag-IoT networks may exhibit additional heterogeneity (device firmware, sensor sampling rates, seasonal traffic patterns) not captured by Dirichlet label-skew alone.
- Proxy dataset. UNSW-NB15 is a general network-intrusion benchmark generated in a laboratory testbed rather than an Ag-IoT-specific capture, and some of its most informative attributes partly reflect the testbed configuration; results should be validated on farm-collected traffic and on IoT benchmarks such as Edge-IIoTset [22] and CIC-IoT2023 [23] before operational deployment.
- Centralized RF dependency. Feature selection depends on a centrally trained RF, which requires at least one round of centralized or centrally pooled data access; fully decentralized feature selection is left to future work.
- No formal privacy guarantee. Privacy protection in the evaluated system rests on data locality alone; model updates are neither clipped nor perturbed, and no (ϵ, δ) privacy budget is derived.
- Static attack taxonomy. The system is evaluated against the attack families present in its training benchmarks and would require retraining or continual-learning extensions to cover novel, previously unseen attack types.

VIII. CONCLUSION AND FUTURE WORK

This study presented a lightweight federated intrusion detection framework for Agricultural IoT that separates a one-time, centralized Random Forest feature-selection stage from repeated federated training of a compact Temporal Convolutional Network. Evaluated with FedAvg on UNSW-NB15 under Dirichlet non-IID partitioning across ten clients, the 8,834-parameter TCN (0.034 MB, 0.034 ms per sample) reached 95.84% accuracy (MCC 0.904, AUC 0.988) on a hold-out of the official training partition and 97.72% accuracy (MCC 0.954) on a hold-out of the official testing partition, converged within three to five communication rounds, and required approximately 0.7 MB of traffic per round for ten clients.

Two implications follow. Practically, the results indicate that a federated detector small enough for farm gateways and

inexpensive enough for metered rural links can retain high detection quality, provided that feature ranking is performed once, outside the federated loop. Methodologically, the threefold FPR difference between the two protocols and the approximately 50-fold spread of client losses show that federated IDS studies should report the exact evaluation split, MCC and per-class error rates, and a client-level fairness measure alongside global accuracy.

The study is limited to one aggregator, one benchmark, a binary task, single-seed runs, simulated clients, and privacy protection by data locality alone, and same-split centralized and published baselines are still required to quantify the accuracy cost of federation.

Future work will, in order of priority, 1) complete the same-split baselines of Table VI and repeat both protocols over at least five random seeds; 2) implement FedProx and FedAvgM and test whether they reduce the client-loss spread and the FPR under increasingly skewed partitions (α from 0.1 to 1.0); 3) ablate the hybrid RF-probability feature, the squeeze-and-excitation block, and k ; 4) extend the evaluation to multi-class detection and to Edge-IIoTset and CIC-IoT2023; 5) add differentially private update protection with formal (ϵ, δ) accounting and robust aggregation against poisoned updates; and 6) validate the pipeline on traffic captured from operational farm gateways.

REFERENCES

- [1] M. A. Ferrag, L. Shu, O. Friha, and X. Yang, "Cybersecurity intrusion detection for agriculture 4.0: Machine learning-based solutions, datasets, and future directions," *IEEE/CAA Journal of Automatica Sinica*, vol. 9, no. 3, pp. 407–436, 2022.
- [2] X. Yang, L. Shu, J. Chen, M. A. Ferrag, J. Wu, E. Nurellari, and K. Huang, "A survey on smart agriculture: Development modes, technologies, and security and privacy challenges," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 2, pp. 273–302, 2021.
- [3] M. Gupta, M. Abdelsalam, S. Khorsandroo, and S. Mittal, "Security and privacy in smart farming: Challenges and opportunities," *IEEE Access*, vol. 8, pp. 34564–34584, 2020.
- [4] M. A. R. Khan, "DenseRSE-ASPPNet: An enhanced DenseNet169 with residual dense blocks and CE-HSOA-based optimization for IoT botnet detection," *International Journal of Advanced Computer Science and Applications*, vol. 16, no. 4, 2025.
- [5] A. J. Abdul Khader Jilani, F. Ahmad, and M. A. R. Khan, "Machine learning-based framework for attack detection on IoT devices," in *Proc. IEEE 8th Int. Conf. Engineering Technologies and Applied Sciences*, 2023.
- [6] O. Friha, M. A. Ferrag, L. Shu, L. Maglaras, K.-K. R. Choo, and M. Nafaa, "FELIDS: Federated learning-based intrusion detection system for agricultural Internet of Things," *Journal of Parallel and Distributed Computing*, vol. 165, pp. 17–31, 2022.
- [7] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Int. Conf. Artificial Intelligence and Statistics (AISTATS)*, 2017.
- [8] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, et al., "Advances and open problems in federated learning," *Foundations and Trends in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.
- [9] D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, J. Li, and H. V. Poor, "Federated learning for Internet of Things: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 3, pp. 1622–1658, 2021.
- [10] E. M. Campos, P. F. Saura, A. González-Vidal, J. L. Hernández-Ramos, J. Bernal Bernabé, G. Baldini, and A. Skarmeta, "Evaluating federated

- learning for intrusion detection in Internet of Things: Review and challenges,” *Computer Networks*, vol. 203, p. 108661, 2022.
- [11] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, “Federated optimization in heterogeneous networks,” in *Proc. Machine Learning and Systems (MLSys)*, 2020.
- [12] T.-M. H. Hsu, H. Qi, and M. Brown, “Measuring the effects of non-identical data distribution for federated visual classification,” *arXiv:1909.06335*, 2019.
- [13] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [14] S. Bai, J. Z. Kolter, and V. Koltun, “An empirical evaluation of generic convolutional and recurrent networks for sequence modeling,” *arXiv:1803.01271*, 2018.
- [15] V. Mothukuri, P. Khare, R. M. Parizi, S. Pouriyeh, A. Dehghantanha, and G. Srivastava, “Federated-learning-based anomaly detection for IoT security attacks,” *IEEE Internet of Things Journal*, vol. 9, no. 4, pp. 2545–2554, 2022.
- [16] S. I. Popoola, R. Ande, B. Adebisi, 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.
- [17] T. D. Nguyen, S. Marchal, M. Miettinen, H. Fereidooni, N. Asokan, and A.-R. Sadeghi, “DfIoT: A federated self-learning anomaly detection system for IoT,” in *Proc. IEEE 39th Int. Conf. Distributed Computing Systems (ICDCS)*, 2019, pp. 756–767.
- [18] B. Li, Y. Wu, J. Song, R. Lu, T. Li, and L. Zhao, “DeepFed: Federated deep learning for intrusion detection in industrial cyber-physical systems,” *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5615–5624, 2021.
- [19] M. A. R. Khan, “Self-adaptive intrusion detection framework for IoT networks using Addax optimization and echo state networks,” *International Journal of Integrated Engineering*, vol. 18, no. 4, pp. 111–129, 2026.
- [20] M. A. R. Khan, S. N. Shavkatovich, B. Nagpal, A. Kumar, M. A. Haq, V. J. Tharini, et al., “Optimizing hybrid metaheuristic algorithm with cluster head to improve performance metrics on the IoT,” *Theoretical Computer Science*, vol. 927, pp. 87–97, 2022.
- [21] N. Moustafa and J. Slay, “UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set),” in *Proc. Military Communications and Information Systems Conf. (MilCIS)*, Canberra, Australia, 2015, pp. 1–6.
- [22] M. A. Ferrag, O. Friha, D. Hamouda, L. Maglaras, and H. Janicke, “Edge-IIoTset: A new comprehensive realistic cyber security dataset of IoT and IIoT applications for centralized and federated learning,” *IEEE Access*, vol. 10, pp. 40281–40306, 2022.
- [23] E. C. P. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu, and A. A. Ghorbani, “CICIoT2023: A real-time dataset and benchmark for large-scale attacks in IoT environment,” *Sensors*, vol. 23, no. 13, p. 5941, 2023.
- [24] S. Almotairi, M. A. R. Khan, O. Alharbi, Z. Alzaid, Y. M. Hausawi, and J. Almutairi, “Detection of Android malware using deep learning ensemble with cheetah-optimized feature selection,” *Advances and Applications in Discrete Mathematics*, vol. 41, no. 5, p. 357, 2024.
- [25] M. A. R. Khan, N. Kumar, and R. C. Tripathi, “Detection of Android malware app through feature extraction and classification of Android image,” *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 5, pp. 906–914, 2022.
- [26] A. K. Jilani and M. A. R. Khan, “An obfuscation attack detection approach in Android APK using ML classification of grayscale and RGB images,” in *Sustainable Innovations in Management in the Digital Transformation Era*, pp. 352–357, 2026.
- [27] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [28] M. A. R. Khan, Y.H.Naif and S. S. Almughairi, “A novel lightweight multi-octave dilated temporal convolutional network with explainable AI and meta-learning approach based digital financial fraud detection,” *Journal of Advances in Information Technology*, vol. 17, no. 4, pp. 816–836, 2026.
- [29] Q. Yang, Y. Liu, T. Chen, and Y. Tong, “Federated machine learning: Concept and applications,” *ACM Transactions on Intelligent Systems and Technology*, vol. 10, no. 2, pp. 1–19, 2019.
- [30] M. A. R. Khan, K. M. Heba, A. H. Abbass, M. A. Shaikh, G. G. Tejani, S. K. Sharma, et al., “An epilepsy prediction and management system based on federated learning combined with hybrid harmony search and mutual information (HSA-MI)-based feature selection approach,” *Scientific Reports*, vol. 16, no. 1, p. 893, 2025.
- [31] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “SMOTE: Synthetic minority over-sampling technique,” *J. Artificial Intelligence Research*, 2002.
- [32] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” in *Proc. IEEE Int. Conf. Computer Vision (ICCV)*, 2017, pp. 2980–2988.
- [33] M. Abadi et al., “Deep learning with differential privacy,” in *Proc. ACM SIGSAC Conf. Computer and Communications Security (CCS)*, 2016.
- [34] M. A. Haq, M. A. R. Khan, and M. Alshehri, “Insider threat detection based on NLP word embedding and machine learning,” *Intelligent Automation & Soft Computing*, vol. 33, no. 1, pp. 619–635, 2022.
- [35] N. Moustafa and J. Slay, “The evaluation of network anomaly detection systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set,” *Information Security Journal: A Global Perspective*, vol. 25, no. 1–3, pp. 18–31, 2016.
- [36] K. Cho et al., “Learning phrase representations using RNN encoder-decoder for statistical machine translation,” in *Proc. Conf. Empirical Methods in Natural Language Processing (EMNLP)*, 2014.
- [37] D. Chicco and G. Jurman, “The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation,” *BMC Genomics*, vol. 21, p. 6, 2020.