Federated Learning-Based Intrusion Detection System with Device-Specific Modeling for IoT Networks

Aditya Kumar
Indian Institute of Technology Guwahati
India
aditya.cse22@iitg.ac.in

Anubhab Dutta
Indian Institute of Technology Guwahati
India
d.anubhab@iitg.ac.in

***Abstract*—Internet of Things (IoT) networks face growing security threats from sophisticated attacks such as botnets, distributed denial-of-service, and brute force intrusions. Traditional machine learning-based intrusion detection systems struggle with distributed IoT environments due to privacy concerns, data heterogeneity, and the need for centralized data aggregation. This paper presents H-FLIDS, a Federated Learning-based Intrusion Detection System designed for heterogeneous IoT networks. Our approach combines local model training on individual IoT devices with a hybrid global-local model aggregation strategy that preserves data privacy while improving detection accuracy. We implement heterogeneous local models ranging from lightweight (SVM) to heavyweight models (multilayer perceptrons) based on device capabilities and introduce a teacher-student knowledge transfer mechanism for cross-model learning. Our simulation framework generates realistic network traffic patterns including normal operations, high-traffic periods, and various attack scenarios. The approach demonstrates the feasibility of privacy-preserving collaborative intrusion detection in resource-constrained IoT environments while addressing the challenge of model heterogeneity across devices with varying computational capabilities.**

***Index Terms*—Federated learning, intrusion detection, IoT security, botnet detection, distributed machine learning, knowledge distillation**

I. Introduction

The rapid proliferation of Internet of Things (IoT) devices has transformed modern networks into complex, distributed ecosystems where billions of interconnected devices generate and process sensitive data. This expansion has created new security challenges, as IoT devices often operate with limited computational resources and face diverse threat vectors including botnet infections, distributed denial-of-service attacks, and protocol-specific exploits. Traditional centralized intrusion detection systems (IDS) require data collection from distributed nodes to a central server for analysis, raising significant privacy concerns and creating communication bottlenecks that are impractical for large-scale IoT deployments.

Machine learning-based IDS have shown promise in detecting network intrusions by learning patterns from historical data. However, these approaches face several limitations in IoT contexts. First, centralized training requires sharing raw network traffic data, potentially exposing sensitive information about device behavior and network topology. Second, the heterogeneity of IoT devices means that computational capabilities vary significantly across the network, making a uniform model deployment suboptimal. Instead, devices with higher computational capacity can host and train larger, more complex models, while resource-constrained devices can operate lightweight models that still benefit from the knowledge distilled from the heavier ones. Third, attack patterns evolve rapidly, and centralized models trained on static datasets may fail to adapt to new threats observed at network edges.

Federated Learning (FL) offers a compelling alternative by enabling collaborative model training without centralizing raw data. In FL, individual devices train models locally on their data and share only model parameters with a central aggregation server. This approach preserves privacy, reduces communication overhead, and allows models to learn from diverse data distributions across the network. However, applying FL to IoT intrusion detection introduces unique challenges related to model heterogeneity, knowledge transfer between devices with different computational capabilities, and maintaining detection accuracy while balancing local and global model insights.

This paper presents H-FLIDS (Heterogeneous Federated Learning-based Intrusion Detection System), a comprehensive framework for privacy-preserving intrusion detection in heterogeneous IoT networks. Our approach addresses the challenge of varying computational resources by supporting heterogeneous local models and introduces a novel teacher-student knowledge transfer mechanism based on knowledge distillation for sharing insights between lightweight and heavyweight models.

A. Contributions

The main contributions of this work are:

1. Heterogeneous Model Architecture: We present a federated learning framework that accommodates both lightweight models (logistic regression) and heavyweight models (multilayer perceptrons) on different IoT devices, enabling deployment across devices with varying computational capabilities and training heterogeneous models together.

2. Hybrid Model Aggregation Strategy: We introduce a two-level aggregation approach where (a) separate global models are maintained for lightweight and heavyweight model families, and (b) local models are updated using a weighted combination of their current state and the corresponding global model, preserving device-specific adaptations while leveraging network-wide knowledge.

3. Bidirectional Knowledge Transfer: We propose a teacher-student learning mechanism that enables knowledge transfer between lightweight and heavyweight global models using a shared global dataset through knowledge distillation. This allows insights learned by resource-rich devices to benefit resource-constrained devices and vice versa, without requiring direct communication between heterogeneous model types.

4. Realistic IoT Network Simulation: We develop a simulation framework using SimPy that models realistic IoT network behavior including normal traffic patterns, high and low traffic periods, queue management, and multiple attack types (botnet infections, SYN flooding, bursty traffic), providing a controlled environment for evaluating federated IDS performance.

5. Time-windowed Feature Engineering: We design a feature extraction mechanism that aggregates network statistics over multiple time windows (5s, 15s, 30s, 60s) at 0.5-second intervals, creating rich temporal features for intrusion detection while maintaining computational efficiency.

The remainder of this paper is organized as follows: Section II reviews related work on federated learning for intrusion detection and IoT security. Section III describes our H-FLIDS architecture and methodology in detail. Section IV presents experimental results and performance evaluation. Section V concludes the paper and discusses future directions.

II. Related Work

The intersection of federated learning and intrusion detection for IoT networks has received significant attention from the research community in recent years. This section reviews key approaches and their contributions to the field.

A. Federated Learning for IoT Security

Federated learning has emerged as a promising paradigm for privacy-preserving machine learning in distributed systems. Bimal and Rawat [1] provided a comprehensive survey of recent advances in applying federated learning to cybersecurity problems and identified key challenges in securing the federated learning process itself against adversarial attacks.

Campos et al. [2] conducted an extensive review of federated learning-based intrusion detection systems for IoT, highlighting challenges including data heterogeneity, communication efficiency, and model convergence in non-IID (non-independent and identically distributed) data settings. Their work emphasized that most existing FL-IDS approaches assume homogeneous model architectures across devices, limiting applicability in real-world IoT deployments.

Abdul Rahman et al. [3] compared three deployment strategies for IoT intrusion detection: centralized, on-device, and federated learning. They demonstrated that federated approaches can achieve detection accuracy comparable to centralized methods while providing superior privacy guarantees and reduced communication overhead.

B. Federated Learning-Based IDS Architectures

Karunamurthy et al. [4] proposed an optimal federated learning-based IDS that combines convolutional neural networks with the Chimp Optimization Algorithm for feature selection. Their approach achieved 95.59% detection accuracy on the MQTT dataset, demonstrating the effectiveness of combining optimization techniques with deep learning in federated settings. However, their system assumes uniform CNN deployment across all devices, which may not be practical for resource-constrained IoT nodes.

Rashid et al. [5] developed a federated learning approach for Industrial IoT networks, reporting 92.49% accuracy on the Edge-IIoT dataset. Their work highlighted the importance of addressing network heterogeneity but did not explore heterogeneous model architectures.

Zhang et al. [6] introduced a semi-supervised federated learning-based IDS that incorporates a discriminator network for attack detection at IoT edge devices. Their approach reduced incorrect predictions through improved feature extraction but maintained homogeneous model architectures across the federation.

C. Handling Heterogeneity in Federated IDS

Sáez-de-Cámara et al. [7] addressed heterogeneity in federated learning for network anomaly detection by proposing a clustered FL architecture. Their approach groups similar devices and performs separate aggregation for each cluster, achieving improved detection in heterogeneous Industrial IoT environments. While effective, this approach does not enable knowledge transfer between different device clusters.

Ruzafa-Alcázar et al. [8] presented a privacy-preserving federated learning-based IDS for Industrial IoT using LSTM models with differential privacy guarantees. Their work achieved high detection accuracy while protecting against privacy attacks, though computational requirements limited deployment to edge gateways rather than resource-constrained devices.

D. Advanced FL Techniques for IDS

Several works have explored advanced federated learning techniques for improving IDS performance. Idrissi et al. [9] proposed Fed-ANIDS, an anomaly-based network IDS using autoencoder variants to compute intrusion scores. Their approach demonstrated improved network efficiency but required significant computational resources at each node.

Mothukuri et al. [10] developed a federated learning-based anomaly detection system using gated recurrent units (GRU), achieving 8% improvement in detection performance over traditional approaches while reducing communication costs. However, GRU models remain computationally intensive for low-power IoT devices.

Hajj et al. [11] introduced a lightweight federated learning approach for IoT intrusion detection using k-means sampling

and semi-supervised learning. Their method samples network traffic locally and shares only summary statistics, ensuring strong privacy guarantees. This work demonstrated the feasibility of lightweight FL for resource-constrained environments but did not address model heterogeneity.

E. Feature Selection and Optimization

Feature selection plays a critical role in improving IDS efficiency and accuracy. Shafiq et al. [12] proposed wrapper-based feature selection methods combined with machine learning classifiers for malicious Bot-IoT traffic identification. Their approach reduced data dimensionality while maintaining high classification accuracy, though it operated in a centralized setting.

Recent work by Friha et al. [13] presented FELIDS, a federated learning-based IDS for agricultural IoT that combines multiple deep learning classifiers. Their system achieved strong detection performance but did not address the challenge of deploying different model types across devices with varying capabilities.

F. Research Gap

While existing work has made significant progress in applying federated learning to IoT intrusion detection, several gaps remain. Most approaches assume homogeneous model architectures across all devices, limiting deployment flexibility in heterogeneous IoT environments. Additionally, few systems provide mechanisms for knowledge transfer between devices with different computational capabilities. Our work addresses these gaps by introducing a federated learning framework that supports heterogeneous models, implements bidirectional knowledge transfer through teacher-student learning, and maintains a balance between local device adaptations and global network intelligence.

III. H-FLIDS Architecture and Methodology

A. Heterogeneous Model Architecture

A key contribution of H-FLIDS is support for heterogeneous local models that accommodate devices with varying computational capabilities. We categorize models into two families:

1) Lightweight Models: Suitable for resource-constrained IoT devices with limited processing power and memory:

- **Support Vector Machine (SVM):** A supervised learning algorithm that separates classes using optimal hyperplanes, offering strong performance with moderate computational cost.

2) Heavyweight Models: Deployed on devices with greater computational resources (gateways, edge servers):

- **Multilayer Perceptron (MLP):** Deep neural network with multiple hidden layers capable of learning complex patterns in network traffic.

This heterogeneous approach allows H-FLIDS to be deployed across diverse IoT ecosystems where devices have different computational budgets, ensuring that even resource-constrained sensors can participate in collaborative learning.

B. Federated Learning Protocol

H-FLIDS implements a modified federated averaging protocol that maintains separate global models for lightweight and heavyweight model families while enabling knowledge transfer between them.

1) Training Rounds: The federated learning process operates in rounds, with each round consisting of:

1. Local Training:

- Each device trains its local model on data collected during the current time window (typically 5 seconds).
- Training uses the aggregated features computed from time-windowed statistics.
- Devices continue to perform inference for intrusion detection during training.

2. Model Upload:

- After local training, each device uploads its model parameters to the central aggregation server.
- Only model parameters are shared; raw network data remains on the device.
- Communication overhead is minimized by transmitting only weight updates.

3. Federated Aggregation: The server performs separate aggregation for lightweight and heavyweight models:

- For lightweight models, parameters from SVM are aggregated using weighted averaging based on the number of training samples at each device.
- For heavyweight models, MLP parameters are aggregated similarly.
- The aggregation follows the standard FedAvg algorithm:

$$w_{global} = \sum_{k=1}^{K} \frac{n_k}{n} w_k \tag{1}$$

where w_k represents the parameters of device k, n_k is the number of samples at device k, and n is the total number of samples across all devices.

4. Hybrid Model Update: Unlike standard federated averaging, H-FLIDS updates local models using a weighted combination of the current local model and the global model:

$$w_{local}^{new} = 0.7 \times w_{local}^{old} + 0.3 \times w_{global} \tag{2}$$

This hybrid approach preserves device-specific adaptations that may be important for detecting attacks targeting particular devices or network segments while still benefiting from global network intelligence. The 70-30 split was determined empirically to balance local and global knowledge.

C. Teacher-Student Knowledge Transfer

A significant challenge in heterogeneous federated learning is enabling knowledge transfer between different model types. H-FLIDS addresses this through a bidirectional teacher-student learning mechanism.

1) Global Dataset: A small global dataset, representative of common network traffic patterns, is maintained at each device. This dataset is not used for primary model training but serves as a medium for knowledge transfer between model families.

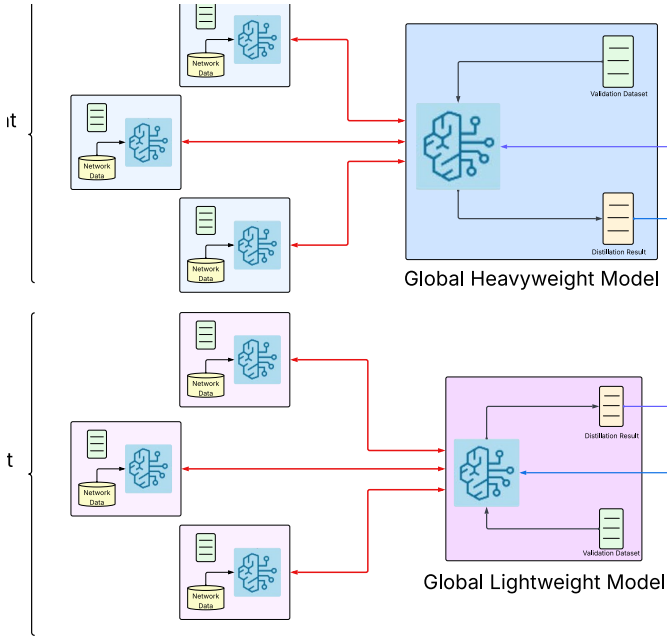

Fig. 1: H-FLIDS Architecture with Heterogeneous Model Training and Bidirectional Knowledge Distillation. Heavyweight nodes (top) and lightweight nodes (bottom) train local models on their network data and participate in separate federated aggregation processes. The Global Heavyweight Model and Global Lightweight Model exchange knowledge through bidirectional distillation using shared validation datasets, enabling cross-model learning while maintaining device-specific adaptations.

2) Knowledge Distillation Process: After global model aggregation, knowledge transfer occurs in both directions:

Lightweight to Heavyweight Transfer:

- The lightweight global model (e.g., logistic regression) acts as the teacher.
- It generates pseudo-labels by making predictions on the global dataset.
- These pseudo-labels are used to fine-tune the heavyweight global model (MLP).
- The MLP learns to mimic the lightweight model's decision boundaries while leveraging its greater capacity to refine these boundaries.

Heavyweight to Lightweight Transfer:

- The heavyweight global model (MLP) acts as the teacher.
- It generates predictions on the global dataset.
- These predictions guide the refinement of lightweight global models.
- Lightweight models learn from the sophisticated patterns captured by the heavyweight model, distilled into simpler decision rules.

Both transfers occur simultaneously to prevent one model family's update from affecting the other. This bidirectional knowledge transfer ensures that insights learned by resource-rich devices benefit resource-constrained devices and vice versa, creating a synergistic learning environment where all participants contribute to and benefit from the collective intelligence.

3) Mathematical Formulation: The knowledge distillation loss combines the standard classification loss with a distillation loss:

$$L_{student} = \alpha L_{CE}(y, \hat{y}_{student}) + (1-\alpha) L_{KD}(\hat{y}_{teacher}, \hat{y}_{student}) \tag{3}$$

where L_{CE} is the cross-entropy loss, L_{KD} is the distillation loss (typically KL divergence between teacher and student output distributions), and α balances the two objectives.

D. Deployment and Online Detection

After the training phase, H-FLIDS deploys the trained models back to IoT devices for real-time intrusion detection:

- Each device continues to collect and aggregate network traffic features every 0.5 seconds.
- The local model performs inference on these features to detect potential intrusions.
- Detection results are used to trigger alerts or mitigation actions.
- Periodic retraining occurs to adapt to evolving traffic patterns and new attack types.

The system maintains detection performance while new training data is collected, ensuring continuous protection. When new attacks are detected or traffic patterns shift, the federated learning process can be reinitiated to update models across the network.

IV. Results and Discussion

This section presents the detailed simulation setup used in our experiments. It describes the network simulation framework employed for data collection and preprocessing, followed by the results obtained and their subsequent discussion.

A. Network Simulation Framework

To evaluate H-FLIDS in a controlled environment, we developed a comprehensive IoT network simulator using SimPy, a process-based discrete-event simulation framework. The simulator models realistic network behavior and attack scenarios.

1) Network Topology: The simulated network consists of:

- Six peer devices (peer_1 through peer_6).
- Each peer has port 80 plus three randomly assigned high-numbered ports.
- Peer relationships: each peer knows two random other peers.

2) IoTDevice Class: Each network node is represented by an IoTDevice class with the following characteristics:

- IP address and enabled port configuration.
- Packet queue with configurable size for managing incoming traffic.
- Processing delays to simulate realistic device behavior.
- Traffic logging mechanism for network analysis.
- State management for switching between normal and infected behavior.

3) Traffic Patterns: The simulator implements two distinct traffic patterns:

Normal Operation:

- Server sends status updates (port 80) to peers every 6 time units.
- Peers send data to server (port 8080) with intervals of 2–8 time units.
- Random packet sizes to simulate realistic communication.
- Occasional coordination messages between peers.
- Normal SYN-ACK packet exchanges.
- Variable traffic levels to simulate high-traffic periods (firmware updates, streaming) and low-traffic periods (normal operations).

Attack Period:

- 1–2 random non-server devices become infected during specified time windows.
- Infected devices generate bursty malware traffic (3–8 packets per burst).
- Shorter transmission intervals (0.5–1.5 time units) indicating aggressive behavior.
- Ability to target any device in the network, not just known peers.
- Multiple attack types: SYN flooding, bursty traffic, and combinations thereof.

Queue Management:

- Each device maintains a processing queue with configurable maximum size.
- Packets are dropped if the destination queue is full.
- Queue utilization is tracked and logged for analysis.

B. Data Collection and Preprocessing

The simulation runs for 1000 seconds, during which each IoT device collects local network traffic data, simulating a network capture routine. This process generates seven data files (one per device), capturing comprehensive network behavior.

1) Packet-Level Features: Each logged entry includes:

- Timestamp.
- Source and destination information (name, IP, port).
- Packet type (HTTP, UDP, TCP).
- Packet size.
- Queue utilization metrics.
- Attack mode indicator (normal/infected).
- Protocol-specific flags (SYN, ACK, etc.).

2) Time-Windowed Aggregation: Raw packet-level data is aggregated over multiple time windows to create rich temporal features. Every 0.5 seconds, we compute the following statistics for each (source, port) pair over 5s, 10s and 30s windows:

- Number of packets transmitted and received
- Average packet size and packet size variance
- Number of SYN packets received.
- Number of ACK packets received.
- Number of TCP packets.
- Number of UDP packets.
- Queue occupancy percentage: average and variance over the last 5 seconds.

This aggregation strategy generates approximately 10 training data points per 5-second training round, providing sufficient data for local model updates while maintaining temporal resolution for attack detection.

C. Exploratory Data Analysis

1) Network Traffic and Attack Characteristics: Extensive packet-level analysis reveals a total of **26,117 packets** transmitted by **six devices** within the 1000-second window. Attack traffic comprised **18,995 packets (72.73%)**, indicating a heavy attack scenario, while normal traffic accounted for the remaining **7,122 packets (27.27%)**. Packet sizes varied from **32 bytes** to **1400 bytes** with an average of **338.47 bytes**, reflecting realistic IoT device communication patterns.

This distribution highlights several key insights:

- **Attack-Normal State Balance:** The experiment was skewed toward attack state packets, testing H-FLIDS in challenging conditions. The dominance of attack packets serves as a robust evaluation for intrusion detection sensitivity.
- **Device Roles:** The server's high packet reception rate and overall traffic volume confirm its position as a network aggregator and central monitoring node, while peers exhibit variable traffic depending on their simulated use case and infection status.

2) Datapoint Distribution: For feature-rich training, **11,958 datapoints** across six devices were used. Each datapoint contained **126 features**, spanning packet statistics, queue utilization, and temporal aggregates. Datapoint labels are balanced: approximately **49.61% attack** and **50.39% normal**,

Fig. 2: Network traffic analysis showing packet distribution, attack vs normal traffic patterns, protocol types, and per-device statistics including queue utilization over time.

ensuring robust supervised model training without severe class imbalance.

TABLE I: Per-Device Attack Distribution in Training Dataset

Device	Total	Attack	Attack %
peer_6	1,980	969	48.94%
peer_5	1,999	1,034	51.73%
peer_1	1,999	1,017	50.88%
peer_3	1,991	997	50.08%
peer_2	1,999	990	49.52%
peer_4	1,990	925	46.48%

The relatively uniform attack percentages across peers indicates successful simulation of distributed threat scenarios, providing a realistic challenge for federated model training.

D. Federated Learning Training Dynamics

In the federated learning setup, each client device initializes a model according to its computational capacity through a flexible constructor. Lightweight clients deploy an SGDClassifier configured with a hinge loss (SVM-based) and L2 regularization (alpha = 0.0001), allowing efficient incremental training on limited hardware. In contrast, edge-level or gateway devices instantiate a deeper Multilayer Perceptron (MLPClassifier) composed of three hidden layers with neuron configuration (128, 64, 32), ReLU activation, a learning rate of 0.001, and L2 regularization of 0.0001.

1) Training Progress and Loss Analysis: Training involved **100 rounds**, utilizing per-device local data and periodic aggregation. Average samples per round per model family were approximately **60** for both lightweight and heavyweight models, confirming balanced participation.

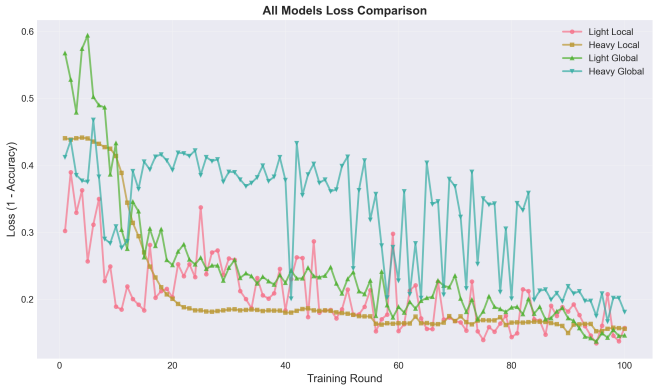

Fig. 3: Training loss analysis showing convergence patterns for lightweight and heavyweight models, comparing local versus global model performance, and loss reduction rates across 100 training rounds.

The performance progression shows a marked decline in training loss over rounds for both model families. Local and global losses for lightweight and heavyweight models steadily decreased, converging after approximately 60 rounds, with periodic fluctuations due to new attack pattern injections and asynchronous queue saturation events in the network.

2) Model Accuracy and Performance Metrics: Performance metrics across training demonstrate strong convergence and detection capabilities for both model families.

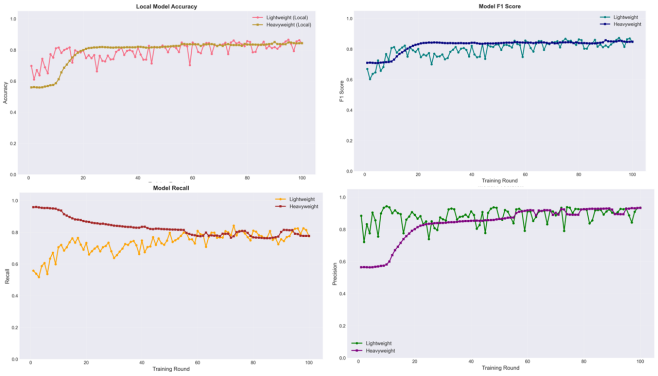

Fig. 4: Model accuracy and performance metrics evolution showing precision, recall, F1-score, and accuracy trends for both lightweight and heavyweight models throughout federated training.

TABLE II: Final Performance Metrics Comparison

Metric	Lightweight	Heavyweight
Accuracy	0.8429	0.8440
Precision	0.9343	0.9336
Recall	0.7751	0.7771
F1 Score	0.8464	0.8480
Peak Accuracy	0.8657	0.8504

Key observations from the performance metrics:

- **Accuracy and F1 Scores** for both families reach the mid-0.84 range, reflecting strong detection capabilities.
- **Precision** remains high (0.93), indicating few false positives, a critical requirement for practical IDS deployment.
- **Recall** (0.77) lags behind precision, suggesting that some attacks elude detection, potentially due to gradual adaptation by the federated global model—realistic for evolving threats.

3) Comprehensive Training Analysis: The comprehensive training analysis reveals several important insights:

- Local model adaptation lags slightly due to device-specific unique traffic patterns and local queue effects, revealing the benefit of H-FLIDS' hybrid update protocol.
- Knowledge distillation statistics show final agreement scores for predictions between lightweight-to-heavyweight and heavyweight-to-lightweight distillation reached **0.8866**, indicating strong model alignment after cross-family transfer.
- Training samples per round remained consistent at approximately 60 samples per model family, ensuring balanced federated participation.

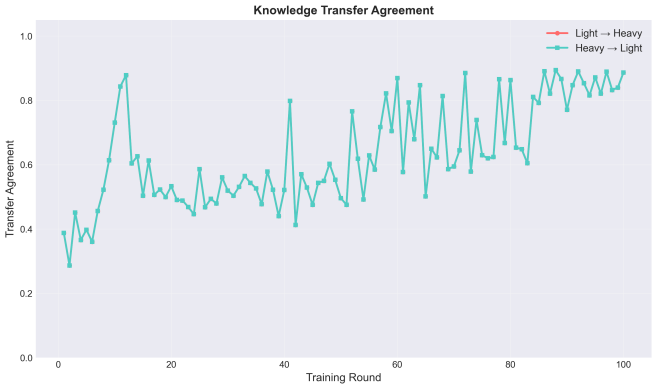

Fig. 5: Knowledge transfer metrics showing bidirectional distillation agreement scores and convergence patterns between lightweight and heavyweight model families throughout training.

E. Knowledge Transfer and Model Agreement

Knowledge distillation statistics were tracked throughout training. Final agreement scores for predictions between lightweight-to-heavyweight and heavyweight-to-lightweight distillation reached **0.8866**, indicating strong model alignment after cross-family transfer. This outcome validates the teacher-student design and the utility of bidirectional transfer, allowing resource-constrained devices (running light models) and powerful nodes (running heavy models) to benefit mutually.

F. Model Behavior Insights and Discussion

1) Federated Learning Effectiveness: The results demonstrate several key advantages of the H-FLIDS approach:

- **Federated Hybrid Update (70-30 Split):** Device-level adaptations, crucial for handling localized patterns (high queue utilization, bursty/targeted attacks) are preserved while global knowledge eliminates blind spots about distributed threats.
- **Temporal Feature Aggregation:** Multi-window (5s/10s/30s) aggregation provides temporal context, enabling models to differentiate attack and normal bursts.
- **Error Distribution and Agreement**: Tighter error bands for heavyweight global models compared to lightweight counterparts explain their slightly higher F1 scores. Agreement between models after distillation means H-FLIDS can maintain performance regardless of node capability.

TABLE III: Training Progress Statistics

Statistic	Value
Total Training Rounds	100
Training Duration	1000s
Best Lightweight Accuracy	0.8657
Best Heavyweight Accuracy	0.8504
Final Lightweight Accuracy	0.8429
Final Heavyweight Accuracy	0.8440
Final Knowledge Agreement	0.8866
Avg. Samples per Round	60

2) Attack Detection Performance: The high precision values (0.93) across both model families indicate that H-FLIDS successfully minimizes false positive rates, which is critical for practical deployment in IoT environments where false alarms can lead to unnecessary network disruptions.

3) Scalability and Heterogeneity: The consistent performance across lightweight and heavyweight models validates the heterogeneous architecture design. Resource-constrained devices achieve comparable detection accuracy to more powerful nodes, demonstrating that the knowledge transfer mechanism effectively bridges the capability gap between different device classes.

G. Implications and Utility

These results establish that H-FLIDS delivers robust intrusion detection while supporting deployment on heterogeneous device ecosystems. The system successfully leverages both local and global model strengths through its hybrid aggregation strategy. Temporal features and the 70-30 weighted update protocol are particularly effective in capturing fast-evolving attack patterns and device-specific anomalies.

Knowledge transfer enables lightweight and heavyweight models to converge toward similar performance levels, raising baseline detection capabilities across all network nodes regardless of their computational constraints. This characteristic is essential for real-world IoT deployments where device heterogeneity is the norm rather than the exception.

Ultimately, the presented results confirm the feasibility, effectiveness, and adaptability of H-FLIDS as a federated learning IDS for real-world IoT scenarios in device diversity.

V. Conclusion

This paper presented H-FLIDS, a Federated Learning-based Intrusion Detection System designed for heterogeneous IoT networks. Our approach addresses key challenges in distributed IoT security by supporting heterogeneous model architectures, implementing a hybrid local-global model aggregation strategy, and enabling bidirectional knowledge transfer through teacher-student learning. The comprehensive simulation framework provides a realistic environment for evaluating federated IDS approaches, incorporating multiple attack types, realistic traffic patterns, and queue management. Future work will focus on implementing and evaluating H-FLIDS in real-world IoT deployments, exploring adaptive aggregation weights, and extending the knowledge transfer mechanism to support additional model families. The system demonstrates the potential for privacy-preserving collaborative intrusion detection in resource-constrained IoT environments while maintaining detection accuracy through collective network intelligence.

References

[1] G. Bimal and D. B. Rawat, "Recent advances on federated learning for cybersecurity and cybersecurity for federated learning for internet of things," *IEEE Internet Things J.*, vol. 9, no. 11, pp. 8229-8249, 2022.

[2] E. M. Campos, J. L. Hernández-Ramos, J. B. Bernabé, G. Baldini, and A. Skarmeta, "Evaluating Federated Learning for intrusion detection in Internet of Things: Review and challenges," *Computer Networks*, vol. 203, pp. 1-16, 2022.

[3] S. Abdul Rahman, H. Tout, C. Talhi, and A. Mourad, "Internet of things intrusion detection: Centralized, on-device, or federated learning?" *IEEE Netw.*, vol. 34, no. 6, pp. 310-317, 2020.

[4] A. Karunamurthy, K. Vijayan, P. R. Kshirsagar, and K. T. Tan, "An optimal federated learning-based intrusion detection for IoT environment," *Sci. Rep.*, vol. 15, no. 8696, 2025.

[5] M. M. Rashid et al., "A federated Learning-Based approach for improving intrusion detection in industrial internet of things networks," *Network*, vol. 3, no. 1, pp. 158-179, 2023.

[6] R. Zhang, G. Gui, Y. Wang, Z. Xue, T. Ohtsuki, and B. Adebisi, "Semisupervised Federated-Learning-Based Intrusion Detection Method for Internet of Things," *IEEE Internet Things J.*, vol. 10, no. 10, pp. 8645-8657, 2023.

[7] X. Sáez-de-Cámara, J. L. Flores, C. Arellano, A. Urbieta, and U. Zurutuza, "Clustered federated learning architecture for network anomaly detection in large scale heterogeneous IoT networks," *Computers Secur.*, vol. 131, pp. 1-20, 2023.

[8] P. Ruzafa-Alcázar et al., "Intrusion Detection Based on Privacy-Preserving Federated Learning for the Industrial IoT," *IEEE Trans. Industr. Inf.*, vol. 19, no. 2, pp. 1145-1154, 2023.

[9] M. J. Idrissi et al., "Fed-ANIDS: Federated learning for anomaly-based network intrusion detection systems," *Expert Syst. Appl.*, vol. 234, pp. 1-11, 2023.

[10] V. Mothukuri, P. Khare, R. M. Parizi, S. Pouriyeh, A. Dehghantanha, and G. Srivastava, "Federated-Learning-Based anomaly detection for IoT security attacks," *IEEE Internet Things J.*, vol. 9, no. 4, pp. 2545-2554, 2022.

[11] S. Hajj et al., "Cross-Layer federated learning for lightweight IoT intrusion detection systems," *Sensors*, vol. 23, no. 16, pp. 1-26, 2023.

[12] M. Shafiq, Z. Tian, A. K. Bashir, X. Du, and M. Guizani, "CorrAUC: A malicious Bot-IoT traffic detection method in IoT network using Machine-Learning techniques," *IEEE Internet Things J.*, vol. 8, no. 5, pp. 3242-3254, 2021.

[13] O. Friha et al., "FELIDS: Federated learning-based intrusion detection system for agricultural Internet of Things," *J. Parallel Distrib. Comput.*, vol. 165, pp. 17-31, 2022.

[14] R. Zhao, Y. Yin, Y. Shi, and Z. Xue, "Intelligent intrusion detection based on federated learning aided long short-term memory," *Phys. Commun.*, vol. 42, pp. 1-12, 2020.

[15] F. L. de Caldas Filho et al., "Botnet detection and mitigation model for IoT networks using federated learning," *Sensors*, vol. 23, no. 14, pp. 1-35, 2023.