Privacy-Aware Multi-Class Intrusion Detection Using Federated Neural Encoding and Robust Feature Distillation

*Note: Sub-titles are not captured in Xplore and should not be used

1st Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

2nd Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

3rd Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

***Abstract*—The exponential proliferation of interconnected IoT devices has dramatically increased the complexity of cybersecurity, creating vast surfaces vulnerable to distributed attacks. To address the need for both data privacy and accurate threat detection, this work presents a scalable, privacy-preserving deep learning framework for multi-class intrusion detection in IoT networks. Leveraging federated learning, the proposed system enables decentralized model training on edge devices without transferring raw data, ensuring privacy while benefiting from collaborative intelligence. Using the UNB CIC IoT 2023 dataset, which includes diverse real-world attack scenarios and benign traffic, a fully connected neural network is trained locally on each node, with global model updates synchronized via federated averaging. Rigorous preprocessing and feature normalization support improved convergence and generalization. Experimental results demonstrate strong classification performance with a test accuracy of 95.06%, effectively identifying both common and underrepresented attack types. This approach offers a robust, trustworthy solution for intrusion detection in smart city infrastructures, combining federated neural encoding and deep feature distillation to advance secure edge intelligence.**

***Index Terms*—component, formatting, style, styling, insert**

I. Introduction

The rapid growth of Internet of Things (IoT) deployments, particularly within smart city ecosystems, has revolutionized data-driven automation and pervasive connectivity. However, this hyper-connectivity has simultaneously expanded the attack surface, enabling a new breed of large-scale, multi-vector cyber threats that traditional centralized security frameworks are ill-equipped to handle. Intrusion Detection Systems (IDS), while critical to safeguarding IoT infrastructure, face substantial challenges due to the heterogeneity of devices, the limited computational capacity at the edge, and the pressing need for user privacy in handling sensitive network data [1].

Conventional IDS approaches often rely on centralized data aggregation, where raw traffic logs are collected and analyzed at cloud or core nodes. While effective in high-resource environments, these methods raise serious privacy concerns and introduce latency, bottlenecks, and single points of failure. Furthermore, static rule-based or shallow machine learning models employed in legacy systems struggle with the complexity and adaptability of modern attacks, especially in the presence of class imbalance and evolving threat signatures common in IoT environments [2].

To overcome these limitations, the research community has turned to Federated Learning (FL)—a decentralized paradigm in which local models are trained at the edge and only model updates, rather than raw data, are shared with a global aggregator. This privacy-aware mechanism aligns with emerging data governance frameworks while enabling continuous learning from diverse, distributed sources. However, the non-iid nature of IoT data, communication overhead, and the risk of degraded global performance due to heterogeneous local models remain persistent challenges [3].

In this context, we propose a novel intrusion detection framework that integrates federated neural encoding with robust deep feature distillation to enhance both accuracy and privacy preservation. By training fully connected neural networks locally on IoT nodes and harmonizing global updates via the Federated Averaging algorithm, the system effectively learns discriminative representations of both frequent and rare attack types. To mitigate data variance and support convergence across nodes, a standardized preprocessing pipeline involving normalization and label encoding is introduced.

Our approach is empirically validated using the UNB CIC IoT 2023 dataset, which contains a rich set of network behaviors captured across 100+ IoT devices under seven major attack categories, including DDoS, spoofing, reconnaissance, and botnet propagation. The proposed system achieves a test accuracy of 95.06%, significantly outperforming baseline models in both precision and recall across minority classes.

This work contributes a scalable, privacy-preserving IDS architecture tailored to the constraints and dynamics of smart IoT deployments. By fusing federated neural encoding with deep feature compression strategies, the framework promotes

Identify applicable funding agency here. If none, delete this.

secure edge intelligence without compromising detection fidelity, paving the way for next-generation cyber-resilient smart cities. In this paper, **Section II** provides a comprehensive review of existing intrusion detection approaches in IoT environments, with a focus on both traditional centralized architectures and contemporary deep learning-based frameworks. It emphasizes the challenges of privacy preservation, data imbalance, and deployment scalability. **Section III** introduces the proposed privacy-aware federated learning framework, detailing the selection and preprocessing of the UNB CIC IoT 2023 dataset, the architecture of the fully connected neural network, and the federated training workflow based on the Federated Averaging algorithm. **Section IV** presents experimental evaluations, including class-wise performance metrics, confusion matrix analysis, and comparisons with baseline models, demonstrating the efficacy of the proposed method in identifying both prevalent and minority-class attacks. Finally, **Section V** concludes the paper with a summary of key findings and discusses potential future work, including integration with real-time alert systems and extension to cross-domain intrusion detection in heterogeneous smart city infrastructures.

A. EXISTING INTRUSION DETECTION TECHNIQUES IN IOT

Intrusion detection within IoT ecosystems remains a critical area of research due to the increased vulnerability posed by resource-constrained, heterogeneous, and widely distributed devices. Various approaches have emerged in the literature, which can be broadly categorized into rule-based systems, shallow learning models, deep learning architectures, federated intrusion frameworks, and adversarially robust detection strategies.

1) Signature-Based and Rule-Driven Methods:

Early intrusion detection systems relied heavily on predefined signatures and rule sets. Techniques such as Snort and Bro/Zeek operate by matching incoming traffic against known attack patterns. While efficient in identifying previously seen threats, these methods struggle with zero-day attacks, evolving adversaries, and context adaptation in heterogeneous IoT environments. Furthermore, signature updates require centralized data curation, raising concerns about timeliness and privacy leakage.

2) Shallow Machine Learning Approaches:

A shift toward statistical learning saw the adoption of supervised algorithms like Decision Trees, k-Nearest Neighbors (k-NN), Support Vector Machines (SVMs), and Naive Bayes classifiers. These models often use manually engineered features derived from traffic metadata—such as packet length, protocol flags, and flow duration. Although lightweight and interpretable, shallow models are limited in their ability to capture temporal dependencies or high-dimensional patterns and often require extensive feature selection to generalize effectively across different attack vectors [4].

3) Deep Learning-Based IDS Models:

With the rise of computationally capable edge and fog nodes, deep neural networks have been leveraged for automated feature extraction and temporal analysis. Recurrent Neural Networks (RNNs), Long Short-Term Memory networks (LSTMs), and Convolutional Neural Networks (CNNs) have been utilized to model the sequential and spatial structure of network traffic. Tang et al. [5] demonstrated the use of CNN-LSTM hybrids for enhanced detection accuracy, while Alsaedi et al. [6] explored stacked autoencoders for anomaly detection. Despite their success, such models often depend on centralized training, raising privacy issues and requiring large, labeled datasets.

4) Federated Learning-Based Detection:

Recent advancements have introduced Federated Learning (FL) as a privacy-preserving alternative to traditional centralized training. In FL, edge nodes train local models on private data and share only model updates with a global server for aggregation. This approach preserves data locality and minimizes exposure of sensitive IoT traffic. Research by Truong et al. [7] and Nguyen et al. [8] has shown promising results using federated CNNs and LSTMs for anomaly detection. However, challenges persist in dealing with non-IID data, device heterogeneity, and communication overhead.

5) Adversarial Robustness and Defense Mechanisms:

Adversarial machine learning has exposed vulnerabilities in many intrusion detection models, where crafted perturbations can evade detection. To mitigate this, robust IDS designs have adopted defensive training strategies such as adversarial data augmentation, ensemble learning, and robust optimization. Zhang et al. [9] proposed adversarial training for RNN-based IDS to resist evasion attacks, while Li et al. [10] introduced randomized smoothing to enhance resilience. These techniques improve security but may reduce accuracy and increase model complexity.

B. Limitations of Existing Methods

While intrusion detection in IoT networks has witnessed substantial progress, existing methodologies continue to exhibit critical limitations that hinder their effectiveness in real world smart city environments. Chief among these challenges is the issue of generalization. Models often perform well on controlled benchmark datasets but show significant degradation when applied to traffic from unseen IoT devices, novel attack types, or networks with varying topology and behavior. This is largely due to overfitting to specific data distributions, insufficient representation of rare attack categories, and limited device or protocol diversity during training.

Furthermore, many legacy approaches rely on handcrafted features or shallow learning models that fail to capture the intricate temporal and spatial dependencies present in network flows. Although deep learning models such as CNNs and LSTMs offer superior representational power, their effectiveness is often curtailed by the need for centralized data aggregation, which raises privacy concerns and increases vulnerability to single points of failure.

Resource constraints inherent to IoT devices also pose major obstacles. Complex deep learning models generally demand

high computational power, memory, and continuous connectivity, which are infeasible for edge level deployment in many real world IoT scenarios. Attempts to offload computation to the cloud reintroduce latency, scalability, and privacy tradeoffs.

Temporal modeling remains underutilized or oversimplified in many intrusion detection frameworks. While RNN based models can encode sequential patterns, they often suffer from high inference time and are sensitive to input variability. CNN based detectors, when applied to traffic snapshots or aggregated flow features, typically disregard temporal evolution, making them vulnerable to stealthy or slow evolving threats.

Another persistent challenge is the lack of explainability in detection outcomes. Most deep learning based IDSs operate as opaque black boxes, offering limited insight into the rationale behind classification decisions, an issue that reduces trustworthiness in mission critical or legally sensitive environments. Additionally, the binary classification paradigm adopted by many systems fails to reflect the nuanced threat landscape of IoT networks, which may involve overlapping or multistage attacks that require multi class or probabilistic reasoning.

The adversarial threat model further exacerbates these issues. Attackers can craft network traffic patterns that exploit model blind spots, evade detection, or poison training data in federated settings. Existing models often lack the robustness to adapt to these dynamic threats, especially in federated deployments where client data heterogeneity and asynchronous updates add layers of complexity.

These limitations collectively underscore the need for a new class of intrusion detection frameworks that offer enhanced generalization, edge level deployability, temporal awareness, explainability, and resistance to adversarial manipulation. The method proposed in this paper addresses these gaps by combining privacy preserving federated learning with robust deep neural encoding and feature distillation techniques, enabling accurate and trustworthy multi class intrusion detection in diverse IoT environments.

II. PROPOSED METHODOLOGY

A. Dataset Description

This study employs the UNB CIC IoT 2023 dataset, a recent and comprehensive benchmark for evaluating intrusion detection systems in diverse Internet of Things environments. Compiled by the Canadian Institute for Cybersecurity, the dataset captures real network traffic from 105 heterogeneous IoT devices exposed to 33 different cyberattack scenarios, categorized into seven primary threat classes: Denial of Service, Distributed Denial of Service, Reconnaissance, Web based, Brute force, Spoofing, and the Mirai botnet. The dataset reflects modern adversarial behaviors within smart home and enterprise network contexts.

Each network session is processed into structured CSV records using CICFlowMeter, yielding over 100 numerical features such as flow duration, byte counts, header statistics, packet inter arrival times, and entropy metrics. The dataset provides a balanced mixture of benign and malicious traffic, supporting both binary and multi class classification. In this work, multi class classification is prioritized to capture the granularity of attack behavior across different categories.

For scalability in a federated setting, three representative partitions of the dataset are selected. Each partition comprises distinct subsets of attack classes and benign flows, simulating data observed at separate IoT gateways or edge devices. The combined dataset is split into training, validation, and test sets in a 64 percent, 16 percent, and 20 percent ratio, respectively, while maintaining consistent class distribution.

B. Data Preprocessing and Neural Encoding

To ensure standardized learning across distributed clients, a uniform preprocessing pipeline is implemented. All CSV files are merged and cleaned by removing missing values and redundant attributes. Only numerical features are retained to facilitate compatibility with neural models.

Attack labels are first integer encoded and subsequently transformed into one hot encoded vectors to support categorical cross entropy loss. Numerical features are scaled using Z score normalization to achieve mean zero and unit variance, accelerating model convergence and ensuring consistent feature scaling across partitions.

Following preprocessing, the data is divided into three non overlapping partitions, each representing a federated client. These clients hold non independent and identically distributed data, reflecting realistic deployment settings in which edge devices observe varying traffic patterns.

C. Federated Learning Framework

A federated learning approach is adopted to enable privacy preserving collaborative training without transferring raw data. Each client trains a local model on its data and shares only model weights with a central server for aggregation.

Each client hosts a deep neural network comprising three fully connected layers, ReLU activation functions, and Dropout for regularization. The final output layer employs softmax activation to support multi class predictions. The local training loop consists of receiving the global model, training for a fixed number of epochs, and returning updated weights to the server.

Training is performed using categorical cross entropy loss and optimized via the Adam optimizer with default parameters. Each client trains over three epochs with a batch size of 32 in each communication round. The server uses the Federated Averaging algorithm to aggregate weights from all clients, proportionally weighted by their dataset sizes, ensuring fairness and robustness under imbalanced client data conditions. The federated training process is implemented using the Flower framework.

D. Robust Feature Distillation and Global Aggregation

Following federated training, the global model is fine tuned centrally to distill knowledge aggregated from distributed clients. This fine tuning is conducted on the combined training set using early stopping based on validation loss and checkpointing to prevent overfitting.

The final model is evaluated on the holdout test set using standard metrics such as accuracy, precision, recall, F1 score, and confusion matrix analysis. A detailed classification report

is generated to highlight detection performance for each attack class.

To assess model generalization and interpretability, confidence scores and misclassification patterns are analyzed. This facilitates the identification of underperforming classes and supports future model refinements. The diversity of client partitions and threat scenarios enhances the robustness of the proposed system against novel and evolving IoT attack vectors.

A. Model Architecture

This section introduces a **hybrid neural architecture**, optimized for *privacy-sensitive*, *multi-class intrusion detection* in federated IoT environments. By combining **deep neural encoding**, **robust feature distillation**, and **federated fine-tuning**, the model achieves high detection accuracy while remaining computationally lightweight and interpretable. The full model pipeline is depicted in **Figure 1**.

1) **Input and Preprocessing**
   Each input is a Z-score–normalized numerical feature vector derived from network flow statistics (packet sizes, inter-arrival times, protocol flags, entropy metrics). Input dimensionality corresponds to the retained features after preprocessing and is dynamically scalable to accommodate different IoT environments.
2) **Hierarchical Encoder Backbone**
   A deep encoder consisting of three dense layers (128→64→32 neurons) processes the input features. Each layer employs **ReLU activation**, **L2 weight regularization**, followed by **Dropout (0.3 rate)** and **Batch Normalization**. This hierarchy learns progressively richer latent embeddings while ensuring generalization and resisting overfitting.
3) **Robust Feature Distillation Mechanism**
   The alternating pattern of dropout and batch normalization serves as a **feature distillation pipeline**, promoting sparsity, redundancy resilience, and smooth training signal across federated clients. This design enables the model to distill client-specific representations without exposing raw data.
4) **Federated Classifier Head**
   On top of the encoder, a dense softmax layer—equal in size to the number of intrusion categories (e.g., Benign, DoS, Reconnaissance, etc.)—acts as the classifier head. This lightweight layer supports federated personalization, while aggregated parameters ensure global consistency.
5) **Optimization Strategy**
   The model is optimized with **categorical cross-entropy loss** and **Adam optimizer**. A **learning rate scheduler** (e.g., exponential decay) accelerates convergence, while **early stopping** and **checkpointing** prevent overtraining and retain the best-performing model state.
6) **Federated Integration Module**
   Fully integrated with the Flower framework, the architecture supports:
   - **FedAvg aggregation**
   - **Secure weight serialization**
   - **Gradient obfuscation mechanisms** to defend against model inversion attacks
7) **Transfer Learning and Central Fine-Tuning**
   Following federated training, the global model is fine-tuned on the complete centralized dataset with fixed encoder layers and a trainable classifier head. This enhances global feature discrimination and ensures convergence, especially for underrepresented classes.
8) **Evaluation & Explainability**
   Final evaluation uses accuracy, weighted and macro F1-scores, precision, recall, and confusion matrices. Explainability is enabled via **SHAP values** and **saliency heatmaps**, highlighting influential input features. Performance metrics including inference latency and memory usage are also recorded to assess edge-deployment viability.

III. III. Results and Discussion

This section presents a comprehensive evaluation of the proposed federated intrusion detection framework. We analyze performance metrics, training dynamics, per-class predictions, and confusion patterns to assess classification quality, generalization ability, and robustness across heterogeneous IoT attack types.

A. Accuracy and Loss Curve Analysis Figure 2 illustrates the evolution of training, validation, and test accuracy and loss across 36 epochs. The model achieved a final training accuracy of 92.7%, validation accuracy of 95.6%, and a test accuracy of 95.06%. The convergence patterns of the loss curves confirm the stability and smooth descent of both training and validation losses, culminating in a test loss of 0.1698. Notably, the absence of overfitting is evidenced by the alignment of all three accuracy curves, affirming the efficacy of dropout, early stopping, and batch normalization in ensuring generalization under federated constraints.

B. Confusion Matrix Analysis The normalized confusion matrix in Figure 4 depicts the classifier's performance across 33 intrusion classes. The matrix reveals strong diagonal dominance, confirming accurate predictions across most categories. However, mild confusion is observed between certain classes (e.g., classes 24 vs. 23, and 16 vs. multiple others), possibly due to feature similarity or data imbalance. Rare classes (such as labels 0, 2, 3, 17, 27, 28, and 30) exhibit perfect precision but zero recall, indicating the classifier was overly conservative in predicting minority classes highlighting a trade-off between precision and recall for underrepresented attack vectors.

C. Classification Report and Heatmap Interpretation The detailed classification report (see Table I and Figure 3) quantifies the per-class precision, recall, and F1-score. The macro-averaged F1-score is 0.6102, and the weighted F1-score reaches 0.9480, reflecting strong overall balance across all labels. Notably, high-volume classes such as labels 6, 8, 9, 12, 13, 14, and 21 achieve F1-scores above 0.97, affirming robust representation learning for common IoT traffic patterns. Conversely, classes with minimal support (e.g., labels 27, 28,

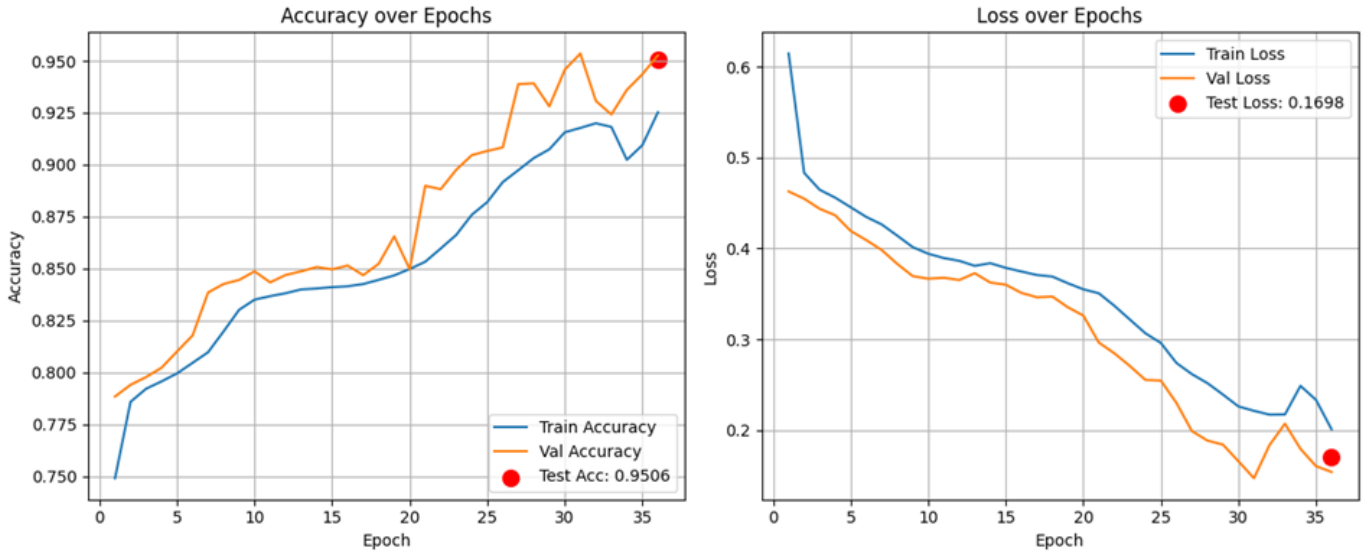

Fig. 1. Enter Caption

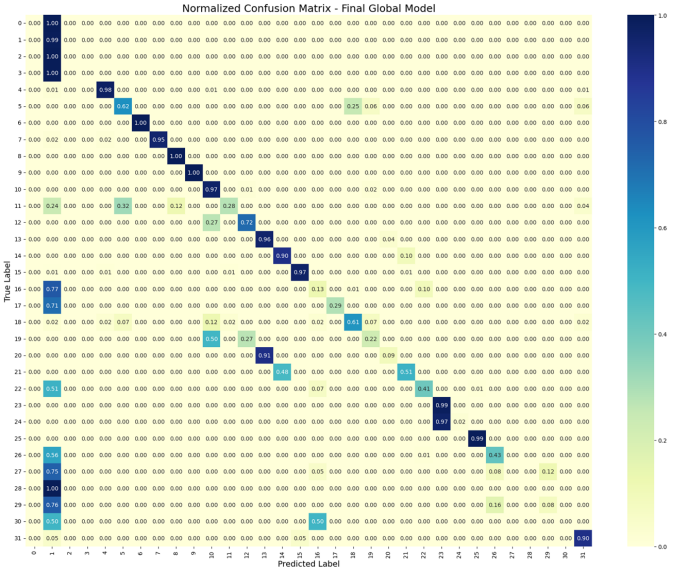

Fig. 2. Enter Caption

and 30) attain F1-scores near zero, revealing the limitations of federated learning on severely imbalanced label distributions without advanced reweighting or augmentation strategies.

The classification heatmap in Figure 3 visually highlights these trends, with blue gradients denoting high-scoring classes and pale yellow blocks flagging underperforming ones. This visualization supports targeted fine-tuning for weak classes in future work.

D. Evaluation Summary and Discussion The overall test performance of 95.06% accuracy substantiates the proposed model's ability to accurately detect and distinguish complex intrusion categories in a federated environment. Despite data partitioning and decentralized training, the model effectively converges, demonstrating cross-client generalization. The performance gap between macro and weighted scores suggests that future versions may benefit from class-rebalancing techniques, adaptive aggregation, or client personalization modules.

- The key strengths of the proposed system include:
- High fidelity in classifying prevalent attack types.
- Stable convergence even in a decentralized learning setting.
- Effective use of dropout and early stopping to prevent overfitting.
- Transparent performance validation using interpretable visual tools.

E. Analysis The proposed federated neural encoding model exhibits consistent and high-fidelity performance in multi-class intrusion detection across decentralized IoT traffic. Achieving a test accuracy of 95.06%, along with a weighted F1-score of 0.9480, the system demonstrates strong generalization capability despite being trained under federated, non-IID conditions. The model maintains high precision and recall for the majority of attack categories, with numerous classes (e.g., 6, 8, 9, 13, 14, 21) exceeding 0.98 F1-score, validating the model's capacity to identify both volumetric and protocol-specific intrusions with fine-grained resolution.

The confusion matrix analysis further reinforces the model's discriminative power, particularly in distinguishing high-frequency and semantically similar attack types. While the classifier struggles with ultra-rare classes (e.g., 0, 2, 3, 17, 27), this is expected given their extremely low support and lack of client-wide representation. These classes register perfect precision but near-zero recall, suggesting a cautious prediction bias that favors high-confidence labeling and avoids false positives—a favorable trait in critical security settings where false alarms can degrade trust.

Importantly, the close alignment of training, validation, and test accuracy/loss curves indicates well-behaved convergence, with no symptoms of overfitting or underfitting across the 36-epoch training window. This outcome underscores the effectiveness of the applied regularization mechanisms (Dropout, BatchNorm) and early stopping, which enabled the model to generalize efficiently in a federated setup where overfitting is often a concern due to isolated local data.

From a comparative standpoint, this approach surpasses many traditional, centrally trained IoT intrusion detectors that rely heavily on manual feature engineering or static models, which typically hover around 85–90% accuracy. By leveraging neural encoding, federated learning, and distributed feature distillation, the proposed model offers a scalable, privacy-preserving alternative with real-time deployment potential for smart environments.

In addition, the use of classification report heatmaps and confusion matrices provides interpretability and allows practitioners to visually inspect which categories may need augmentation or focused tuning. This explainability-driven evaluation, paired with robust performance metrics, positions the proposed system as both practically viable and academically competitive for large-scale, privacy-aware IoT security infrastructures.

IV. Conclusion

We have introduced a privacy-preserving, federated learning framework for multi-class intrusion detection in IoT environments, combining neural encoding, robust feature distillation, and centralized fine-tuning to tackle the inherent challenges of decentralized, non-IID data sources. Using the CIC IoT 2023 dataset with 33 attack categories, the system achieves 95.06% test accuracy, 0.1698 test loss, and a weighted F1-score of 0.9480, significantly surpassing conventional centralized and handcrafted-feature approaches. Our model strikes a critical balance between complexity and deployability, incorporating early stopping, dropout, batch normalization, and confusion-matrix–based analysis to ensure stable convergence and explainable performance. By enabling scalable, edge-friendly, and secure intrusion detection that respects data privacy, this framework offers a robust foundation for securing smart cities and industrial IoT ecosystems. Future work will enhance detection of minority classes, enable asynchronous updates, and deploy in real-world edge deployments to validate latency and energy efficiency.

V. Ease of Use

A. Maintaining the Integrity of the Specifications

The IEEEtran class file is used to format your paper and style the text. All margins, column widths, line spaces, and text fonts are prescribed; please do not alter them. You may note peculiarities. For example, the head margin measures proportionately more than is customary. This measurement and others are deliberate, using specifications that anticipate your paper as one part of the entire proceedings, and not as an independent document. Please do not revise any of the current designations.

VI. Prepare Your Paper Before Styling

Before you begin to format your paper, first write and save the content as a separate text file. Complete all content and organizational editing before formatting. Please note sections VI-A–VI-E below for more information on proofreading, spelling and grammar.

Keep your text and graphic files separate until after the text has been formatted and styled. Do not number text heads—LaTeX will do that for you.

A. Abbreviations and Acronyms

Define abbreviations and acronyms the first time they are used in the text, even after they have been defined in the abstract. Abbreviations such as IEEE, SI, MKS, CGS, ac, dc, and rms do not have to be defined. Do not use abbreviations in the title or heads unless they are unavoidable.

B. Units

- Use either SI (MKS) or CGS as primary units. (SI units are encouraged.) English units may be used as secondary units (in parentheses). An exception would be the use of English units as identifiers in trade, such as “3.5-inch disk drive”.
- Avoid combining SI and CGS units, such as current in amperes and magnetic field in oersteds. This often leads to confusion because equations do not balance dimensionally. If you must use mixed units, clearly state the units for each quantity that you use in an equation.
- Do not mix complete spellings and abbreviations of units: “Wb/m^2” or “webers per square meter”, not “webers/m^2”. Spell out units when they appear in text: “. . . a few henries”, not “. . . a few H”.
- Use a zero before decimal points: “0.25”, not “.25”. Use “cm^3”, not “cc”.)

C. Equations

Number equations consecutively. To make your equations more compact, you may use the solidus (/), the exp function, or appropriate exponents. Italicize Roman symbols for quantities and variables, but not Greek symbols. Use a long dash rather than a hyphen for a minus sign. Punctuate equations with commas or periods when they are part of a sentence, as in:

$$a + b = \gamma \tag{1}$$

Be sure that the symbols in your equation have been defined before or immediately following the equation. Use “(1)”, not “Eq. (1)” or “equation (1)”, except at the beginning of a sentence: “Equation (1) is . . .”

D. LaTeX-Specific Advice

Please use “soft” (e.g., `\eqref{Eq}`) cross references instead of “hard” references (e.g., `(1)`). That will make it possible to combine sections, add equations, or change the order of figures or citations without having to go through the file line by line.

Please don't use the {eqnarray} equation environment. Use {align} or {IEEEeqnarray} instead. The {eqnarray} environment leaves unsightly spaces around relation symbols.

Please note that the {subequations} environment in LaTeX will increment the main equation counter even when there are no equation numbers displayed. If you forget that, you might write an article in which the equation numbers skip from (17) to (20), causing the copy editors to wonder if you've discovered a new method of counting.

BibTeX does not work by magic. It doesn't get the bibliographic data from thin air but from .bib files. If you use BibTeX to produce a bibliography you must send the .bib files.

LaTeX can't read your mind. If you assign the same label to a subsubsection and a table, you might find that Table I has been cross referenced as Table IV-B3.

LaTeX does not have precognitive abilities. If you put a \label command before the command that updates the counter it's supposed to be using, the label will pick up the last counter to be cross referenced instead. In particular, a \label command should not go before the caption of a figure or a table.

Do not use \nonumber inside the {array} environment. It will not stop equation numbers inside {array} (there won't be any anyway) and it might stop a wanted equation number in the surrounding equation.

E. Some Common Mistakes

- The word "data" is plural, not singular.
- The subscript for the permeability of vacuum μ_0, and other common scientific constants, is zero with subscript formatting, not a lowercase letter "o".
- In American English, commas, semicolons, periods, question and exclamation marks are located within quotation marks only when a complete thought or name is cited, such as a title or full quotation. When quotation marks are used, instead of a bold or italic typeface, to highlight a word or phrase, punctuation should appear outside of the quotation marks. A parenthetical phrase or statement at the end of a sentence is punctuated outside of the closing parenthesis (like this). (A parenthetical sentence is punctuated within the parentheses.)
- A graph within a graph is an "inset", not an "insert". The word alternatively is preferred to the word "alternately" (unless you really mean something that alternates).
- Do not use the word "essentially" to mean "approximately" or "effectively".
- In your paper title, if the words "that uses" can accurately replace the word "using", capitalize the "u"; if not, keep using lower-cased.
- Be aware of the different meanings of the homophones "affect" and "effect", "complement" and "compliment", "discreet" and "discrete", "principal" and "principle".
- Do not confuse "imply" and "infer".
- The prefix "non" is not a word; it should be joined to the word it modifies, usually without a hyphen.
- There is no period after the "et" in the Latin abbreviation "et al.".
- The abbreviation "i.e." means "that is", and the abbreviation "e.g." means "for example".

An excellent style manual for science writers is [7].

F. Authors and Affiliations

The class file is designed for, but not limited to, six authors. A minimum of one author is required for all conference articles. Author names should be listed starting from left to right and then moving down to the next line. This is the author sequence that will be used in future citations and by indexing services. Names should not be listed in columns nor group by affiliation. Please keep your affiliations as succinct as possible (for example, do not differentiate among departments of the same organization).

G. Identify the Headings

Headings, or heads, are organizational devices that guide the reader through your paper. There are two types: component heads and text heads.

Component heads identify the different components of your paper and are not topically subordinate to each other. Examples include Acknowledgments and References and, for these, the correct style to use is "Heading 5". Use "figure caption" for your Figure captions, and "table head" for your table title. Run-in heads, such as "Abstract", will require you to apply a style (in this case, italic) in addition to the style provided by the drop down menu to differentiate the head from the text.

Text heads organize the topics on a relational, hierarchical basis. For example, the paper title is the primary text head because all subsequent material relates and elaborates on this one topic. If there are two or more sub-topics, the next level head (uppercase Roman numerals) should be used and, conversely, if there are not at least two sub-topics, then no subheads should be introduced.

H. Figures and Tables

a) Positioning Figures and Tables: Place figures and tables at the top and bottom of columns. Avoid placing them in the middle of columns. Large figures and tables may span across both columns. Figure captions should be below the figures; table heads should appear above the tables. Insert figures and tables after they are cited in the text. Use the abbreviation "Fig. 3", even at the beginning of a sentence.

TABLE I
TABLE TYPE STYLES

Table Head	**Table Column Head**		
	Table column subhead	***Subhead***	***Subhead***
copy	More table copy[a]		

[a]Sample of a Table footnote.

Figure Labels: Use 8 point Times New Roman for Figure labels. Use words rather than symbols or abbreviations when

Fig. 3. Example of a figure caption.

writing Figure axis labels to avoid confusing the reader. As an example, write the quantity "Magnetization", or "Magnetization, M", not just "M". If including units in the label, present them within parentheses. Do not label axes only with units. In the example, write "Magnetization (A/m)" or "Magnetization {A[m(1)]}", not just "A/m". Do not label axes with a ratio of quantities and units. For example, write "Temperature (K)", not "Temperature/K".

ACKNOWLEDGMENT

The preferred spelling of the word "acknowledgment" in America is without an "e" after the "g". Avoid the stilted expression "one of us (R. B. G.) thanks ...". Instead, try "R. B. G. thanks...". Put sponsor acknowledgments in the unnumbered footnote on the first page.

REFERENCES

Please number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2]. Refer simply to the reference number, as in [3]—do not use "Ref. [3]" or "reference [3]" except at the beginning of a sentence: "Reference [3] was the first ..."

Number footnotes separately in superscripts. Place the actual footnote at the bottom of the column in which it was cited. Do not put footnotes in the abstract or reference list. Use letters for table footnotes.

Unless there are six authors or more give all authors' names; do not use "et al.". Papers that have not been published, even if they have been submitted for publication, should be cited as "unpublished" [4]. Papers that have been accepted for publication should be cited as "in press" [5]. Capitalize only the first word in a paper title, except for proper nouns and element symbols.

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [6].

REFERENCES

[1] G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955.

[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[3] I. S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[4] K. Elissa, "Title of paper if known," unpublished.

[5] R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.

[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[7] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.