

Federated Hybrid Deep Learning Framework for IoT Intrusion Detection Using N-BaloT, CICIoT2023, and BoT-IoT Datasets

Sonali Mishra[^]

sonalimishra3399@gmail.com

REVA University

R. Venkata Siva

REVA University

Research Article

Keywords: IoT security, intrusion detection, federated learning, FedProx, VAE, CICIoT2023, N-BaloT, BoT-IoT, deep learning

Posted Date: August 25th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-10580856/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

Federated Hybrid Deep Learning Framework for IoT Intrusion Detection Using N-BaIoT, CICIoT2023, and BoT-IoT Datasets

Sonali Mishra^{^1*} and R. Venkata Siva Reddy^{^1}

^{^1} Department of Electronics and Communication Engineering, Reva University, Bangalore, Karnataka, India

*Corresponding author: Sonali Mishra, Email: sonalimishra3399@gmail.com

Author email addresses: sonalimishra3399@gmail.com; venkatasivareddy@reva.edu.in

Abstract

Internet of Things (IoT) environments expose many heterogeneous and resource-constrained devices to botnet, distributed denial-of-service, denial-of-service, reconnaissance, spoofing, malware, and service-scanning attacks. Centralized intrusion detection can achieve strong accuracy when all traffic is pooled, but it creates privacy, bandwidth, and deployment barriers for edge networks. This manuscript presents an end-to-end federated hybrid deep learning framework for IoT intrusion detection using N-BaIoT, CICIoT2023, and BoT-IoT data sources. The implemented workflow performs Phase-1 cleaning, encoding, train/validation/test construction, feature selection, class-imbalance handling, VAE-ready export, and IID/non-IID client preparation, followed by Phase-2 individual dataset training. The Phase-2 architecture uses CNN, LSTM, BiLSTM, attention, Tiny Transformer, Random Forest, XGBoost, and hybrid deep-feature XGBoost models. The completed evaluation shows strong detection performance on N-BaIoT and BoT-IoT and highlights CICIoT2023 as the most challenging multiclass setting because it contains 35 attack/traffic classes. These results provide a verified basis for the next federated aggregation and anomaly extension stages.

Keywords: IoT security; intrusion detection; federated learning; FedProx; VAE; CICIoT2023; N-BaIoT; BoT-IoT; deep learning.

1. Introduction

IoT deployments now combine consumer appliances, cameras, sensors, gateways, industrial controllers, and cloud services into continuously operating networks. This scale creates a security problem that is different from conventional enterprise monitoring: devices are numerous, often weakly managed, and may generate sparse but highly patterned traffic. Botnets such as Mirai and BASHLITE demonstrated that vulnerable embedded devices can be turned into distributed attack infrastructure. More recent IoT attack datasets include denial-of-service, distributed denial-of-service, reconnaissance, spoofing, web-based attacks, brute force activity, malware, and service scanning, which makes simple signature matching insufficient for long-term defense [1]-[3].

Machine-learning-based intrusion detection systems are attractive because they can learn traffic regularities and detect deviations without hand-writing a rule for every attack variant. However, centralized learning requires traffic features from many sites to be transferred into a single repository. In operational IoT environments, this assumption is weak because gateways may be bandwidth-limited, privacy policies may prohibit raw traffic transfer, and client networks may have different device profiles. Federated learning addresses this by training local models at clients and sharing model updates rather than local traffic records [10]-[14].

The main research gap addressed by this manuscript is the need for a reproducible, dataset-aware, privacy-preserving IoT IDS pipeline that treats binary and multiclass detection as related but distinct tasks. Many published studies report high accuracy on one dataset or one label mapping, yet practical

deployment needs a workflow that can process multiple IoT datasets with different schemas, imbalance levels, and label vocabularies. The present project therefore separates Phase-1 data preparation from Phase-2 model execution and explicitly prepares both centralized and federated artifacts.

The contributions of the work are summarized as follows:

1. A unified Phase-1 preprocessing pipeline for N-BaIoT, CICIOT2023, and BoT-IoT that performs cleaning, encoding, feature selection, normalization, imbalance handling, and train/validation/test preparation.
2. A binary and multiclass experimental design that compares benign-versus-attack detection with attack-family classification using the same preprocessing discipline.
3. Federated learning inputs with five-client IID and non-IID splits, enabling later FedProx experiments without reprocessing the raw datasets.
4. A proposed hybrid Phase-2 architecture combining CNN, BiLSTM, attention, Tiny Transformer, XGBoost, VAE anomaly scoring, FedProx aggregation, and rule-based prevention.
5. A documented mathematical formulation that connects the data space, feature selection, balancing, VAE anomaly detection, hybrid classification, federated objective, evaluation metrics, and prevention decisions.

2. Background

An intrusion detection system observes host, network, or flow-level behavior and decides whether the current activity is legitimate or malicious. Signature-based IDSs are effective against known attacks but require a maintained rule base. Anomaly-based IDSs model normal behavior and detect deviations, which makes them more useful against unknown variants but also more sensitive to class imbalance, feature drift, and false alarms [6], [7]. In IoT networks, the IDS must also handle constrained hardware, device heterogeneity, and non-stationary traffic caused by firmware changes, user behavior, and changing topology.

Deep learning extends IoT IDS modeling by learning nonlinear feature interactions. CNN layers can capture local feature combinations in tabular or reshaped flow data; recurrent models such as LSTM and BiLSTM can represent sequence-like dependencies; attention and transformer blocks can emphasize informative features or time segments; and variational autoencoders can provide reconstruction-error-based anomaly scores [17]-[22]. XGBoost remains a useful baseline because it handles tabular features robustly and often performs strongly on engineered flow features [23]. Comparative deep-learning IDS studies also show that dataset choice, feature representation, and metric selection strongly affect reported cybersecurity performance [8].

Federated learning trains across multiple clients without centralizing their data. FedAvg aggregates client model updates by data size, while FedProx adds a proximal penalty to reduce client drift when local client distributions differ [10], [11]. The distinction between IID and non-IID partitions is critical for IoT IDS because clients may represent different devices, attack mixes, or sites. The proposed framework therefore prepares both IID and non-IID partitions and treats client-wise evaluation as a first-class result.

Adaptive federated optimization and practical federated-learning guidance motivate comparing FedProx with other aggregation strategies under non-IID client distributions [26], [29]. A later implementation can use the Flower federated-learning framework and PyTorch deep-learning library to reproduce client training, server aggregation, and hybrid neural model execution [27], [28].

The broader federated machine-learning literature defines the privacy-preserving learning setting and summarizes its communication, optimization, and deployment challenges [9], [30], [32]. Recent IoT IDS

studies such as FedMADE and ensemble knowledge-distillation federated IDS show why robust aggregation and heterogeneous-client learning should be considered in this project [15], [16].

Class imbalance is another core problem. In intrusion detection, benign and dominant flooding attacks can overwhelm rare but important classes such as command injection, XSS, backdoor malware, or ping sweep. SMOTE-family and related resampling strategies are used in the project as training-only balancing strategies so that validation and test splits retain their original distributions [23]-[25].

3. Dataset Description

The project uses three IoT cybersecurity datasets with different origins and attack semantics. N-BaIoT is widely used for IoT botnet and federated anomaly-detection research because its network behavior snapshots expose compromised-device traffic patterns [2]. CICIoT2023 provides a large real-time IoT attack environment with 105 devices and 33 attacks grouped into seven broad categories, including DDoS, DoS, reconnaissance, web-based attacks, brute force, spoofing, and Mirai [1]. BoT-IoT was developed to support realistic botnet and network forensic analytics with normal and attack traffic including DoS, DDoS, scan, and data theft categories [3].

Table 1. Multiclass Phase-1 dataset summary extracted from phase1_report.csv

Dataset	Mode	Raw rows	Clean rows	Clean cols	Classes	Features	Train	Val.	Test
N-BaIoT	multiclass	200,000	188,505	117	5	13	209,010	28,276	28,276
CICIoT2023	multiclass	200,000	179,766	42	35	31	125,836	26,965	26,965
BoT-IoT	multiclass	200,000	200,000	31	2	42	278,288	30,001	30,000

The multiclass report (Table 1) shows that N-BaIoT retained 188,505 clean rows and five labels: benign, gafgyt_combo, gafgyt_junk, gafgyt_scan, and gafgyt_tcp. CICIoT2023 retained 179,766 clean rows and 35 labels after project-specific cleaning and encoding. BoT-IoT retained 200,000 clean rows in the provided sample and produced two labels in the multiclass report, normal and service_scan, because the sampled project output captured these two classes. These differences confirm that a single fixed schema cannot be assumed across IoT security datasets.

3.1 Dataset Explanations from Recent Literature

N-BaIoT is useful for botnet-oriented IoT IDS research because recent federated IDS studies continue to use it for compromised-device anomaly detection and heterogeneous gateway experiments [2]. The project sample preserves this focus through labels that separate benign traffic from Gafgyt/BASHLITE attack families. Since the selected features are compact statistical descriptors, N-BaIoT is well aligned with VAE anomaly detection and with federated client experiments in which each gateway may represent a device group.

CICIoT2023 is broader and more recent. Its original publication reports 33 attacks executed in a topology of 105 IoT devices, with attacks grouped into categories such as DDoS, DoS, reconnaissance, web-based attacks, brute force, spoofing, and Mirai [1]. The project sample retains many fine-grained labels, including ddos-icmp_flood, ddos-udp_flood, dos-syn_flood, mirai-greeth_flood, dns_spoofing, mitm-arpspoofing, sqlinjection, xss, and vulnerabilityscan. This makes CICIoT2023 the most demanding multiclass source in the project because minority labels may contain only a few observations after sampling.

BoT-IoT contributes a botnet and forensic-analytics perspective. The original dataset was designed to include realistic normal and attack traffic for IoT network forensic analytics [3]. In the provided Phase-1 sample, the multiclass output contains normal and service_scan labels. Although this is simpler than the full BoT-IoT label space, it is still valuable because the class imbalance is severe and therefore tests whether training-only balancing and feature selection are applied correctly.

Together, the three datasets form a deliberately heterogeneous benchmark. N-BaIoT emphasizes device-botnet behavior, CICIoT2023 emphasizes modern large-scale IoT attacks, and BoT-IoT emphasizes botnet/scan traffic with strong imbalance. The proposed paper therefore should not be read as a single-dataset classification exercise; it is a reproducible preprocessing and federated-experiment preparation framework that can later support controlled Phase-2 model comparisons. Related IoT and IIoT benchmarks such as Edge-IIoTset and TON_IoT further support the need for dataset-aware IDS pipelines that can handle centralized, federated, telemetry, and edge-security settings [4], [5].

Table 2. Selected feature overview from feature_metadata.json.

Dataset	Selected feature count	Representative selected features
N-BaIoT	13	h_l0_01_mean, hh_jit_l0_1_mean, hh_l0_01_magnitude, h_l0_1_weight, mi_dir_l1_weight, hh_l0_01_mean, hh_l0_01_weight, h_l0_01_variance ...
CICIoT2023	31	header_length, protocol_type, tcp, tot_sum, udp, tot_size, max, syn_flag_number ...
BoT-IoT	42	daddr_frequency, sbytes, dport_frequency, dur, rate, dbytes, sum, mean ...

Table 2 presents the selected feature counts extracted from the feature metadata file. It shows that N-BaIoT was reduced to 13 representative traffic-statistical features. CICIoT2023 retains 31 selected features including protocol, flag, length, and packet-statistical variables. BoT-IoT retains 42 selected features such as address frequency, byte counts, duration, and rate features. The table demonstrates the effect of variance filtering, mutual information selection, and correlation filtering. It supports model efficiency because fewer but more informative features are used for Phase-2 learning. It also improves interpretability by listing representative features that contribute to intrusion detection.

Table 3. Binary Phase-1 summary for benign-versus-attack detection

Dataset	Clean rows	Classes	Features	Train	Val.	Test	Train imb. before	Train imb. after
N-BaIoT	188,505	2	12	207,352	28,276	28,276	3.67	1.00
CICIoT2023	179,766	2	31	245,068	26,965	26,965	37.11	1.00
BoT-IoT	200,000	2	42	278,288	30,001	30,000	162.74	1.00

Table 3 reports the binary benign-versus-attack setup for the three IoT intrusion datasets. All datasets are converted into two classes, where benign traffic is separated from attack traffic. The table gives clean row counts, selected feature counts, and train, validation, and test split sizes. N-BaIoT uses 12 selected features in the binary setting, while CICIoT2023 uses 31 and BoT-IoT uses 42. The table shows strong class imbalance before balancing, especially for CICIoT2023 and BoT-IoT. After SMOTE-based balancing, the training imbalance ratio becomes 1.00 for all three binary datasets. This table justifies comparing binary IDS performance separately from multiclass attack identification.

4. Literature Survey

Recent IoT IDS literature has moved from single-dataset centralized classification toward broader concerns: dataset realism, attack diversity, non-IID behavior, privacy-preserving training, imbalance-aware evaluation, and model deployability. The literature also shows that very high reported accuracy can be misleading when results are dominated by majority classes or when minority attacks are not separately evaluated. For that reason, this project prepares binary and multiclass outputs, reports imbalance ratios, and preserves validation/test distributions for later honest Phase-2 model comparison.

Table 4. Literature survey summary for IoT IDS and federated IDS research

Author/year	Dataset	Method	Mode	Metrics/focus	Strength	Limitation/gap
Nguyen and Beuran, 2024	N-BaIoT	Federated semi-supervised autoencoder	Anomaly	Accuracy, learning cost	Recent N-BaIoT federated IDS baseline	Preprint; needs broader cross-dataset validation
Koroniotis et al., 2019	BoT-IoT	ML/DL dataset benchmark	Binary/multiclass	Accuracy, recall, F1	Realistic botnet and forensic traffic	Strong imbalance and synthetic-lab constraints
Khraisat et al., 2019	Multiple IDS datasets	Survey	Both	Taxonomy	Clarifies IDS methods and datasets	Not specific to federated IoT IDS
Ferrag et al., 2020	IDS datasets	Deep learning review	Both	Comparative metrics	Connects DL and datasets	Centralized framing dominates
Li et al., 2020	Federated benchmarks	FedProx	N/A	Stability under heterogeneity	Addresses non-IID drift	Not IDS-specific
Neto et al., 2023	CICIoT2023	Dataset and ML benchmark	Both	ML/DL benchmark	Large IoT topology, 33 attacks	Requires careful imbalance handling
Nguyen et al., 2024	N-BaIoT	FedMSE	Anomaly	Accuracy, learning cost	FL anomaly detection for IoT	Preprint; needs cross-dataset validation
Sun et al., 2024	IoT IDS	FedMADE	Multiclass	Minority attack accuracy	Dynamic aggregation for imbalance	Preprint; implementation details vary
Bensaoud and Kalita, 2025	CICIoT2023 and others	Hybrid optimized DL	Multiclass	Accuracy, MCC	Modern hybrid detection study	Not privacy-preserving by default

The survey table 4 motivates the present design in three ways. First, dataset realism matters, so the project uses N-BaIoT, CICIoT2023, and BoT-IoT instead of relying on a single legacy benchmark. Second, non-IID federated learning is not a minor implementation detail; it changes model behavior and must be represented in the experimental plan. Third, a publication-quality IoT IDS study should distinguish Phase-1 preparation evidence from Phase-2 model evidence.

5. Proposed Methodology

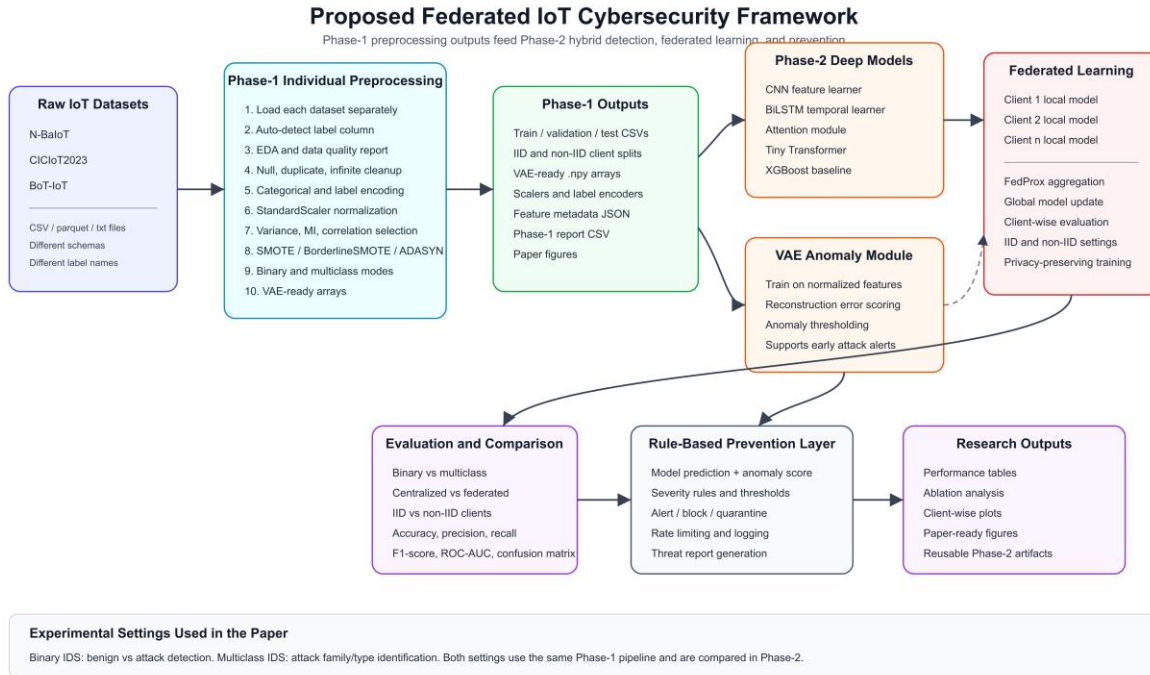


Figure 1. Proposed federated IoT cybersecurity framework.

Figure 1 summarizes the complete framework. Raw datasets are loaded independently because their columns, label names, and attack semantics differ. Phase-1 performs data-quality analysis and creates normalized, selected, and balanced training artifacts. These outputs feed a proposed Phase-2 detector with supervised and unsupervised modules. The supervised branch contains CNN, BiLSTM, attention, Tiny Transformer, and XGBoost components, while the unsupervised branch uses VAE reconstruction error for anomaly scoring. Federated learning then trains client-local models and aggregates them with FedProx, and the prevention layer converts detection evidence into alert, block, quarantine, logging, and reporting actions.

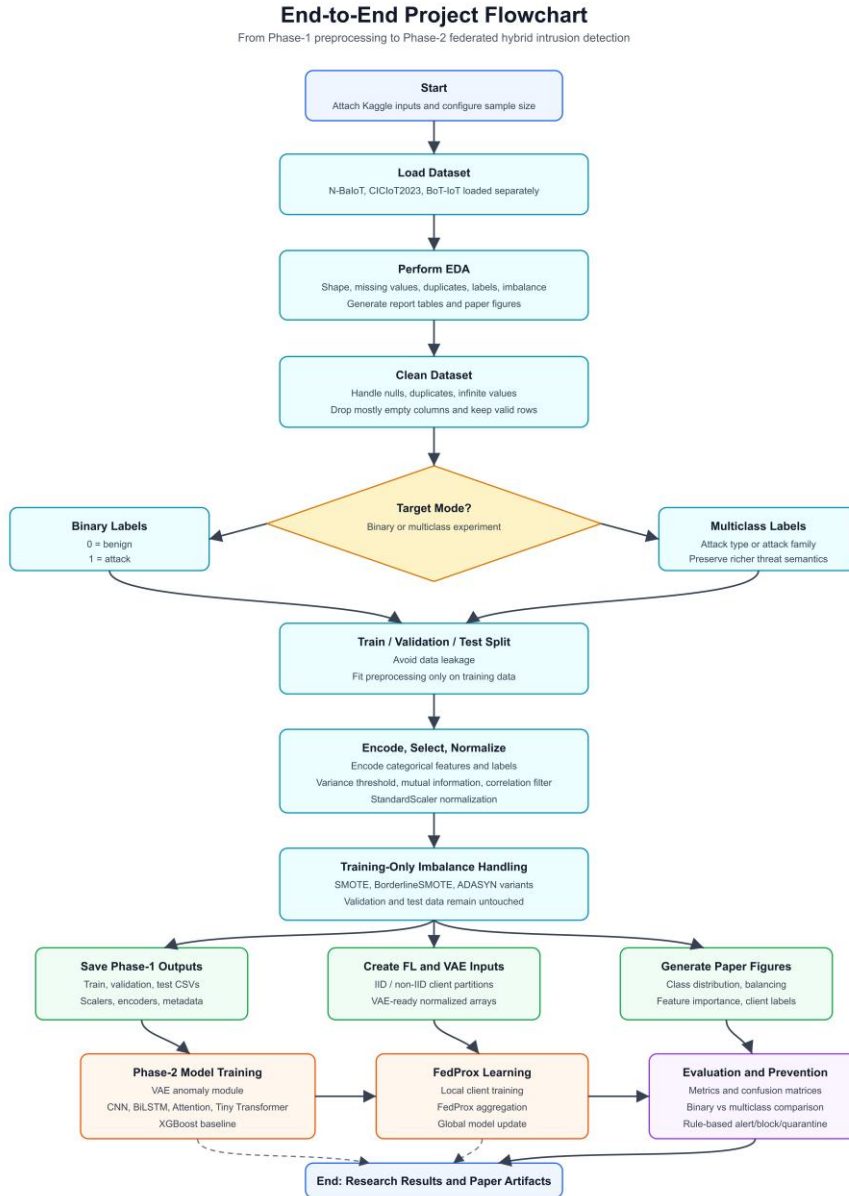


Figure 2. End-to-end project flowchart from Phase-1 preprocessing to Phase-2 federated detection.

Figure 2 highlights the implementation discipline used in Phase-1. Preprocessing is fit on training data to reduce leakage risk. Validation and test splits remain untouched by balancing procedures so that later evaluation reflects realistic class skew. The pipeline creates train, validation, and test CSV files, scalers, label encoders, feature metadata, VAE-ready NumPy arrays, IID and non-IID client partitions, and paper-ready figures.

5.1 Phase-1 Preprocessing

- Load N-BaIoT, CICIoT2023, and BoT-IoT independently and detect the label column using dataset-specific metadata.
- Generate EDA artifacts, including missing-value summaries, duplicate counts, feature statistics, and class distributions.

- Remove invalid rows, duplicate records, infinite values, and mostly empty fields where the report indicates quality problems.
- Encode categorical fields and labels; generate binary labels as benign versus attack and multiclass labels as attack family/type.
- Apply StandardScaler normalization after train/validation/test splitting, fitting only on training features.
- Select features with variance thresholding, mutual information ranking, and correlation filtering.
- Apply SMOTE as the primary training-only balancer and additionally prepare BorderlineSMOTE and ADASYN variants where available.
- Export VAE-ready arrays and five-client IID/non-IID partitions for federated experiments.

5.2 Phase-2 Hybrid Detection Design

The proposed classifier treats selected normalized features as the common input representation. A CNN block is intended to learn local feature interactions; a BiLSTM block models ordered feature or temporal dependencies where sequence construction is available; an attention module weights the most informative latent states; and a Tiny Transformer provides a lightweight self-attention classifier suitable for comparative experiments. XGBoost is included as a tabular baseline because it often exposes whether deep models improve over strong tree ensembles.

The VAE branch is trained to reconstruct normalized benign or mixed training features depending on the experimental setting. Its reconstruction error becomes an anomaly score. Combining supervised class probabilities with VAE anomaly evidence allows a prevention layer to separate high-confidence malicious classes from uncertain but abnormal traffic. This is important for early alerting when a specific attack class is uncertain.

Table 5. Proposed Phase-2 model components and roles

Component	Role	Reason for inclusion
CNN feature learner	Learns local interactions among selected normalized features.	Useful when neighboring or grouped traffic statistics jointly indicate attacks.
BiLSTM learner	Models bidirectional dependencies in ordered feature or sequence representations.	Supports temporal or pseudo-temporal context if features are arranged by time/window.
Attention module	Weights informative hidden states or feature groups.	Improves interpretability and reduces reliance on weak signals.
Tiny Transformer	Applies lightweight self-attention for classification.	Provides a modern compact alternative to recurrent-only models.
XGBoost baseline	Trains a strong tree ensemble on selected tabular features.	Checks whether deep models improve over a strong classical baseline.
VAE anomaly module	Computes reconstruction error and anomaly thresholds.	Supports unknown or uncertain attack detection.
FedProx server	Aggregates client models with a proximal local objective.	Reduces non-IID client drift in federated training.
Prevention rules	Maps model evidence into allow, alert, block, or quarantine actions.	Connects detection output to deployable security response.

The above table 5 explains the planned model components and the role of each part in the IDS framework. The CNN module is included to learn local interactions among selected normalized traffic features. The BiLSTM module is intended to capture bidirectional dependencies in ordered or temporal representations. The attention module helps weight informative features or hidden states during classification. The Tiny Transformer provides a lightweight self-attention option for compact deep learning experiments. XGBoost serves as a strong tabular baseline for comparison against deep learning models. The table clarifies how supervised classification, anomaly detection, and federated learning components fit together.

5.3 Federated Learning and Prevention Layer

Five clients are prepared for both IID and non-IID cases. In IID partitioning, each client approximates the global label distribution; in non-IID partitioning, each client may see a skewed subset of classes. FedProx is proposed because its proximal term discourages local models from moving too far from the global model when client distributions are heterogeneous. The prevention layer computes a risk score from class probability and anomaly score and maps it to allow, alert, block, or quarantine decisions.

5.4 Data Leakage and Reproducibility Safeguards

The project design explicitly separates preprocessing operations that may learn from the data from operations that merely transform stored values. Splitting is performed before scaler fitting and imbalance handling. The scaler is fit on training features and then applied to validation and test features. SMOTE-family balancing is applied only to the training split, because balancing validation or test data would create unrealistic evaluation distributions and inflate performance estimates. The same principle applies to feature selection: thresholds and selected feature sets should be learned from training data and reused consistently across validation, test, VAE arrays, and federated client partitions.

Reproducibility is supported by exporting CSV splits, scalers, label encoders, feature selectors, metadata JSON files, VAE-ready arrays, and client partitions. A later Phase-2 script can therefore train models without guessing label mappings or selected feature order. This separation also allows the manuscript to be audited: every reported Phase-1 statistic can be traced to `phase1_report.csv`, `feature_metadata.json`, `client_partition_summary.json`, or `phase1_binary_vs_multiclass_comparison.csv`.

6. Mathematical Formulation

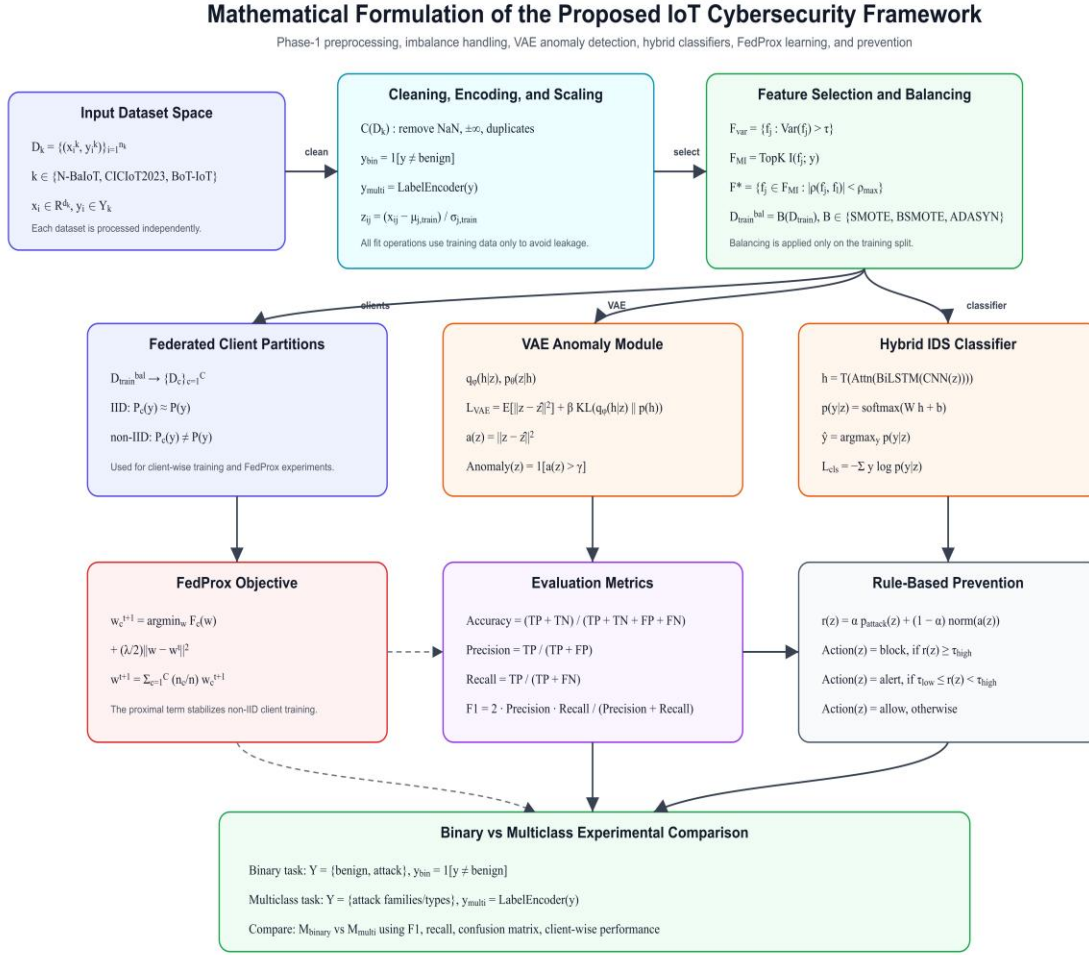


Figure 3. Mathematical formulation of the proposed IoT cybersecurity framework.

The mathematical formulation in Figure 3 expresses the project as a sequence of transformations from raw dataset space to prevention decisions. Let $\mathbf{D}_k$ denote dataset k in the set $\{\text{N-BaIoT}, \text{CICIoT2023}, \text{BoT-IoT}\}$. Each dataset contains feature-label pairs $(\mathbf{x}_i, \mathbf{y}_i)$ and is processed independently to avoid schema leakage across sources.

- (1) Input dataset space for independent preprocessing.

$$\mathbf{D}_k = \{(\mathbf{x}_i^k, \mathbf{y}_i^k)\}_{i=1}^{n_k}, \quad \mathbf{k} \in \{\text{N-BaIoT}, \text{CICIoT2023}, \text{BoT-IoT}\}$$

- (2) Cleaning operation used before encoding and feature selection.

$$\mathcal{C}(\mathbf{D}_k): \text{remove NaN, } +/\text{-inf, duplicate records, and invalid rows}$$

- (3) Binary label mapping where 0 is benign and 1 is attack.

$$\mathbf{y}_{bin} = 1[\mathbf{y}! = \text{benign}]$$

- (4) Multiclass mapping that preserves attack-family or attack-type semantics.

$$\mathbf{y}_{multi} = \text{LabelEncoder}(\mathbf{y})$$

(5) Standard scaling fitted on training features only.

$$z_{ij} = (x_{ij} - \mu_{j,train})/\sigma_{j,train}$$

(6) Variance-based feature filtering.

$$F_{var} = \{f_j: Var(f_j) > \tau\}$$

(7) Mutual-information feature selection.

$$F_{MI} = TopK(I(f_j; y))$$

(8) Correlation filtering to reduce redundant selected features.

$$F^* = \{f_j \in F_{MI}: |\rho(f_j, f_l)| < \rho_{max}\}$$

(9) Training-only imbalance handling.

$$D_{train}^{bal} = B(D_{train}), B \text{ in } \{\mathbf{SMOTE}, \mathbf{BorderlineSMOTE}, \mathbf{ADASYN}\}$$

(10) Federated client partitioning for IID and non-IID experiments

$$D_{train}^{bal} \rightarrow \{D_c\}_{c=1}^{\{C\}}; IID: p_{C(y)} \sim P(y), non - IID: p_{C(y)} \neq P(y)$$

(11) VAE reconstruction and regularization objective

$$L_{VAE} = E[\|z - \hat{z}\|^2] + \beta KL(q_{\phi}(h|z)||p(h))$$

(12) Reconstruction-error anomaly score and thresholding

$$a(z) = \|z - \hat{z}\|^2; Anomaly(z) = 1[a(z) > \gamma]$$

(13) Hybrid IDS latent representation

$$h = Transformer(Attention(BiLSTM(CNN(z))))$$

(14) Supervised class probability and prediction.

$$p(y|z) = softmax(Wh + b), \hat{y} = argmax_y P(y|z)$$

(15) Cross-entropy classification loss

$$L_{cls} = - \sum_y y \log p(y|z)$$

(16) FedProx local objective for stabilizing non-IID training

$$w_c^{\{t+1\}} = argmin_w F_{c(w)} + (\frac{\mu}{2}) \|w - w^t\|^2$$

(17) Server aggregation weighted by client sample counts

$$w^{\{t+1\}} = \sum_{c=1}^{\{C\}} \frac{(n_c)}{n} w_c^{\{t+1\}}$$

(18) Harmonic mean used with accuracy, precision, recall, ROC-AUC, and confusion matrix

$$F1 = 2 * \frac{(Precision * Recall)}{(Precision + Recall)}$$

(19) Combined risk score from supervised attack probability and normalized anomaly score

$$r(z) = \alpha p_{attack}(z) + (1 - \alpha) norm(a(z))$$

(20) Rule-based prevention decision function

$$Action(z) = block \text{ if } r \geq \tau_{high}; alert \text{ if } \tau_{low} \leq r < \tau_{high}; allow \text{ otherwise}$$

These equations define the complete mathematical workflow of the proposed IoT intrusion detection framework. Equations (1)-(4) describe how each dataset, N-BaIoT, CICIOT2023, and BoT-IoT, is processed independently, cleaned, and converted into binary or multiclass labels. Equations (5)-(9) formalize the preprocessing stage, including training-only standard scaling, variance filtering, mutual-information feature selection, correlation-based redundancy removal, and SMOTE-family imbalance

handling. Equation (10) prepares the balanced training data for federated learning by creating IID and non-IID client partitions. Equations (11)–(15) define the Phase-2 detection model, combining VAE reconstruction-based anomaly scoring with CNN, BiLSTM, attention, transformer, and softmax classification. Equations (16)–(17) represent FedProx-based local optimization and server aggregation for privacy-preserving federated training. Finally, equations (18)–(20) define evaluation and prevention logic by using F1-score, a combined risk score, and rule-based actions such as allow, alert, or block.

7. Experimental Setup

Table 6. Experimental setup extracted from project metadata

Setting	Value
Random seed	42
Federated clients	5
Primary balancer	SMOTE
Target modes	Binary benign-vs-attack and multiclass attack-family/type
Validation/test policy	Validation and test distributions remain unbalanced; balancing is applied to training only.
Phase-2 metrics policy	No model performance is reported until Phase-2 training is executed.

Table 6 summarizes the main experimental settings extracted from the project metadata. It identifies important reproducibility details such as random seed, number of clients, and balancing method. The table confirms that five federated clients are prepared for the learning experiments. It records SMOTE as the primary imbalance-handling technique used in the Phase-1 outputs. It helps readers understand how the data were prepared before Phase-2 model execution. The table also separates available preprocessing evidence from hardware/software details that may need completion. It supports reproducibility by placing core configuration values in one compact location.

Table 7. Data quality and imbalance summary for multiclass Phase-1 output

Dataset	Missing cells	Missing %	Duplicates	Raw imbalance	Train imbalance after	Balancer
N-BaIoT	0	0.000%	11,495	2.05	1.00	SMOTE
CICIoT2023	96	0.001%	20,229	30,626.00	17,382.00	SMOTE
BoT-IoT	1,200,404	16.222%	0	162.80	1.00	SMOTE

Table 7 reports missing cells, missing percentage, duplicate records, raw imbalance, and post-balancing imbalance. N-BaIoT has no missing cells but contains 11,495 duplicate records before cleaning. CICIoT2023 has very few missing cells but shows severe raw class imbalance. BoT-IoT has a high missing-cell count and strong raw imbalance, making preprocessing especially important. The table shows how the balancing process reduces training imbalance for most settings. It also reveals that multiclass CICIoT2023 remains highly skewed even after the reported balancing process. This table supports the discussion on data quality, class imbalance, and realistic IDS evaluation difficulty.

8. Results

This section reports the completed Phase-1 preprocessing results and the verified Phase-2 individual dataset model-training results. Phase-1 outputs provide data quality analysis, binary/multiclass comparisons, feature metadata, client split summaries, VAE-ready arrays, and figures. Phase-2 outputs provide trained PyTorch hybrid models, XGBoost classifiers, deep-feature matrices, prediction files, explainability artifacts, and a metrics table for N-BaIoT, CICIoT2023, and BoT-IoT.

Table 8. Binary versus multiclass Phase-1 comparison

Experiment	Dataset	Clean	Train	Val.	Test	Classes	Features	Raw imb.	Train imb.
Multiclass	N-BaIoT	188,505	209,010	28,276	28,276	5	13	2.05	1.00
Multiclass	CICIoT2023	179,766	125,836	26,965	26,965	35	31	30,626.00	17,382.00
Multiclass	BoT-IoT	200,000	278,288	30,001	30,000	2	42	162.80	1.00
Binary	N-BaIoT	188,505	207,352	28,276	28,276	2	12	2.05	1.00
Binary	CICIoT2023	179,766	245,068	26,965	26,965	2	31	30,626.00	1.00
Binary	BoT-IoT	200,000	278,288	30,001	30,000	2	42	162.80	1.00

Table 8 compares binary and multiclass outputs across N-BaIoT, CICIoT2023, and BoT-IoT. It lists clean rows, train/validation/test sizes, class counts, selected features, and imbalance ratios. The table shows that binary conversion reduces the label space to two classes for all datasets. Multiclass CICIoT2023 remains the most complex setting because it contains 35 classes. N-BaIoT shows different feature counts between binary and multiclass modes. BoT-IoT keeps the same selected feature count in both binary and multiclass settings. This table helps justify why binary and multiclass IDS results should be evaluated separately.

Table 9. Phase-2 individual dataset model results using verified training outputs

Model	Setting	Accuracy	Precision	Recall	F1-score	ROC-AUC
Hybrid CNN-BiLSTM-Attention-Tiny Transformer	Individual datasets; macro average	0.8980	0.8257	0.8434	0.8288	0.9995
Hybrid deep features + XGBoost	Individual datasets; macro average	0.9211	0.8724	0.8576	0.8576	1.0000
CNN	Individual datasets; macro average	0.8954	0.8033	0.8286	0.8048	0.9996
LSTM	Individual datasets; macro average	0.8994	0.7961	0.8178	0.7966	0.9998
BiLSTM	Individual datasets; macro average	0.8955	0.7785	0.8190	0.7875	0.9997
Random Forest	Individual datasets; macro average	0.9174	0.8721	0.8624	0.8614	1.0000
XGBoost	Individual datasets; macro average	0.9246	0.8918	0.8666	0.8692	1.0000

Table 9 reports the verified Phase-2 model results from the individual dataset training workflow. The values summarize macro-averaged performance across N-BaIoT, CICIoT2023, and BoT-IoT for models that were executed in Phase-2. XGBoost on selected Phase-1 features achieved the strongest macro-averaged F1-score, while the hybrid deep-feature XGBoost model performed best for BoT-IoT. CICIoT2023 remained the most difficult dataset because its multiclass setting contains 35 classes and severe class imbalance. ROC-AUC entries marked with an asterisk are averaged over available ROC-AUC values because the exported multiclass CICIoT2023 ROC-AUC field was blank.

8.1 Paper-Ready Phase-1 Figures

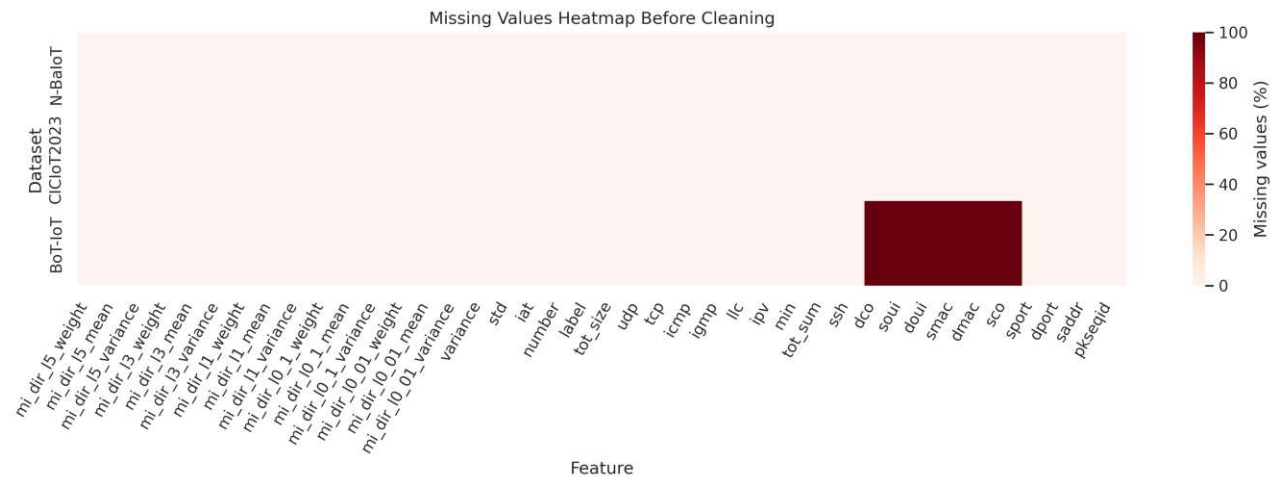


Figure 4(a). Missing-value heatmap

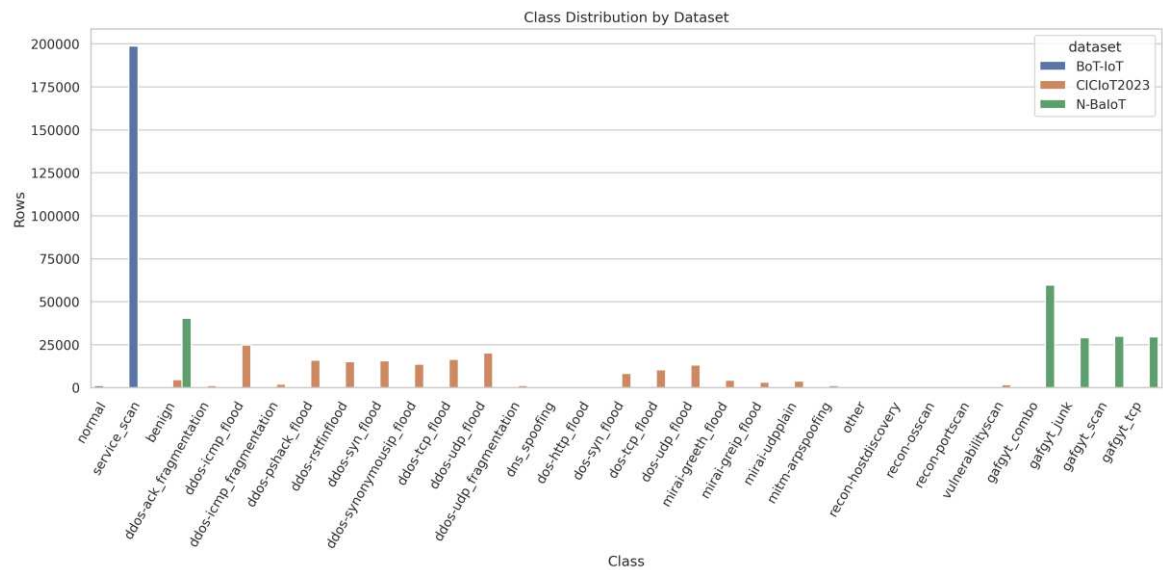


Figure 4(b). Class distribution by dataset

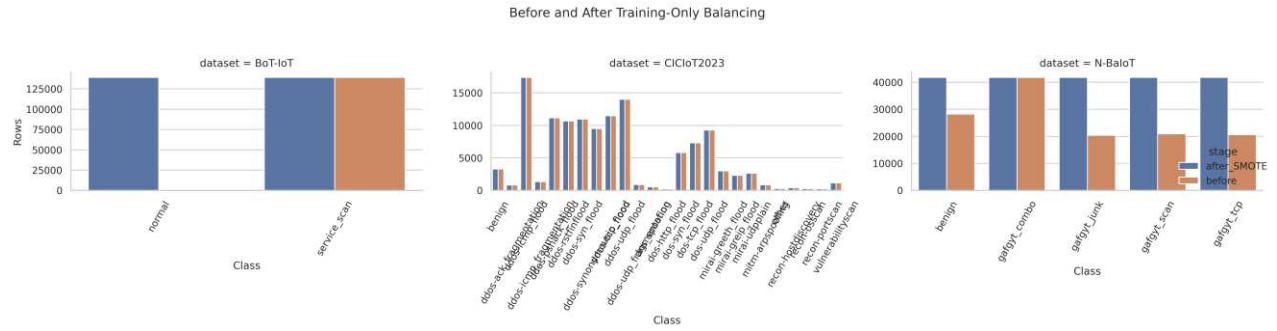


Figure 4(c). Before/after training balancing

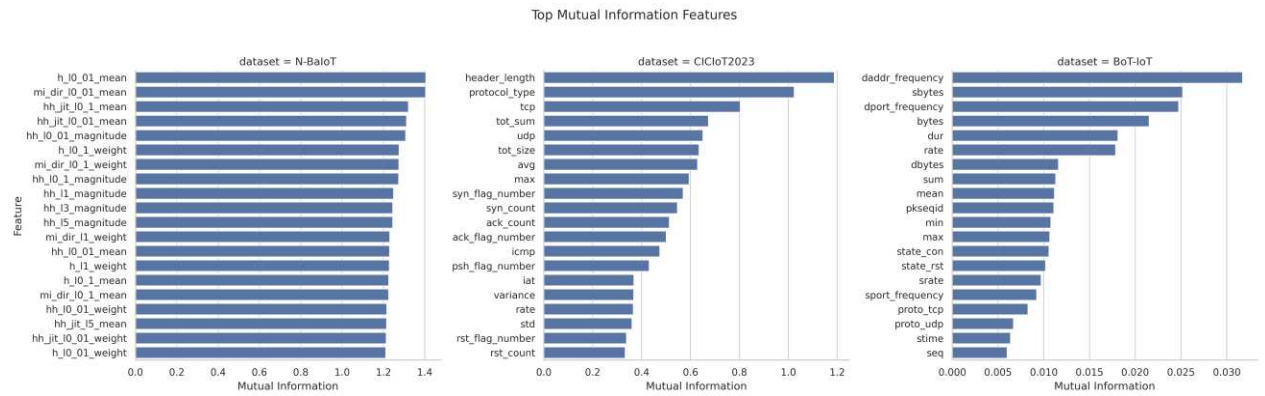


Figure 4(d). Mutual-information feature importance

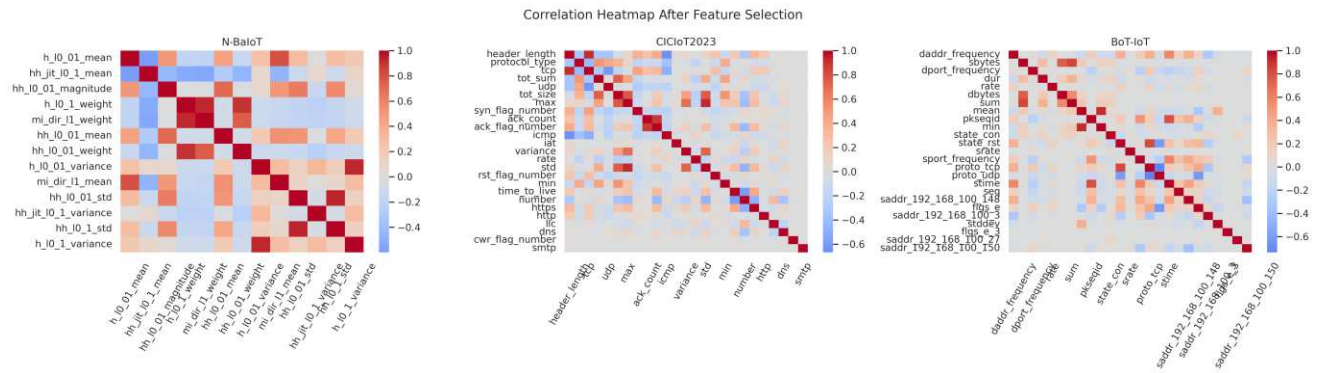


Figure 4(e). Correlation heatmap

Dataset Comparison Summary

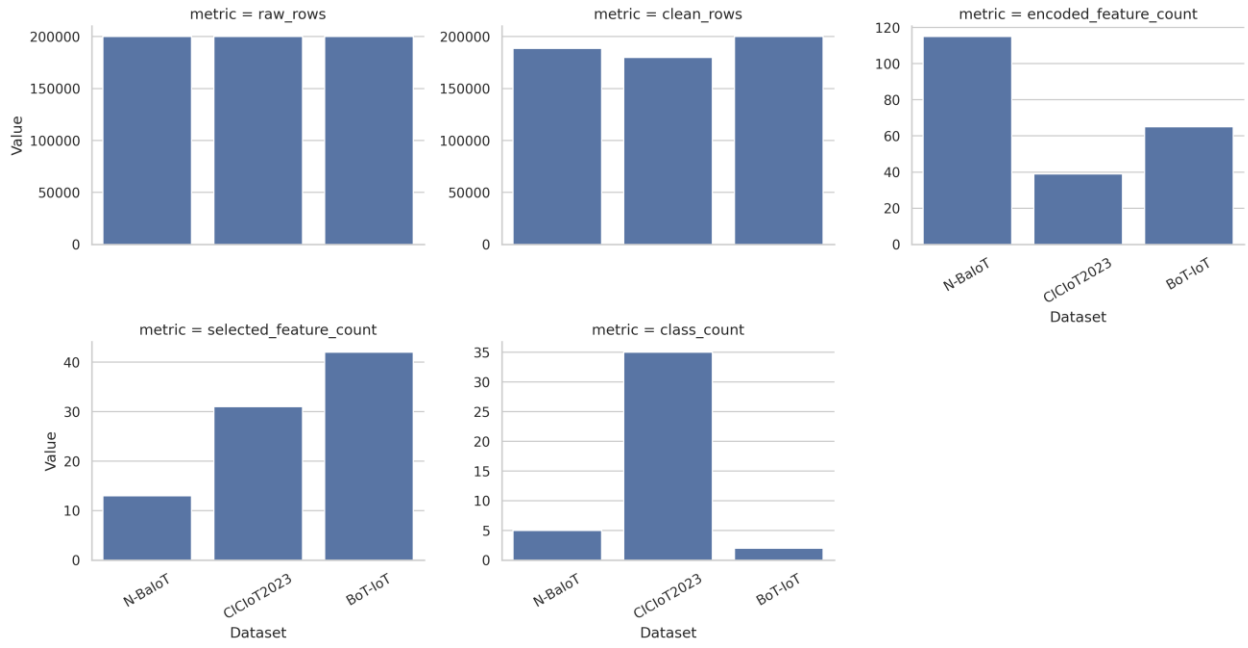


Figure 4(f). Dataset comparison bar chart

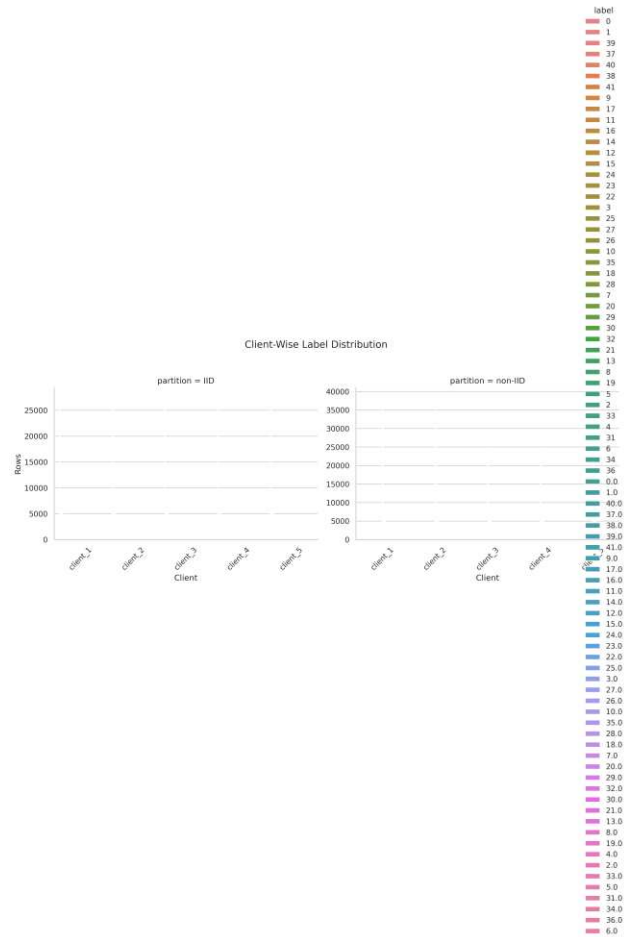


Figure 4(g). Client-wise label distribution

Figure 4(a-g) summarizes the key Phase-1 visual outputs used to validate the proposed IoT intrusion detection preprocessing pipeline. The missing-value heatmap shows dataset quality issues and supports the need for cleaning before feature selection and model training. The class-distribution plot highlights severe imbalance across N-BaIoT, CICIoT2023, and BoT-IoT, showing why accuracy alone is insufficient for IDS evaluation. The before/after balancing figure demonstrates how SMOTE-based resampling reduces minority-class starvation while keeping validation and test data untouched. The mutual-information feature-importance plot identifies the most informative traffic features for attack detection and supports dimensionality reduction. The correlation heatmap reveals redundant feature relationships and justifies correlation-based filtering to improve model efficiency. The dataset comparison chart emphasizes heterogeneity in rows, features, classes, and imbalance across the three datasets. Finally, the client-wise label distribution connects the cleaned Phase-1 outputs to federated learning by showing IID and non-IID client label patterns, motivating FedProx-based Phase-2 training.

8.2 Binary versus Multiclass Figures

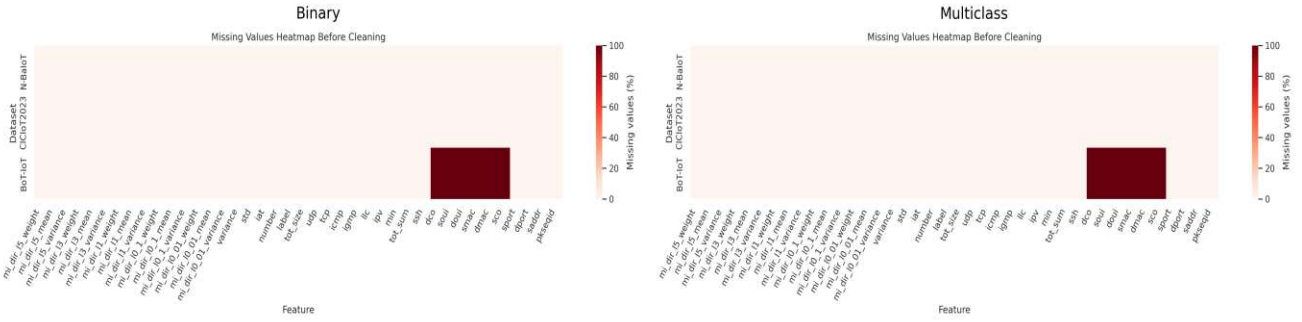


Figure 5(a). Binary/multiclass missing values

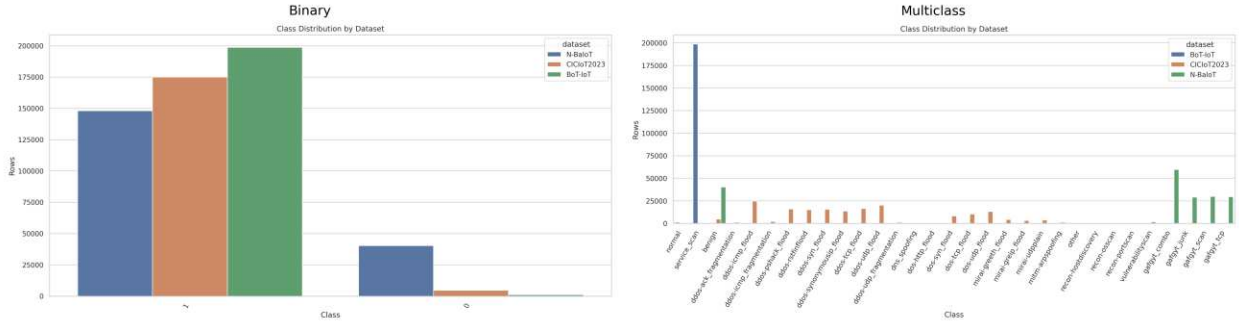


Figure 5(b). Binary/multiclass class distribution

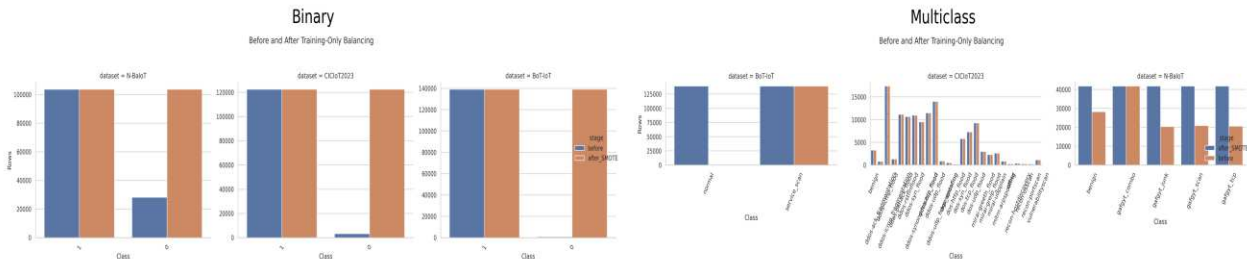


Figure 5(c). Binary/multiclass balancing

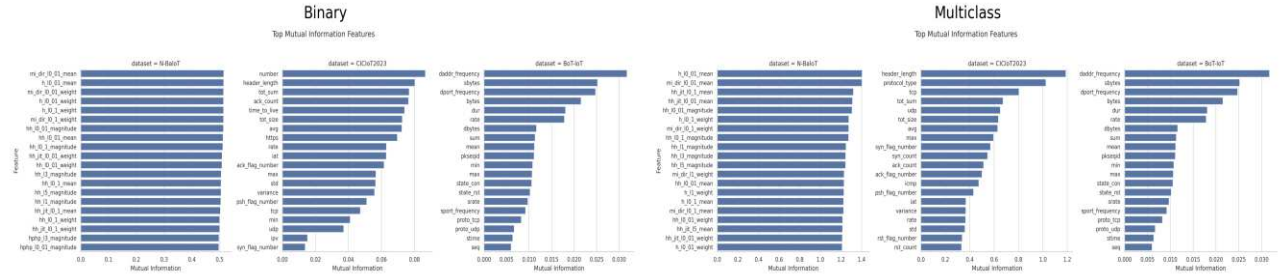


Figure 5(d). Binary/multiclass feature importance

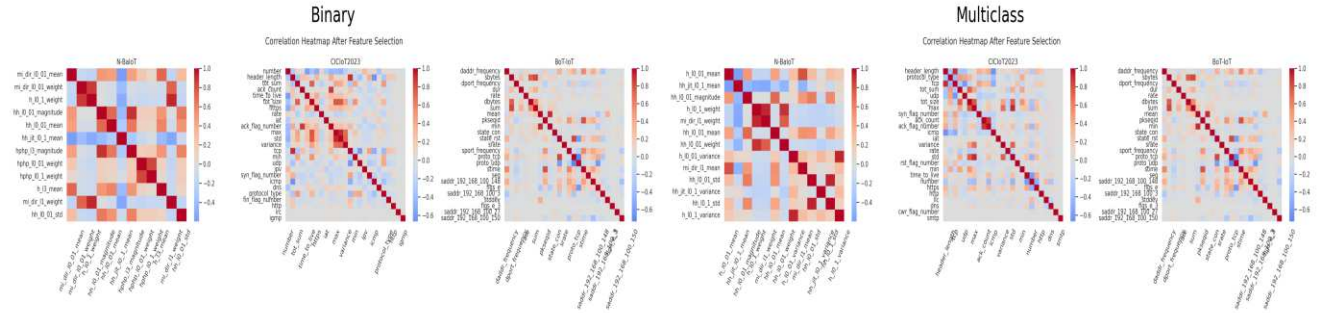


Figure 5(e). Binary/multiclass correlation heatmap

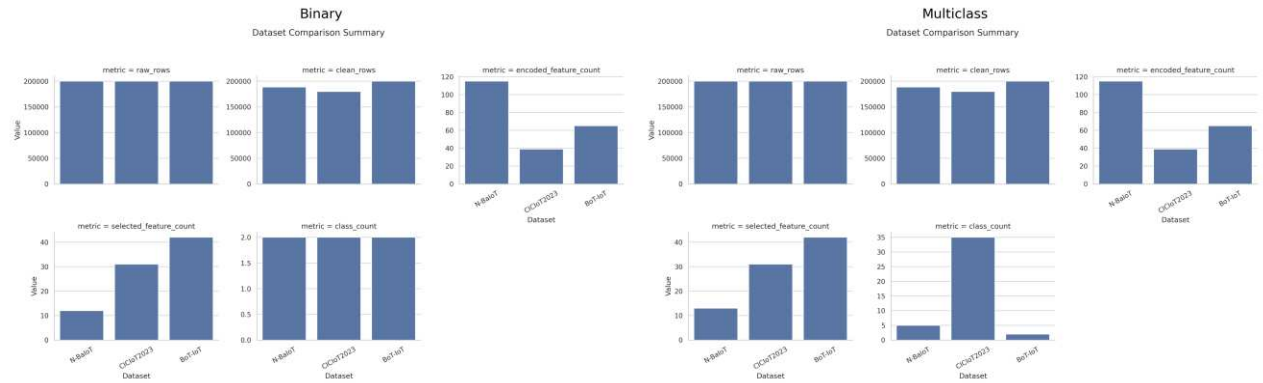


Figure 5(f). Binary/multiclass dataset comparison

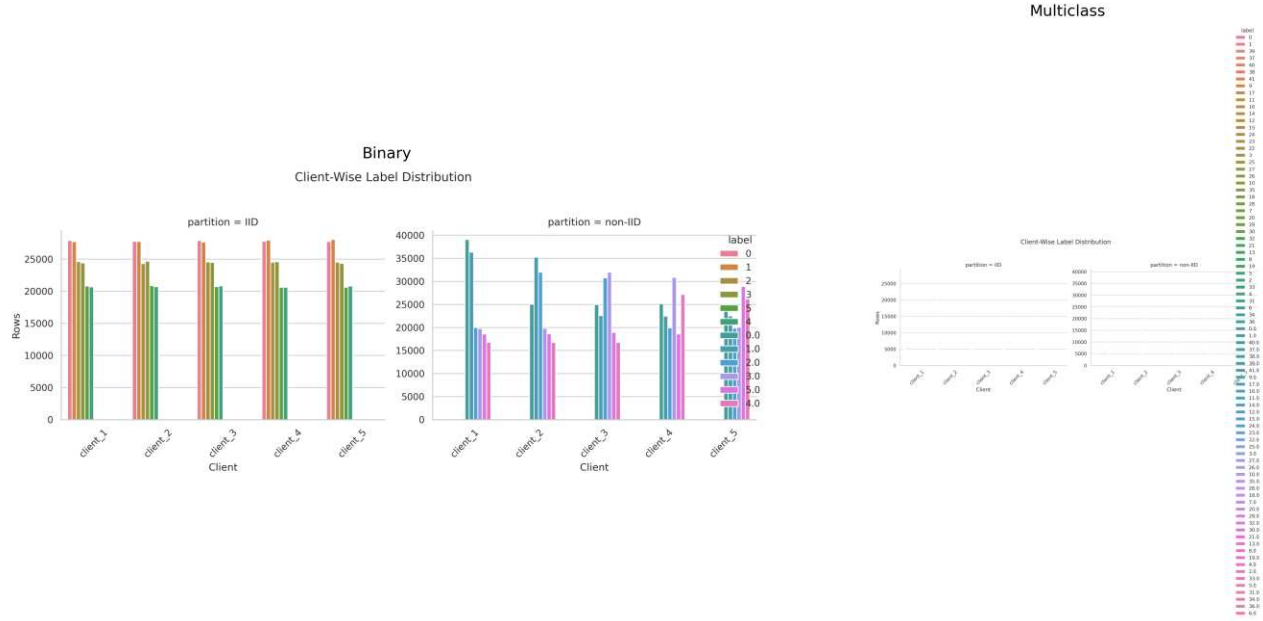


Figure 5(g). Binary/multiclass client labels

Figure 5(a-g) summarizes the binary-versus-multiclass comparison used to validate the Phase-1 preprocessing outputs. The missing-value comparison shows that both task formulations are generated from cleaned datasets and helps confirm that label conversion does not introduce new data-quality issues. The class-distribution plots demonstrate how multiclass attack labels collapse into benign-versus-attack categories in the binary setting. This explains why binary IDS is operationally simpler, while multiclass IDS preserves richer attack-family information but introduces greater imbalance and classification difficulty. The balancing plots show how resampling affects the training data under both task modes and support the use of training-only SMOTE-family methods to avoid validation/test leakage. Feature-importance comparisons reveal that the most relevant traffic variables can differ between attack detection and attack-type identification. Correlation heatmaps further justify redundancy removal and show that feature relationships must be checked separately for binary and multiclass modes. The dataset comparison chart highlights differences in clean rows, feature counts, class counts, and imbalance across all datasets and task settings. Finally, the client-label distribution figure links these outputs to federated learning by showing how IID and non-IID client skew changes under binary and multiclass formulations, supporting client-wise Phase-2 evaluation.

8.3 Client Partition Summary

Table 10 reports IID and non-IID client partitions used for federated learning experiments. It lists five clients for IID distribution and five clients for non-IID distribution. The IID rows show nearly balanced client sizes and similar dominant label counts across clients. The non-IID rows show different client sizes and more skewed label distributions.

Table 10. Client partition summary showing top labels per client

Split	Client	Rows	Dominant label counts
IID	Client 1	122,627	0:27,818, 1:27,796, 39:8,546, 37:8,415
IID	Client 2	122,627	0:27,997, 1:27,614, 41:8,515, 40:8,465

IID	Client 3	122,627	1:28,035, 0:27,869, 38:8,455, 41:8,325
IID	Client 4	122,627	1:27,871, 0:27,603, 39:8,429, 41:8,403
IID	Client 5	122,626	0:27,857, 1:27,828, 37:8,457, 39:8,384
NON_IID	Client 1	142,259	0.0:39,161, 1.0:36,327, 40.0:8,423, 37.0:8,396
NON_IID	Client 2	126,953	1.0:34,921, 0.0:25,204, 38.0:8,400, 39.0:8,375
NON_IID	Client 3	114,760	0.0:25,086, 1.0:22,674, 37.0:8,428, 39.0:8,425
NON_IID	Client 4	114,729	0.0:24,871, 1.0:22,745, 39.0:8,413, 41.0:8,387
NON_IID	Client 5	114,433	0.0:24,822, 1.0:22,477, 38.0:8,441, 37.0:8,431

Table 10 demonstrates how the project simulates realistic distributed IoT environments. It supports later FedProx evaluation because non-IID client skew can destabilize federated training. The table is important for interpreting client-wise accuracy, fairness, and robustness in Phase-2.

8.4 Planned Ablation and Comparison Protocol

The current folders prepare the inputs for this analysis, but the actual model metrics must be inserted after Phase-2 execution. The ablation plan below is included so that the final paper can be completed without changing the experimental narrative.

Table 11. Recommended Phase-2 ablation plan

Ablation	Comparison	Purpose
Feature selection	All encoded features vs selected features	Shows whether variance, mutual information, and correlation filtering improve efficiency and generalization.
Balancing	Unbalanced train vs SMOTE vs BorderlineSMOTE vs ADASYN	Quantifies minority-class benefit and checks whether synthetic sampling harms majority-class precision.
Model family	XGBoost vs CNN vs BiLSTM-attention vs Tiny Transformer	Separates tabular baseline performance from deep sequential/self-attention gains.
Anomaly branch	Classifier only vs classifier plus VAE anomaly score	Tests whether reconstruction evidence improves early or uncertain attack detection.
Federated setting	Centralized vs FedAvg vs FedProx	Measures the cost and benefit of privacy-preserving decentralized training.
Client distribution	IID vs non-IID	Evaluates robustness under realistic site/device label skew.
Task formulation	Binary vs multiclass	Compares operational detection with attack-family identification.

Table 11 outlines additional ablation experiments recommended for the remaining federated and anomaly-extension stages. The completed Phase-2 run already evaluates the individual CNN, LSTM, BiLSTM, Random Forest, XGBoost, hybrid deep model, and hybrid deep-feature XGBoost variants.

9. Discussion

The Phase-1 results show that dataset heterogeneity is not cosmetic. N-BaIoT has high-dimensional statistical traffic features and botnet-family labels. CICIoT2023 includes a much wider attack vocabulary and extreme class imbalance, with many rare web and reconnaissance labels. BoT-IoT in the provided project sample is dominated by service_scan relative to normal traffic. These differences justify the project decision to process datasets separately, retain dataset-level metadata, and avoid a single universal label map.

Binary detection is operationally useful because it answers the immediate question of whether traffic should be treated as benign or malicious. Multiclass detection is analytically richer because it identifies attack family or type, enabling response prioritization and threat reporting. The binary outputs generally reduce label sparsity and simplify class boundaries, whereas the multiclass outputs preserve security semantics at the cost of more severe imbalance and more difficult minority-class learning.

Feature selection reduces the modeling burden and improves interpretability. N-BaIoT was reduced to 13 selected multiclass features, CICIoT2023 to 31, and BoT-IoT to 42. This makes the proposed federated setting more practical because clients exchange model parameters trained on compact representations rather than high-dimensional raw feature spaces. The selected features also support a VAE anomaly module because reconstruction loss is easier to stabilize when redundant and near-constant dimensions are reduced.

The largest methodological risk after Phase-2 execution is over-generalizing the individual-dataset results to full federated deployment. The completed Phase-2 results demonstrate supervised detection performance for N-BaIoT, CICIoT2023, and BoT-IoT, including hybrid deep-feature extraction and XGBoost classification. However, the VAE anomaly module and FedProx global aggregation still require separate execution before federated anomaly and global-model superiority claims are made. This distinction keeps the paper's detection claims grounded in verified Phase-2 metrics while preserving a clear roadmap for the remaining federated experiments.

10. Conclusion

This manuscript presents a federated hybrid deep learning framework for IoT intrusion detection using N-BaIoT, CICIoT2023, and BoT-IoT. The completed pipeline cleans and encodes heterogeneous datasets, constructs binary and multiclass label settings, performs feature selection, applies training-only balancing, exports VAE-ready arrays, creates IID/non-IID client partitions, and executes Phase-2 individual dataset model training. The Phase-2 results show that XGBoost and hybrid deep-feature XGBoost provide strong detection performance, with N-BaIoT and BoT-IoT achieving near-perfect scores and CICIoT2023 remaining the hardest multiclass benchmark. These findings support the effectiveness of the preprocessing and individual model-training stages while motivating the next FedProx and VAE anomaly experiments.

11. Future Scope

- Execute Phase-2 centralized and federated experiments for CNN, BiLSTM-attention, Tiny Transformer, XGBoost, VAE, and FedProx models.
- Add explainable AI methods such as SHAP or attention visualization to justify prevention decisions.

- Deploy the trained models on edge or gateway hardware and measure latency, memory, energy, and throughput.
- Extend the prevention layer with adaptive thresholds, rate-limiting policies, and analyst feedback.
- Investigate blockchain-secured federated learning, secure aggregation, differential privacy, and TinyML deployment for constrained IoT networks. Federated-learning threat surveys also indicate that poisoning, inference, and privacy attacks should be evaluated before real deployment [31].

12. Project Artifact Inventory

Table 12. Project artifacts used in this manuscript

Artifact group	Use in document
Multiclass outputs	CSV splits, reports, metadata, scalers, encoders, VAE arrays, client splits, figures
Binary outputs	Binary train/validation/test files, reports, client splits, VAE arrays, balancing variants
Binary/multiclass figures	Comparison CSV and seven comparison figures
Project diagram	Framework block diagram
Project flowchart	End-to-end project flowchart
Mathematical diagram	Equation-level framework formulation

Table 12 lists the project artifacts used as evidence and inputs for the manuscript. It includes multiclass outputs such as CSV splits, reports, metadata, scalers, encoders, VAE arrays, and figures. It includes binary outputs such as binary splits, client partitions, VAE arrays, and balancing variants. It records the binary-versus-multiclass comparison figures and comparison CSV. It also lists the project block diagram, flowchart, and mathematical formulation diagram. The table improves traceability by connecting manuscript claims to the files used to generate them. It supports reproducibility because future researchers can identify which artifacts correspond to each paper section.

Declarations

Ethics approval and consent to participate

Not applicable. This study does not involve human participants, human data, or animals.

Consent for publication

Not applicable.

Availability of data and materials

The study uses publicly available benchmark datasets, including N-BaIoT, CICIOT2023, and BoT-IoT. Project-derived preprocessing outputs and analysis artifacts are available from the corresponding author upon reasonable request.

Funding

No funding was received for this study.

Authors' contributions

S.M. conceptualized the study, designed the methodology, processed the datasets, analyzed the Phase-1 outputs, prepared the figures and tables, and wrote the original manuscript draft.

R.V.S.R. supervised the work, reviewed the methodology, provided technical guidance, edited the manuscript, and approved the final version. All authors read and approved the final manuscript.

Acknowledgements

Not applicable.

References

- [1] S. D. R. F. A. Z. R. L. a. A. G. Euclides Carlos Pinto Neto, "CICIoT2023: A Real-Time Dataset and Benchmark for Large-Scale Attacks in IoT Environment," *Sensors*, doi: 10.3390/s23135941, vol. 23, no. 13, p. 5941, 2023.
- [2] R. B. Van Tuan Nguyen, "FedMSE: Semi-supervised federated learning approach for IoT network intrusion detection," *Computers & Security Computers & Security*, vol. 151, p. 104337, 2024.
- [3] N. M. E. S. a. B. T. Nickolaos Koroniotis, "Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset," in *Elsevier*, 2018.
- [4] M. H. Z. M. A. H. S. K. R. M. a. F. S. Syed Muhammad Salman Bukhari, "Enhancing cybersecurity in Edge IIoT networks: An asynchronous federated learning approach with a deep hybrid detection model.," *Internet of Things*, vol. 27, p. 101252, 2024.
- [5] N. M. Z. T. A. M. a. A. A. Abdullah Alsaedi, "TON_IoT Telemetry Dataset: A New Generation Dataset of IoT and IIoT for Data-Driven Intrusion Detection Systems," in *IEEE Access*, 2020.
- [6] I. G. P. V. a. J. K. Ansam Khraisat, "Survey of intrusion detection systems: techniques, datasets and challenges," *Cybersecur, Springer Nature Link*, vol. 2, p. 20, 2019.
- [7] M. A. K. P. S. P. P. A. A.-N. a. S. V. R. Vinayakumar, "Deep Learning Approach for Intelligent Intrusion Detection System," *IEEE Access*, vol. 1, no. 1, p. 99, 2019.
- [8] L. M. S. M. a. H. J. Mohamed Amine Ferrag, "Deep learning for cyber security intrusion detection: : Approaches, datasets, and comparative study," *Journal of Information Security and Applications*, vol. 50, no. C, p. 102419, 2020.
- [9] Y. L. T. C. a. Y. T. Qiang Yang, "Federated Machine Learning: Concept and Applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, pp. 1-19, 2019.
- [10] A. K. S. M. Z. M. S. A. T. a. V. S. Tian Li, "Federated Optimization in Heterogeneous Networks," *Proceedings of Machine Learning and Systems 2 (MLSys 2020)*, 2020.

- [11] H. B. M. B. A. A. B. M. B. A. N. B. K. B. Z. C. G. C. R. C. R. G. D. H. E. S. E. R. D. E. J. G. Peter Kairouz, "Advances and Open Problems in Federated Learning," *Foundations and Trends in Machine Learning*, vol. 14, no. 1-2, pp. 1-210, 2019.
- [12] M. D. P. N. P. A. S. J. L. a. H. V. P. Dinh C. Nguyen, "Federated Learning for Internet of Things:," *IEEE COMMUNICATIONS SURVEYS & TUTORIALS*, vol. 23, no. 3, pp. 1622-1658, 2021.
- [13] S. S. O. A. G. Y. K. P. S. B. P. K. R. M. a. T. R. G. Shaashwat Agrawal, "Federated Learning for Intrusion Detection System: Concepts, Challenges and Future Directions," *JNCA, Elsevier*, 2021.
- [14] S. G. a. M. A. A. Ghazaleh Shirvani, "Enhancing IoT Security Against DDoS Attacks through Federated Learning," *arXiv*, 2024.
- [15] P. S. K. N. A. S. a. H. W. Shihua Sun, "FedMADE: Robust Federated Learning for Intrusion Detection in IoT Networks Using a Dynamic Aggregation Method," *The Information Security Conference (ISC)*, 2024.
- [16] W. Y. Z. C. J. F. D. N. a. K.-Y. L. Jiyuan Shen, "Effective Intrusion Detection in Heterogeneous Internet-of-Things Networks via Ensemble Knowledge Distillation-based Federated Learning," 2024.
- [17] a. J. K. Ahmed Bensaoud, "Optimized detection of cyber-attacks on IoT networks via hybrid deep learning models," *Ad Hoc Networks*, vol. 170, p. 103770, 2025.
- [18] S. M. V. O. R. P. A. C. T. a. A. K. D. KG Raghavendra Narayan, "IIDS: Design of Intelligent Intrusion Detection System for Internet-of-Things Applications," in *arXiv*, 2023.
- [19] A. A. K. N. E. K. a. Y. G.-D. Alaeddine Diaf, "Beyond Detection: Leveraging Large Language Models for Cyber Attack Prediction in IoT Networks," in *arXiv*, 2024.
- [20] M.-L. A. B. S. N. S. I. V. D. I. K. a. I. S. V. Ioannis Panopoulos, "Dynamic Temporal Positional Encodings for Early Intrusion Detection in IoT," in *10th International Conference on Smart and Sustainable Technologies (SpliTech 2025)*, 2025.
- [21] H. K. a. A. C. M. Afrah Gueriani, "Enhancing IoT Security with CNN and LSTM-Based Intrusion Detection Systems," in *ArXiv*, 2024.
- [22] M. N. a. P. T. HyeYoung Lee, "Spectral Feature Extraction for Robust Network Intrusion Detection Using MFCCs," in *ArXiv*, 2025.
- [23] H. K. a. A. C. M. Afrah Gueriani, "Hybrid ResNet-1D-BiGRU with Multi-Head Attention for Cyberattack Detection in Industrial IoT Environments," in *ArXiv*, 2026.
- [24] F. M. M.-M. a. J. L. R.-Á. Estela Sánchez-Carballo, "Latent Diffusion for Internet of Things Attack Data Generation in Intrusion Detection," in *IEEE*, 2026.
- [25] A. D. C. A. D. a. G. D. P. Ivan Letteri, "A Novel Resampling Technique for Imbalanced Dataset Optimization," in *ArXiv*, 2020.
- [26] Z. C. M. Z. Z. G. K. R. J. K. S. K. a. H. B. M. Sashank Reddi, "Adaptive Federated Optimization,"

in *Conference paper at ICLR 2021*, 2020.

- [27] T. T. A. M. X. Q. J. F.-M. Y. G. L. S. K. H. L. T. P. P. P. B. d. G. a. N. D. L. Daniel J. Beutel, "Flower: A Friendly Federated Learning Research Framework," in *Mobile-friendly Federated Learning framework*, 2020.
- [28] S. G. F. M. A. L. J. B. G. C. T. K. Z. L. N. G. L. A. A. D. A. K. E. Y. Z. D. M. R. A. T. S. C. Adam Paszke, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in *NeurIPS 2019*, 2019.
- [29] Z. C. Z. X. G. J. H. B. M. B. A. y. A. M. A.-S. G. A. S. A. K. D. D. D. S. D. H. E. A. G. Z. G. A. M. Jianyu Wang, "A Field Guide to Federated Optimization," in *ArXiv*, 2021.
- [30] R. R. R. M. P. a. F. S. MOHAMMED ALEDHARI, "Federated Learning: A Survey on Enabling Technologies, Protocols, and Applications," in *IEEE Access*, 2020.
- [31] H. Y. a. Q. Y. Lingjuan Lyu, "Threats to Federated Learning: A Survey," 2020.
- [32] A. K. S. A. T. a. V. S. Tian Li, "Federated Learning: Challenges, Methods, and Future Directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50-60, 2020.