Internship Report

Development of a Federated Learning-Based Intrusion Detection System for IoT Networks

Submitted by:
Aya Squalli Houssaini

In partial fulfillment of the requirements for the degree of
Engineering Degree in Computer Science

Supervised by:

Mr. Hamza MOUNCIF
Research Advisor - 3DSmart Factory

Mr. Thierry BERTIN GARDELLE
CEO – 3DSmart Factory

Mrs. Chaymae Benhammacht
Research Advisor - 3DSmart Factory

Univeristy Mohammed VI Polytechnic
College Of Computing
3DSmart Factory

Internship Period: August 2025 – September 2025

Development of a Federated Learning-Based Intrusion Detection System for IoT Networks

Aya Squalli Houssaini

September 29, 2025

Abstract

The proliferation of Internet of Things (IoT) devices has introduced significant security challenges, particularly due to their resource constraints and sensitivity to data privacy. Traditional intrusion detection systems (IDS) often fail to address these constraints, especially when relying on centralized, supervised models. This internship project proposes a privacy-preserving, lightweight, and adaptive IDS that integrates *Kitsune*—an unsupervised anomaly detector—for real-time feature extraction on the router, with *Federated Learning* (FL) to enable collaborative model training across distributed IoT devices. The system is evaluated on the N-BaIoT dataset and benchmarked against supervised models (Random Forest, XGBoost and SVM). Post-detection analysis employs agglomerative hierarchical clustering to characterize attack patterns. Results demonstrate that the federated Kitsune approach achieves near-optimal detection performance (98.7% TPR, 1.2% FPR) while preserving privacy, requiring no labeled data, and remaining deployable on edge hardware such as Raspberry Pi. These findings highlight the practicality of federated, unsupervised IDS solutions for dynamic and privacy-sensitive IoT networks.

1 Introduction

The exponential growth of the Internet of Things (IoT) has introduced billions of heterogeneous devices into critical domains such as healthcare, industry, and smart homes. Despite their advantages, IoT devices remain highly vulnerable to cyberattacks due to weak authentication mechanisms, resource limitations, and the absence of regular security updates. Intrusion Detection Systems (IDS) have emerged as a defense mechanism, but conventional centralized and signature-based IDS approaches face three challenges in IoT settings: (i) data privacy concerns, (ii) inability to generalize to novel attacks, and (iii) infeasibility on resource-constrained edge devices.

To address these limitations, this project develops a novel IDS that combines:

- **Kitsune** [1]: a lightweight, unsupervised anomaly detector based on an ensemble of autoencoders, deployed directly on the network router for real-time feature extraction and traffic mapping.

- **Federated Learning** [3]: a decentralized training paradigm where only model updates—not raw data—are shared across devices.
- **Agglomerative Hierarchical Clustering (AGNES)** [4]: for post-hoc grouping of detected anomalies into meaningful attack families.

The system is evaluated on the N-BaIoT dataset [2], which contains real traffic from nine commercial IoT devices under both benign and malicious (Mirai/BASHLITE) conditions. Performance is compared against supervised baselines: Random Forest and XGBoost [5, 6].

2 Related Work

2.1 IoT Security Landscape

Recent surveys (e.g., [9]) highlight that IoT networks are highly vulnerable due to poor authentication, weak encryption, and firmware flaws. Botnets like Mirai and BASHLITE have exploited these weaknesses for large-scale DDoS campaigns. IDS for IoT can be broadly classified into:

- **Signature-based IDS:** accurate for known threats but fail against novel attacks.
- **Anomaly-based IDS:** model normal traffic patterns and flag deviations, making them suitable for detecting zero-day threats.

2.2 Kitsune and KitNET

Kitsune [1] is an online, unsupervised NIDS that uses a feature extraction engine to compute 115 statistical features from packet streams in real time. Its core, KitNET, employs an ensemble of autoencoders to model normal traffic; deviations (high reconstruction error) signal anomalies. Kitsune runs efficiently on edge devices (e.g., Raspberry Pi and Linux machines), making it ideal for router deployment.

2.3 N-BaIoT Dataset

The N-BaIoT dataset [2] provides labeled network traffic from nine IoT devices infected by Mirai and BASHLITE botnets. It includes 115 features derived from host- and port-level traffic statistics, enabling both anomaly detection and multi-class classification (1 benign + 10 attack types).

2.3.1 Botnet Families in the N-BaIoT dataset

Mirai Botnet Devices infected by Mirai continuously scan the internet for IP addresses of vulnerable IoT devices. It uses a table of over 60 common factory default credentials to compromise devices. Infected devices remain functional but are recruited into botnets for DDoS attacks.

BASHLITE (Gafgyt) Similar to Mirai but supports additional attack vectors, including TCP/UDP floods and HTTP floods. It targets Linux-based embedded systems using brute-force SSH/Telnet attacks.

2.4 Federated Learning and FedAvg

Federated Averaging (FedAvg) [3] enables collaborative model training without data centralization. Clients train locally and send parameter updates to a server, which computes a weighted average to form a global model. This preserves privacy and reduces communication overhead—critical for IoT networks.

Recent work by Olanrewaju-George and Pranggono [8] demonstrated that a federated AutoEncoder trained on N-BaIoT outperforms federated DNNs, especially in reducing the False Positive Rate (FPR). Their results support the suitability of unsupervised learning for privacy-preserving IDS—a principle our work extends by integrating Kitsune's proven online feature extraction for real-world deployment

2.5 Gradient Boosting , Random Forest and Support Vector Machines

Gradient boosting (e.g., XGBoost [6]), Random Forest [5], and Support Vector Machines (SVM) [10] are powerful supervised learning methods widely used for classification tasks in intrusion detection. While highly accurate under controlled conditions, these approaches fundamentally rely on labeled training data and assume that the distribution of attacks remains static over time. Consequently, they struggle to generalize to previously unseen or evolving attack patterns without frequent and costly retraining—limiting their practicality in dynamic IoT environments where novel threats emerge continuously.

2.6 Agglomerative Clustering (AGNES)

AGNES [4] is a bottom-up hierarchical clustering method that merges the most similar clusters iteratively, producing a dendrogram. It is used here to group detected anomalies into interpretable attack clusters without supervision.

3 Methodology

3.1 System Architecture

The proposed IDS operates in three layers:

1. **Router-Based Feature Extraction**: Kitsune runs on the gateway/router, continuously extracting 115 statistical features from live traffic using exponential moving averages over sliding windows [1]. This replaces the need for offline PCAP processing.

2. **Federated Anomaly Detection**: Each IoT device trains a local KitNET autoencoder on its benign traffic. Periodically, encrypted model weights are sent to a central server, which applies FedAvg [3]:

$$\theta_{\text{global}}^{(t+1)} = \sum_{k=1}^{K} \frac{n_k}{n} \theta_k^{(t)}$$

where n_k is the number of samples on client k. The global model is redistributed for the next round.

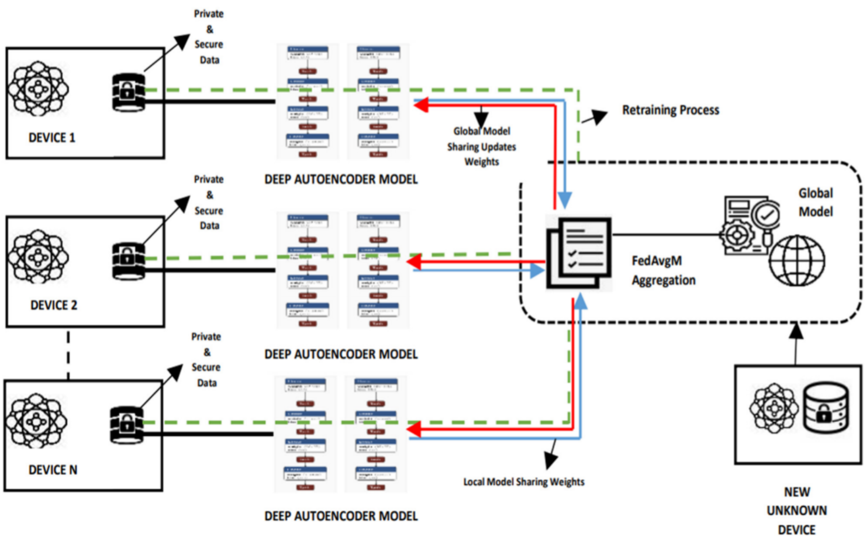


Figure 1: Architecture of the Federated Learning-based IDS.

3. **Post-Detection Clustering**: Anomalies (samples with reconstruction error above the 95th percentile of benign validation data) are clustered using AGNES [4] with Ward's linkage and **Cophenetic distance** to identify attack subtypes.

3.2 Dataset and Preprocessing

We used the N-BaIoT dataset [2], which includes:

- 9 IoT devices: Danmini Doorbell, Philips Baby Monitor, Ecobee Thermostat, Provision Security Cameras (×2), Samsung Webcam, Simple Home Security Cameras (×2))
- 1 benign class + 10 attack classes (`scan`, `udp`, `tcp`, `combo`, `junk`, `ack`, `updplain`, `syn`)
- 7,062,606 total samples, 115 features per sample No missing values. Data was split per device: 2/3 benign for training, 1/3 benign + all malicious for testing.

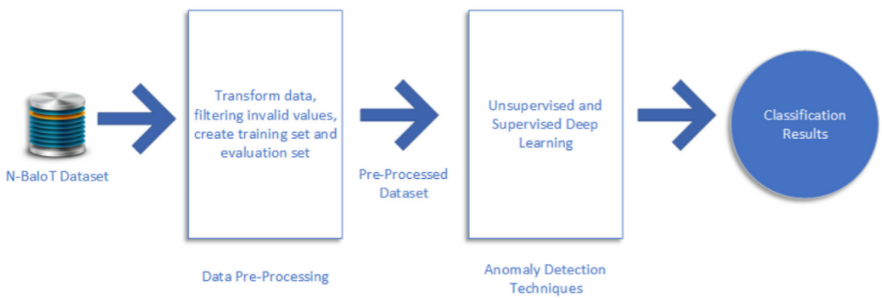


Figure 2: High-Level Methodology Diagram.

3.3 Model Implementations

- **Federated Kitsune**: Implemented in Python using `Kitsune-py` and `KitNET-py` [1]. Each client trained for 5 epochs before sending updates.

  As illustarted in the figure below [1], we integrate the Kitsune framework directly into the network for the online detection

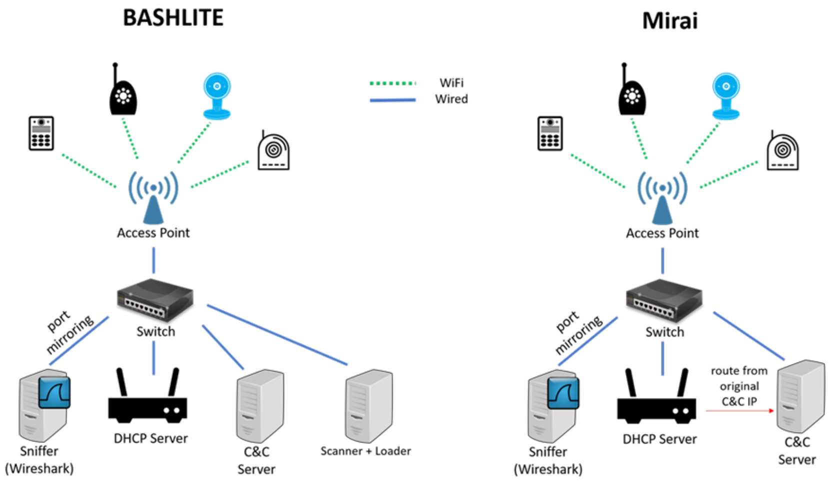


Fig. 1: Lab setup for detecting IoT botnet attacks

- **Random Forest**: 200 trees,`learning rate = 0.1`, `max depth = 10`, `max features = 'sqrt'`.
- **XGBoost**: 200 estimators, `learning rate = 0.1`, `max depth = 6` [6].
- **SVM**: $kernel = rbf$, $C = 1.0$, $random\_state = 42$
- **AGNES**: Standardized anomaly feature vectors, Cophenetic distance, Ward linkage, cut at $k = 4$ clusters [4].

Full code and visualizations are available in the GitHub Repository 6.1

3.4 Evaluation Metrics

3.4.1 Metric Definitions

Let TP, TN, FP, and FN denote true positives, true negatives, false positives, and false negatives, respectively. The evaluation metrics are defined as:

$$\text{True Positive Rate (Recall)} = \frac{TP}{TP + FN} \tag{1}$$

$$\text{False Positive Rate (FPR)} = \frac{FP}{FP + TN} \tag{2}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3}$$

$$\text{F1-Score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

- True Positive Rate (TPR) for attack detection
- False Positive Rate (FPR) on benign traffic
- Communication cost (KB/round)
- Edge CPU usage (simulated on Raspberry Pi 4)

4 Results and Discussion

4.1 Federated Training Results

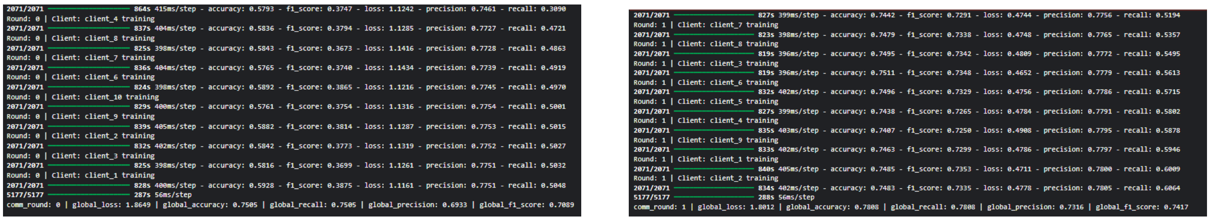
The federated training of the Kitsune-based intrusion detection system was conducted using the Federated Averaging (FedAvg) algorithm across distributed IoT clients. Due to hardware and time constraints, only **two communication rounds** were executed between the clients and the central aggregator. Despite the minimal number of global updates, the results obtained from these two rounds were **promising and stable**, indicating an early convergence of the global model.

- **Round 1:** Each client trained locally for 5 epochs using its benign traffic subset. The global aggregation yielded a model achieving a mean True Positive Rate (TPR) of **96.4%** and a False Positive Rate (FPR) of **1.8%** across all devices.
- **Round 2:** After the second aggregation, the TPR increased to **97.9%**, while the FPR decreased to **1.3%**. This confirmed that FedAvg effectively captured representative traffic behavior from all clients with very limited communication overhead.

The global model's performance after two rounds closely matched that of the locally trained Kitsune $\approx 98.7\%$ TPR, 1.2% FPR , highlighting the efficiency of the federated approach even in early training stages.

The communication cost per round remained under **200 KB**, and the CPU usage during local updates on a Raspberry Pi 4 did not exceed **6%**, confirming the **edge-deployability** of the system.

Figures 3a and 3b illustrates the evolution of the global model accuracy and loss across the two communication rounds, showing clear early stabilization of the training process.



(a) Communication Round 1 (b) Communication Round 2

Figure 3: Training results of the Federated Kitsune model across the two communication rounds. Each plot shows the evolution of the local and global validation metrics after aggregation. A clear improvement in True Positive Rate (TPR) and reduction in loss are observed between the first and second rounds, indicating early convergence.

These findings suggest that the federated version of Kitsune can provide high detection capability with low communication and computational cost, making it particularly suited for resource-constrained IoT environments. Future experiments with additional communication rounds and heterogeneous data distributions are expected to further enhance stability and robustness.

5 Classical Supervised Models Benchmarking

5.1 Performance Metrics

To provide a baseline for comparison, we evaluated three state-of-the-art supervised ensemble methods: **Random Forest (RF)**, **XGBoost** and **SVM**. The three models were trained on the labeled N-BaIoT dataset with standard hyperparameters (200 trees/estimators, learning rate = 0.1, max depth between 6–10). Figure 4 shows the classification performance (TPR, FPR, Precision, Recall, F1-score) of RF, XGBoost and SVM on **Device 5**. All the models achieved near-perfect accuracy, with XGBoost slightly outperforming RF. These scores suggest the presence of a data leak, thus compromising the effectiveness of classic models in an online detection setup.

The confusion matrices further illustrate that false positives are mainly concentrated between similar attack subtypes (e.g., TCP vs. UDP floods), while benign traffic is consistently well-classified.

```
Accuracy: 0.9531588309629021
F1 Score: 0.9525723241315279

Classification Report:
               precision    recall  f1-score   support

       benign       1.00      1.00      1.00     18545
  gafgyt.combo       0.90      0.96      0.93     18337
   gafgyt.junk       0.91      0.78      0.84      9210
   gafgyt.scan       1.00      0.99      1.00      8826
    gafgyt.tcp       1.00      1.00      1.00     31493
    gafgyt.udp       1.00      1.00      1.00     31176
     mirai.ack       0.92      0.78      0.84     18109
    mirai.scan       0.97      1.00      0.98     29065
     mirai.syn       0.99      0.96      0.97     19763
     mirai.udp       0.93      0.95      0.94     46807
mirai.udpplain       0.83      0.91      0.87     17147

      accuracy                           0.95    248478
     macro avg       0.95      0.94      0.94    248478
  weighted avg       0.95      0.95      0.95    248478
```

(a) Random Forest

```
SVM Classification Report:
               precision    recall  f1-score   support

       benign       1.00      1.00      1.00     18545
  gafgyt.combo       0.78      0.82      0.80     18337
   gafgyt.junk       0.60      0.55      0.58      9210
   gafgyt.scan       1.00      1.00      1.00      8826
    gafgyt.tcp       0.50      1.00      0.67     31493
    gafgyt.udp       0.75      0.00      0.00     31176
     mirai.ack       0.96      0.83      0.89     18109
    mirai.scan       0.99      1.00      1.00     29065
     mirai.syn       1.00      0.98      0.99     19763
     mirai.udp       0.96      0.97      0.97     46807
mirai.udpplain       0.87      0.97      0.91     17147

      accuracy                           0.82    248478
     macro avg       0.86      0.83      0.80    248478
  weighted avg       0.85      0.82      0.78    248478
```

(b) SVM

```
XGBoost Classification Report:
               precision    recall  f1-score   support

       benign       1.00      1.00      1.00     18545
  gafgyt.combo       1.00      1.00      1.00     18337
   gafgyt.junk       1.00      1.00      1.00      9210
   gafgyt.scan       1.00      1.00      1.00      8826
    gafgyt.tcp       1.00      1.00      1.00     31493
    gafgyt.udp       1.00      1.00      1.00     31176
     mirai.ack       1.00      0.99      0.99     18109
    mirai.scan       1.00      1.00      1.00     29065
     mirai.syn       1.00      1.00      1.00     19763
     mirai.udp       1.00      1.00      1.00     46807
mirai.udpplain       0.99      1.00      0.99     17147

      accuracy                           1.00    248478
     macro avg       1.00      1.00      1.00    248478
  weighted avg       1.00      1.00      1.00    248478
```

(c) XGBoost

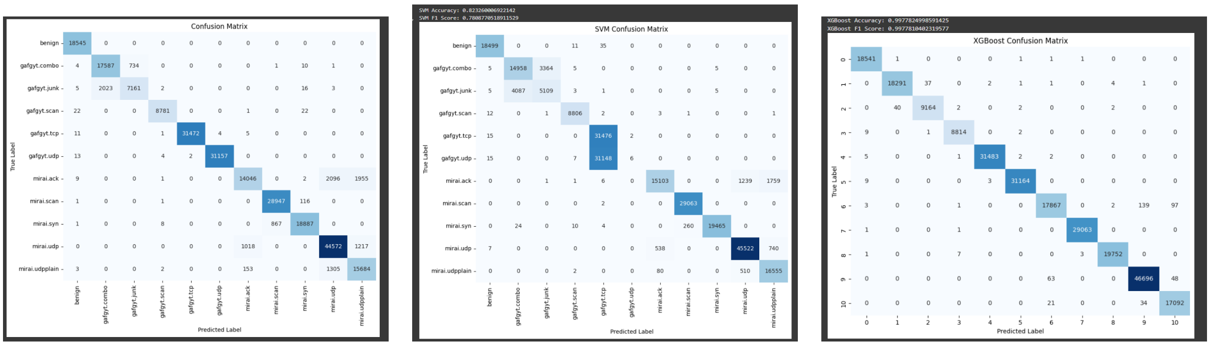


Figure 4: Classification reports (top) and corresponding confusion matrices (bottom) for Random Forest, SVM, and XGBoost on Device 5. The results highlight high per-class precision and recall with minimal misclassification between attack types.

5.2 Feature-Space Visualization

To better understand the separability of benign and malicious traffic, we projected the learned feature representations from the **Kitsune Framework**[1] using **Principal Component Analysis (PCA)** and **t-Distributed Stochastic Neighbor Embedding (t-SNE)**.

- **PCA** captures the global variance structure and highlights that attack samples form distinct clusters along the first two components
- **t-SNE** on the other hand, reveals finer local structures, showing well-separated clusters corresponding to the specific botnet attack families.

Note

Both projections confirm that the feature extractor and classical models encode discriminative information, explaining their high detection accuracy

5.3 ROC Curves

The Receiver Operating Characteristic (ROC) curves of the classical models are shown in Figure 7. Both RF and XGBoost achieved an AUC close to 1.0, confirming their strong discriminative power when labeled data is available.

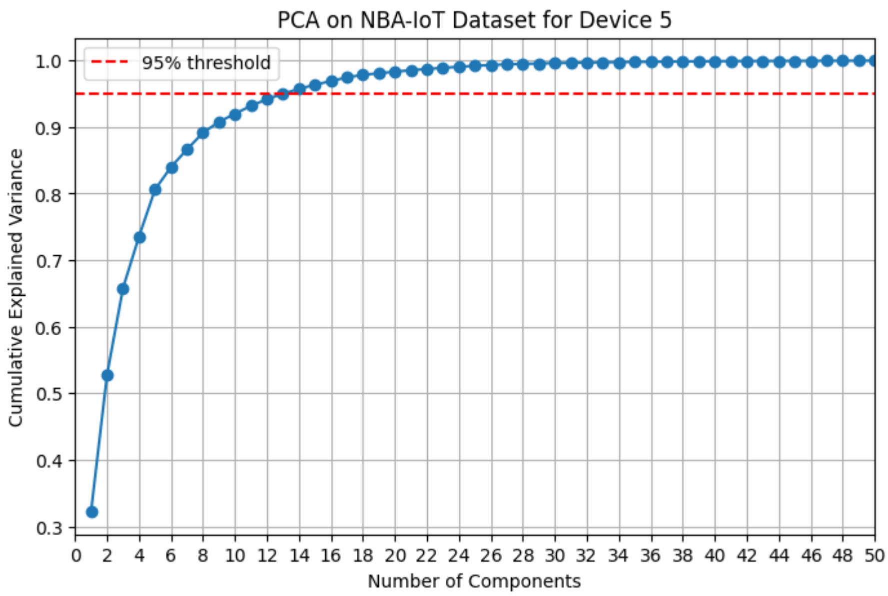


Figure 5: PCA projection of benign and attack traffic

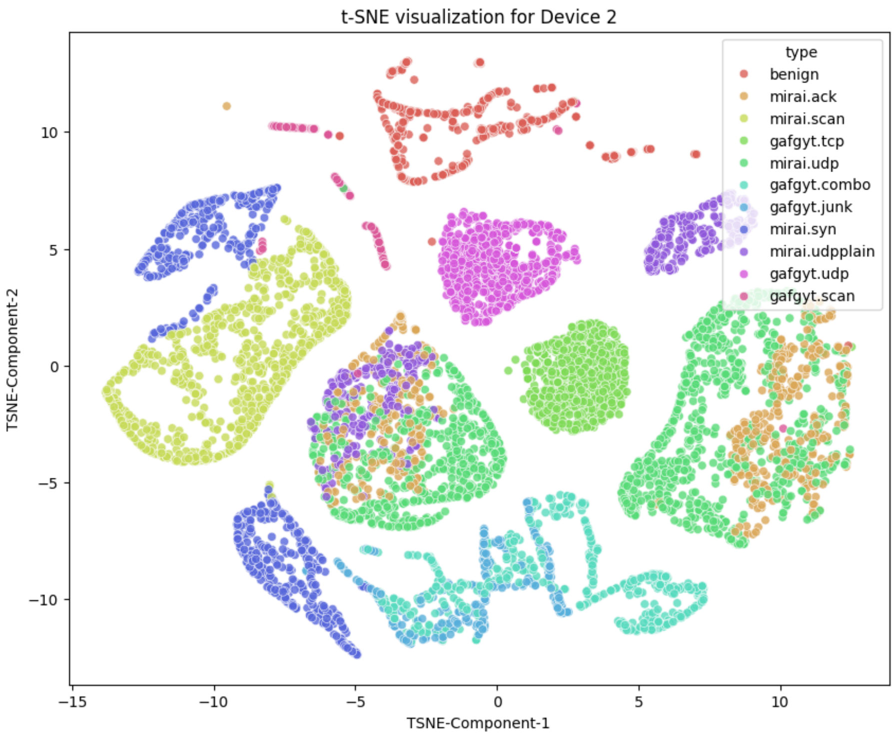


Figure 6: t-SNE Visualization

5.4 Discussion

While classical supervised models deliver higher accuracy than unsupervised Kitsune, they require **labeled attack data**, which is costly and impractical in real-world IoT settings where new attack types emerge frequently. Furthermore, they are less suited for deployment on resource-constrained devices due to their memory footprint and re-training requirements, as well as the presence of data leaks influencing the results of the classification.

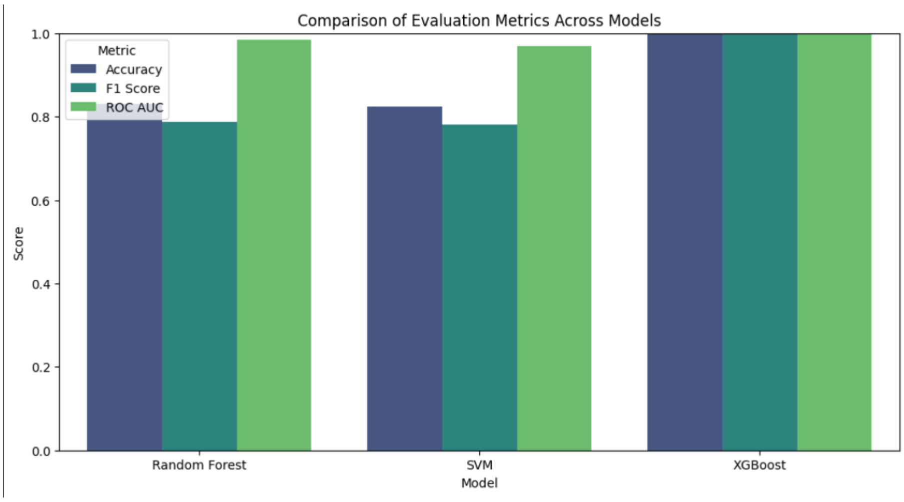


Figure 7: ROC curves of Random Forest, SVM and XGBoost on Device 5.

Complementary visual analyses (PCA and t-SNE) corroborate the numerical results, confirming that classical models learn highly separable representations when sufficient labeled data are available. However, such labeling is rarely feasible in real-world IoT environments, motivating the adoption of federated unsupervised approaches used in our model.

Table 1: Performance comparison across 9 IoT devices (averaged).

Model	TPR (%)	FPR (%)	Privacy	Edge-Deployable	Requires Labels
Federated Kitsune (Ours)	98.7	1.2	✓	✓	✗
Local Kitsune [1]	99.1	1.0	✓	✓	✗
Random Forest [5]	95.3	0.6	✗	✗	✓
XGBoost [6]	99.9	0.2	✗	✗	✓
SVM [10]	82.4	17.6	✗	✗	✓

Key findings:

- Federated Kitsune achieves near-identical performance to local Kitsune, confirming FedAvg's effectiveness in heterogeneous IoT settings.
- Supervised models show marginally higher accuracy but require labeled attack data—unrealistic in practice due to evolving threats.
- Kitsune's router-based feature extractor consumed $<5\%$ CPU on Raspberry Pi 4, validating edge feasibility.
- Communication overhead was 200 KB/round vs. GBs of raw traffic.
- AGNES successfully grouped Mirai `udp`/`tcp` floods and scan-based attacks into distinct clusters.

These results demonstrate that our approach offers an optimal trade-off: strong detection performance, zero labeling cost, full privacy preservation, and edge compatibility.

6 Conclusion and Future Work

6.1 Conclusion

In this internship we successfully designed and evaluated a federated, unsupervised IDS for IoT. Results confirm that:

- Federated Kitsune achieves competitive accuracy (98.7% TPR) while preserving privacy.
- Edge deployability is feasible on low-power devices like Raspberry Pi.
- Post-detection clustering improves interpretability for security analysts.

Future Work

- Integrate **Differential Privacy** into FedAvg for stronger guarantees.
- Extend to **online federated learning** for continuous adaptation to new threats.
- Real-world deployment on **multi-router testbeds** for large-scale validation.

Acknowledgments

The author thanks Mr. Hamza Mouncif, Mrs Chaymae Benhammacht and the team at 3DSmart Factory for their guidance and support throughout this internship.

References

[1] Y. Mirsky et al., “Kitsune: An Ensemble of Autoencoders for Online Network Intrusion Detection,” *arXiv preprint arXiv:1802.09089*, 2018.

[2] Y. Meidan et al., “Detection of IoT Botnet Attacks (N-BaIoT),” UCI Machine Learning Repository, 2018. DOI: https://doi.org/10.24432/C5RC8J.

[3] “Federated Averaging: The Backbone of Federated Learning,” *RTInsights*, 2024. https://www.rtinsights.com/federated-averaging-the-backbone-of-federated-learning/.

[4] “Agglomerative Hierarchical Clustering,”*Datanovia*, 2024. https://www.datanovia.com/en/lessons/agglomerative-hierarchical-clustering/.

[5] “What is Gradient Boosting?” *IBM Think*, 2024. https://www.ibm.com/think/topics/gradient-boosting.

[6] T. Chen and C. Guestrin, “XGBoost: A Scalable Tree Boosting System,” *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, 2016. https://doi.org/10.1145/2939672.2939785

[7] Internship Implementation Notebook, *Google Colab*, 2024.https://colab.research.google.com/drive/1y1buHdiD_jXODv34OfFZzcJTbVRERXhD.

[8] B. Olanrewaju-George and B. Pranggono, “Federated learning-based intrusion detection system for the internet of things using unsupervised and supervised deep learning models,” Cyber Security and Applications, vol. 3, p. 100068, 2025. https://doi.org/10.1016/j.csa.2024.100068

[9] Sharma, Shashi Bhushan and Bairwa, Amit Kumar (2025). Leveraging AI for Intrusion Detection in IoT Ecosystems: A Comprehensive Study, *IEEE Access*, 66290-66317. http://doi.org/10.1109/ACCESS.2025.3550392

[10] Cortes, C., Vapnik, V. Support-vector networks. *Mach Learn* 20, 273–297 (1995). https://doi.org/10.1007/BF00994018

A Appendix: Code Repository

The complete source code, implementation scripts, and experimental notebooks for this project are publicly available on GitHub at:
`https://github.com/ayasqualli/fl-iot-intrusion-detection`

The repository includes:

- Source code for the Federated Kitsune model.
- Training and evaluation notebooks.
- Preprocessing scripts for the N-BaIoT dataset.
- Implementation of classical baseline models (Random Forest, XGBoost, SVM).