A Feature-aware Adaptive Ensemble Framework for IoT Intrusion Detection Systems

Youssef Laraig *†, Yann Ben Maissa*, Sébastien Roy†, Pierre-Martin Tardif†, Brahim El bhiri‡
*National Institute of Posts and Telecommunications (INPT), STRS Laboratory, Rabat, Morocco
†University of Sherbrooke, Sherbrooke QC, Canada
‡Harmony Technology, Morocco
youssef.laraig@usherbrooke.ca, benmaissa@inpt.ac.ma, sebastien.roy13@usherbrooke.ca,
pierre-martin.tardif@usherbrooke.ca, b.elbhiri@harmony.ma

***Abstract*—Intrusion Detection Systems (IDS) are essential for Internet of Things (IoT) security, but single models often fail due to IoT data heterogeneity. While machine learning ensembles combine complementary strengths, conventional static weighting schemes, such as majority voting and temporal stacking, do not adapt to sample-specific features and may underperform in diverse IoT scenarios. To address these limitations, we propose a dynamic feature-weighting ensemble framework for intrusion detection in IoT networks that combines adaptive weighting with a selected set of complementary base models suited to different traffic patterns. The approach combines four complementary models: Gradient Boosted Trees (LightGBM), Bagging-based Random Forest (RF), Instance-based k-Nearest Neighbors (k-NN), and Deep Feedforward Neural Networks (FNN). It dynamically adjusts their weights based on the active features of each incoming traffic flow, emphasizing models best suited to specific patterns (e.g., LightGBM for packet-header patterns, FNN for nonlinear TCP flag interactions). Evaluated on the CICIoT2023 dataset, the framework achieved 99.95% precision and 98.59% recall, resulting in a 73%-97% reduction in false positives (FP) and a 9%-30% reduction in false negatives (FN) compared to individual models.**

***Index Terms*—Intrusion Detection, Internet of Things, Ensemble Learning, Dynamic Feature Weighting, Context Awareness, Wireless Traffic Analysis**

I. Introduction

Context. The Internet of Things has become pervasive in sensitive areas like healthcare, industrial automation, and smart city infrastructure [1]. This, however, seems to lag behind the implementation of adequate security mechanisms. The diverse device architectures, their low computing power, and emerging threats have made IoT networks vulnerable [2]. Additionally, due to the lack of adaptability to IoT environments, traditional Intrusion Detection Systems (IDS) are inadequate for precision-recall balancing, which poses operational challenges such as high false positive rates and undetected attacks [3].

Motivation. Machine learning (ML)-based IDSs improve the detection of new threats [4], but single models remain limited. IoT networks include diverse devices and protocols, generating heterogeneous traffic with complex attacks at various anomaly levels. Tree-based models like LightGBM and Random Forest accurately detect packet-level patterns but struggle with encrypted or temporal features [5]. Deep neural networks (DNN) capture complex relations but lack interpretability [6], while K-Nearest Neighbors are interpretable but sensitive to data scale and dimensionality [7]. No single "silver bullet" model handles this diversity well, and relying on one risks bias and variance. Ensembles help, but most use static aggregation like stacking or majority voting, which poorly adapts to evolving IoT traffic and attacks [8], [9], limiting false positive/negative tradeoff optimization and reducing reliability [10].

Contribution. We propose a dynamic ensemble framework that adjusts model contributions based on incoming flow characteristics. Through feature importance analysis, we studied models' expertise in distinct domains, such as LightGBM for packet-header statistics and FNNs for temporal TCP flag patterns. Unlike static ensembles that use majority voting [11] or global weighting like attention-based methods [12], we dynamically assign more weight to models whose strengths better align with the traffic's active features. We merge these models to address the diverse feature space of IoT traffic. The approach was evaluated on the CICIoT2023 dataset [13], focusing on minimizing false positives and false negatives. Early results show promising improvements, though further validation is needed on other datasets. By integrating diverse models, each with its strengths, we gain useful diversity, and aligning each model's contribution with specific traffic features improves interpretability [3].

Contents. The rest of the paper is organized as follows: Section II reviews related work on IDSs and ensemble approaches. Section III outlines our methodology. Section IV presents the experimental results and some insights. Section V concludes the paper.

II. Related Work

Ensemble-based Intrusion Detection Systems have been used to address diverse IoT attack types. Early works relied on static ensembles like majority voting and stacking. For instance, Alotaibi and Ilyas [14] combined RF, Decision Tree, Logistic Regression, and k-NN through stacking and voting, achieving high accuracy on the TON-IoT dataset. Hossain and Islam [15] applied a stacked ensemble of RF, XGBoost, and other classifiers with a Logistic Regression meta-model, reaching 99% accuracy, but with added complexity due to meta-learner training.

Static ensembles generally apply uniform weighting, limiting their ability to emphasize models that better capture

979-8-3503-9281-4/25/$31.00 ©2025 IEEE

specific attack patterns, such as protocol-based anomalies. Kumar et al. [11] observed this drawback in their fog-cloud Internet of Medical Things (IoMT) architectures using majority voting. Verma and Ranga [16] proposed a resampling-based ELNIDS (Ensemble Learning-Based Network Intrusion Detection System). While sometimes effective, resampling introduced bias toward minority-class detection, problematic in heterogeneous IoT traffic.

Recent research has turned to dynamic weighting to improve model adaptability. Wardana et al. [17] used Differential Evolution to tune ensemble weights, but their Weighted Averaging Deep Neural Network (WEA-DNN) was computationally demanding. Similarly, metaheuristic approaches such as Brain Storm Optimization (BSO) [18] adjust classifier weights iteratively, making them less suitable for real-time IoT applications. Deep learning-based ensembles also showed promise: the Deep Neural Network-Long Short-Term Memory (DNN-LSTM) model by Dutta et al. [19] improved detection but at the cost of latency, while Thakkar and Lohiya's hybrid DNN–bagging model [20] struggled with limited computational resources, a common constraint in IoT systems.

Challenges related to class imbalance and interpretability remain. Verma et al. [21] addressed imbalance using a combination of Random Forest and Gradient Boosting with undersampling techniques, but their reliance on static weighting limited the adaptability of the approach. Alshehri et al. [12] introduced a Self-Attention Deep Convolutional Neural Network (SA-DCNN) to dynamically weight features, but did not explicitly capture localized anomalies like those in encrypted payloads.

Despite this progress, a significant gap remains in how most ensemble approaches handle feature diversity. Static or global weighting schemes often overlook the varying relevance of features across different samples and carefully chosen base models, limiting adaptability in dynamic environments like IoT networks. This gap in effectively leveraging feature diversity, while choosing complementary base models, is where our inspiration comes from. Our work proposes a dynamic feature-weighted ensemble framework to address this issue.

III. Methodology

This section first presents a "big picture" of the framework, then details dataset preprocessing, motivation behind the models and weighting mechanism, as well as validation metrics (for reproducibility).

A. Framework Overview

The proposed dynamic feature-weighting ensemble framework for IoT intrusion detection is shown in Fig. 1. The raw IoT traffic data is first preprocessed: cleaned, normalized, and scaled, to ensure consistency. This data is then fed into four complementary models: LightGBM, Random Forest, Feedforward Neural Network, and k-Nearest Neighbors, each trained separately to maintain decision diversity.

Feature importance is analyzed for each model to highlight which inputs most influence its predictions. These insights guide dynamic, instance-specific weight assignment during inference, based on how well each model aligns with the input's feature profile.

Finally, the ensemble module combines the weighted outputs to deliver the intrusion detection result.

B. Data Preprocessing and Feature Engineering

We use the CICIoT2023 dataset from the Canadian Institute for Cybersecurity (CIC) [13], selected for its realistic IoT traffic, reproducibility, and diverse attack coverage. The dataset captures traffic among 105 real IoT devices in a lab environment, including 33 different attack types grouped into seven categories: Distributed Denial of Service (DDoS), Denial of Service (DoS), Reconnaissance, Web-based, Brute Force, Spoofing, and Mirai. Each network flow is labeled as benign or malicious, making it suitable for evaluating intrusion detection methods in varied IoT scenarios.

1) Dataset Overview

Table I provides a summary of the dataset's key characteristics and distribution, where each sample includes 39 features and a label $y \in \{0, 1\}$, where 0 represents benign traffic and 1 indicates an attack.

TABLE I: Dataset Overview and Class Distribution

Metric	Training Set	Testing Set
Total Samples	1,748,197	
Number of Features	39	
Normal Flows	659,602	282,682
Attack Flows	686,001	119,912
Normal:Attack Ratio	1:1.0	1:0.4

2) Class Distribution

The dataset's class distribution reflects the imbalance commonly seen in real-world IoT networks. The test set contains 70.2% normal traffic and 29.8% attacks, challenging detection systems to remain effective under imbalanced conditions.

3) Preprocessing Pipeline

The steps we followed are:

Cleaning. For each feature ($j \in \{1, \ldots, 39\}$), missing or infinite values are replaced with the median $\tilde{x}_j$ from the training set:

$$x_{i,j} = \begin{cases} \tilde{x}_j & \text{if } x_{i,j} = \pm\infty \text{ or NaN} \\ x_{i,j} & \text{otherwise} \end{cases}$$

Scaling. Features used by FNN and k-NN models are standardized as:

$$\tilde{x}_{i,j} = \frac{x_{i,j} - \mu_j}{\sigma_j}$$

where μ_j and σ_j are the mean and standard deviation of feature j from training data. Tree-based models (e.g., Random Forest, LightGBM) use unscaled values.

C. Base Models Architecture and Specialization

All base models were configured using standard parameters inspired by prior work and library defaults, since our focus is on ensemble dynamics rather than hyperparameter tuning.

1) Random Forest (RF) Model

a) Model Overview and Motivation

We adopt the Random Forest classifier due to its robustness to noise and ability to generalize across heterogeneous unen-

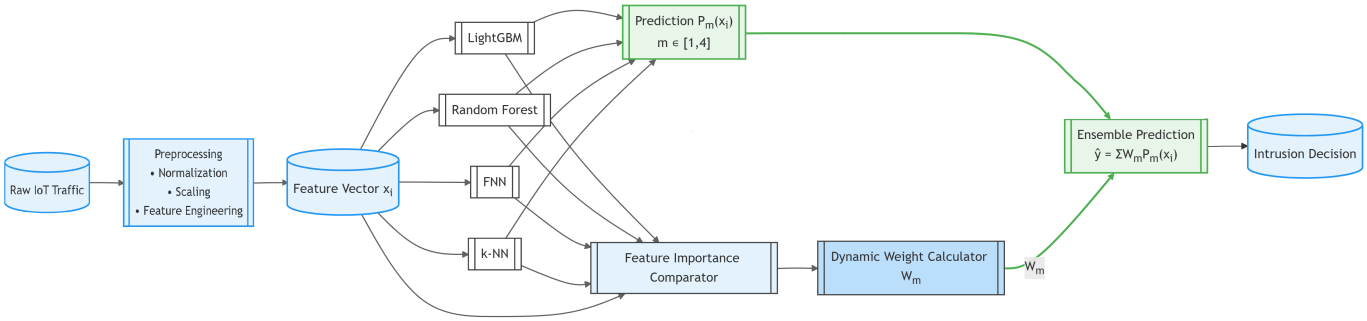


Fig. 1: High-Level Architecture of the Proposed Adaptive Ensemble-Based Intrusion Detection Framework

crypted IoT traffic. The model aggregates predictions from all T decision trees via majority voting:

$$\hat{y}(x) = \text{majority vote}\left(\{h_1(x), \ldots, h_T(x)\}\right), \qquad (1)$$

where x is the input vector with 39 features, and each tree minimizes Gini impurity.

b) Feature Importance and Specialization

Figure 2 shows Random Forest's top features by Mean Decrease in Impurity (MDI), emphasizing reliance on network-layer metadata such as:

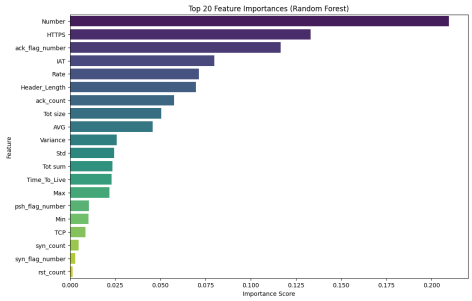


Fig. 2: Top 20 Features by Mean Decrease in Impurity (RF)

- `Number`: Packets per flow, linked to DDoS activity.
- `Rate`: Packets per second, flags fast traffic spikes.
- `HTTPS`: Encrypted session pattern recognition.
- `ack_flag_num`: TCP flags signals protocol misuse.
- `IAT`: Detects timing anomalies between packets.

RF is prioritized in the ensemble when network-layer indicators dominate (e.g., volumetric DoS/DDoS), but its weight is reduced on payload-intensive flows, where models like FNN perform better. Payload features showed low importance (MDI < 0.005), reinforcing RF's focus on statistical metadata.

2) LightGBM (LGBM)

a) Model Overview and Motivation

We chose LightGBM for its fast, accurate handling of large, high-dimensional IoT traffic. Its ability to capture nonlinear network patterns suits complex attack detection, and its memory efficiency supports future IoT/edge IDS deployment.

The model minimizes binary cross-entropy loss by:

$$L(y, \hat{y}) = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)\right],$$

where $y_i \in \{0, 1\}$ is the true class label and $\hat{y}_i \in [0, 1]$ is the predicted probability.

b) Feature Importance and Specialization

Figure 3 shows LightGBM's split-based feature importance, highlighting its strength in detecting packet-level and statistical anomalies. Key features include:

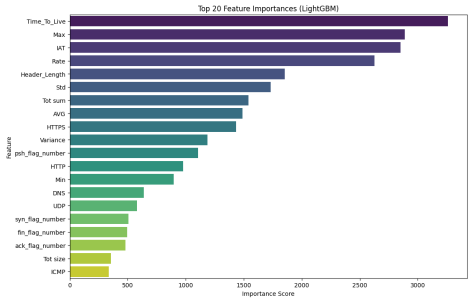


Fig. 3: LightGBM split-based feature importance

- `Time_To_Live`: Flags abnormal TTL (e.g., spoofing).
- `Max` (packet size): Detects irregular packets.
- `Inter-Arrival Time (IAT)`: Captures irregular traffic timing.
- `HTTPS`: Monitors encrypted session metadata.

LightGBM is dynamically favored in the ensemble when such anomalies are dominant, complementing Random Forest's focus on flow-level patterns. This division of specialization improves coverage of encrypted threats (e.g., TLS session hijacking) and enhances ensemble diversity and precision.

3) Feedforward Neural Network (FNN)

a) Model Overview and Motivation

The FNN models complex nonlinear relationships, especially temporal sequences in IoT traffic. Its sequential architecture is simpler than deep convolutional or recurrent networks, making it suitable for deployment on IoT/edge devices with hard resource constraints.

Training minimizes the binary cross-entropy loss:

$$L(y, \hat{y}) = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)\right],$$

where $y_i \in \{0, 1\}$ are true labels and $\hat{y}_i \in [0, 1]$ are the predicted probabilities.

b) Feature Importance and Specialization

Figure 4 shows the FNN's top features via permutation importance, highlighting sensitivity to complex and encrypted patterns:

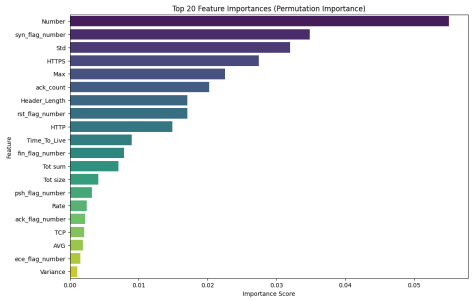


Fig. 4: Permutation-Based Feature Importance in the FNN

- **Number:** Total packets per flow, identifies volumetric patterns (e.g., DDoS).
- **syn_flag_number:** TCP SYN flag counts, detects SYN flood via flag ratios.
- **Std:** Packet size variation, flags polymorphic malware.
- **HTTPS:** TLS metadata, captures forged certificates or handshake anomalies.

The FNN is emphasized in the ensemble when nonlinear or cross-feature patterns emerge, such as multi-stage attacks involving SYN spikes and payload irregularities. Its weight increases on encrypted flows, while tree models handle simpler, metadata-driven anomalies.

4) K-Nearest Neighbors (k-NN)

a) Model Overview and Motivation

k-NN excels at recognizing rare localized deviations and anomalies in IoT traffic through instance-based learning. As a non-parametric method, it is *agnostic* to distributional assumptions, making it robust to heterogeneous data.

The distance between a sample x and a neighbor x_i is computed as:

$$d(x, x_i) = \sqrt{\sum_{j=1}^{m} (x_j - x_{i,j})^2},$$

where m is the number of features. Classification is based on majority voting among the k nearest neighbors.

b) Feature Importance and Specialization

While k-NN lacks built-in feature weights, permutation importance (Fig. 5) reveals its sensitivity to protocol-level patterns.

- **Time_To_Live:** Flags routing anomalies like spoofing.
- **ack_count:** Indicates abnormal TCP state use (e.g., ACK floods).
- **Number:** Captures volumetric shifts via packet counts.
- **rst_flag_num:** Detects scan-induced connection resets.

In the ensemble, k-NN excels at identifying local or protocol-specific deviations, especially near known attack clusters. It's weighted higher in such cases, offering fine-grained validation, while tree-based models dominate on broader volumetric attacks.

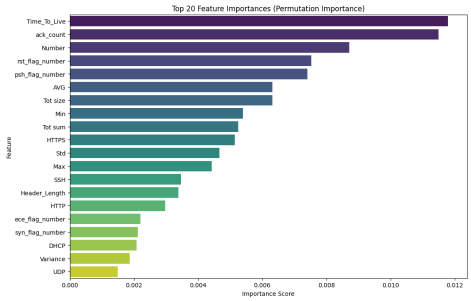


Fig. 5: Permutation-Based Feature Importance for k-NN

D. Dynamic Weighting Mechanism

To adapt to varying IoT traffic, the ensemble dynamically adjusts model weights based on feature relevance per input.

Step 1: Feature normalization. Each feature x_f is scaled to $[0, 1]$ to remove scale bias:

$$\tilde{x}_f = \frac{x_f - \min(f)}{\max(f) - \min(f)}$$

Step 2: Importance scaling. Raw feature importance values are normalized using percentile ranking.

$$w_{m,f} = \begin{cases} \frac{\text{rank}_m(f)}{F}, & f \in m \\ 0, & \text{otherwise} \end{cases}$$

Step 3: Relevance scoring. Each model's score is the weighted sum over active features:

$$\text{Relevance}_m = \sum_{f=1}^{F} w_{m,f} \cdot \tilde{x}_f$$

Step 4: Weighting. Normalize scores to get weights α_m:

$$\alpha_m = \frac{\text{Relevance}_m}{\sum_{k=1}^{M} \text{Relevance}_k}$$

Step 5: Final Prediction. The ensemble output is a weighted sum:

$$p_{\text{ens}} = \sum_{m=1}^{M} \alpha_m \cdot p_m \quad \Rightarrow \quad \hat{y} = \begin{cases} 1 & \text{if } p_{\text{ens}} \geq 0.5 \\ 0 & \text{otherwise} \end{cases}$$

This feature-aware mechanism dynamically adapts model weights per input, ensuring context-awareness and interpretability by prioritizing models relevant to active features.

E. Evaluation Metrics

Given class imbalance (29.8% attacks, 70.2% normal), we emphasize metrics that favor attack detection while limiting false alarms.

Recall (Sensitivity):

$$\text{Recall} = \frac{TP}{TP + FN}$$

Proportion of actual attacks correctly identified.

Precision:

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

Proportion of predicted attacks that are truly malicious.

F1-Score:

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Balances recall and precision, penalizes missing attacks.

Balanced Accuracy:

$$\text{Balanced Acc.} = \frac{1}{2}\left(\frac{TP}{TP+FN} + \frac{TN}{TN+FP}\right)$$

Accounts for both classes, reducing imbalance bias.

We also compute the confusion matrix and Precision vs. Recall Curve to assess FP/FN tradeoffs.

IV. Experimental Results and Insights

In this section, we validate our approach and discuss the implications of the simulation results.

A. Dynamic Weight Assignment and Instance-Level Analysis

To demonstrate the effectiveness of the dynamic weighting mechanism, we present the weight allocation for early instances in Fig. 6 (generated by our framework) and analyze two representative flows.

```
Dynamic Weights for the First Instances:
Instance   0 - LGBM: 0.242800, RF: 0.377263, FNN: 0.191198, KNN: 0.188740
Instance   1 - LGBM: 0.191717, RF: 0.434395, FNN: 0.228180, KNN: 0.145708
Instance   2 - LGBM: 0.096941, RF: 0.248546, FNN: 0.287747, KNN: 0.366765
Instance   3 - LGBM: 0.228052, RF: 0.359167, FNN: 0.169923, KNN: 0.242859
Instance   4 - LGBM: 0.163432, RF: 0.487038, FNN: 0.248854, KNN: 0.100676
Instance   5 - LGBM: 0.097058, RF: 0.248600, FNN: 0.431487, KNN: 0.222856
Instance   6 - LGBM: 0.192075, RF: 0.445624, FNN: 0.224705, KNN: 0.137596
Instance   7 - LGBM: 0.254970, RF: 0.318561, FNN: 0.191354, KNN: 0.235115
```

Fig. 6: Dynamic Models' Weight Distribution on Initial Flows

1) Interpretation of Instances

a) Instance 3

This flow corresponds to a DDOS-RSTFINFLOOD attack, marked by high packet count and frequent TCP reset (RST) and finish (FIN) flags, signs of repeated connection endings. RF model gets the highest weight (36%), showing its strength in detecting traffic spikes and TCP flag anomalies. LGBM and k-NN contribute by identifying packet header irregularities and flow statistics. The FNN has a smaller weight as the attack features are simple and well detected by tree-based models.

b) Instance 5

This flow is a DDOSSYNFLOOD attack, marked by a very high number of TCP SYN flags starting connections, almost no ACK or FIN flags, and very short packet intervals. These are typical signs of a SYN flood attack. The ensemble assigns the largest weight (43%) to the FNN because it can detect nonlinear interactions in TCP flag patterns and subtle timing irregularities that tree-based models miss. RF and k-NN have moderate influence, while LGBM is down-weighted due to its focus on packet header statistics, which is less effective here.

As a consequence, by dynamically adapting weights, the ensemble leverages tree-based models for spotting traffic spikes and protocol deviations, NNs for complex temporal patterns, k-NN for similarity recognition between instances.

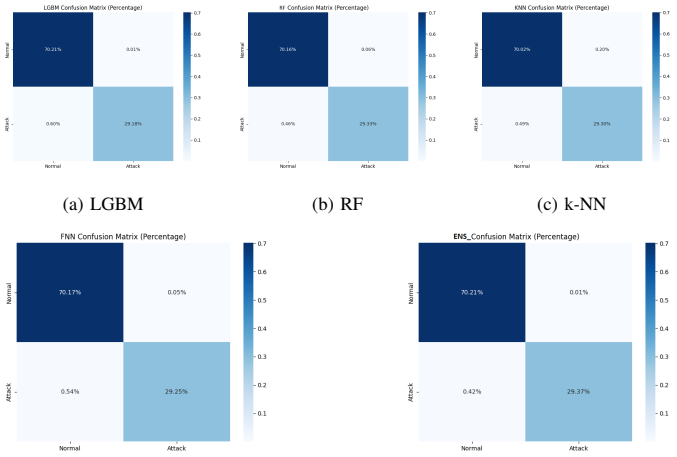
B. Overall Performance Comparison

As shown in Table II, no single base model dominates all metrics. LGBM achieves the highest precision (0.9998), indicating strong ability to avoid false alarms, but its lower recall (0.980) suggests it misses more actual attacks. FNN shows consistency: a better balance between precision (0.9984) and recall (0.982). RF and KNN improve recall (0.985 and 0.984) by detecting more attacks, but at the cost of reduced precision due to more false alarms. The ensemble combines the strengths of all models, achieving the best balance overall: highest balanced accuracy (0.993), great precision (0.9995), improved recall (0.986), low false positive and negative rates.

TABLE II: Performance of Individual Models and Ensemble

Model	Balanced Acc	Prec	Recall	FPR(%)	FNR(%)
LGBM	0.990	0.9998	0.980	0.018	0.85
RF	0.992	0.9980	0.985	0.196	0.54
KNN	0.990	0.9933	0.984	0.673	1.84
FNN	0.991	0.9984	0.982	0.060	0.43
Ensemble	0.993	0.9995	0.986	0.047	0.59

C. Confusion Matrix and Precision–Recall Curve Analysis



(a) LGBM (b) RF (c) k-NN

(d) FNN (e) Ensemble

Fig. 7: Confusion Matrices of Base Models and Ensemble

All models maintain high true positive and true negative values, correctly identifying most attack and normal traffic. However, LGBM has a higher false negative (0.60%), missing more attacks, RF and FNN still suffer false positives (0.06% and 0.046%), and k-NN performs worse. The ensemble effectively balances this trade-off, significantly reducing both false positives (0.01%) and false negatives (0.42%) compared to most individual models.

This confirms that combining diverse model strengths improves overall classification accuracy and error reduction. The ensemble's Precision-Recall curve (Fig 8) further illustrates consistently high precision across almost the full recall range.

D. Discussion & Insights

This analysis delivers four key insights: (1) Dynamic expert selection; (2) Operational reliability; (3) Flow-adaptive IDS; (4) Improved interpretability.

First, the dynamic expert selection is displayed in Fig. 6, where the ensemble changes model weights based on flow characteristics, leveraging strengths of each one. For example, tree-based models are more used in attacks with clear traffic spikes (Instance 3), while the FNN dominates in detecting

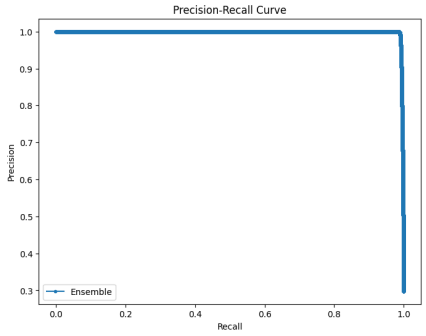


Fig. 8: Precision–Recall curve of the ensemble model.

nonlinear complex patterns in the attacks (Instance 5).

Second, the table II and the confusion matrices in Fig. 7 show that our strategy significantly reduces the error. Indeed, the ensemble achieves a balanced accuracy of 0.993 and precision of 0.9995. This highlights the *operational reliability* we gained through model combination.

Third, our approach's ability to adapt its weighting in real time, without manual intervention, transforms a traditionally static IDS into a flow-adaptive one suitable for evolving IoT networks. This is critical for real world settings with rapid attack-vectors changing quickly.

Finally, the weighting mechanism also improves *interpretability*. By measuring the contribution of each model per instance, our approach moves beyond the black-box paradigm, as the ensemble offers transparency on which features and models drive the decision. This is important for delivering insight into an attack to a security expert.

V. Conclusion

Summary. This paper introduces a dynamic ensemble framework for intrusion detection tailored to the heterogeneous and evolving nature of IoT network traffic. Unlike traditional static decision rules, our approach dynamically adjusts the influence of each model according to the relevant features of the incoming data, enabling a more context-sensitive and precise detection process. This flexibility allows the IDS to better adapt to diverse traffic types and emerging attack behaviors, which is critical in IoT environments characterized by complex and varied anomaly patterns. By carefully selecting four complementary base models – Random Forest, LightGBM, Feed-forward Neural Network, and k-NN – each targeting distinct IoT traffic patterns, and using their strengths through feature-aware dynamic weighting, our approach improves detection accuracy while enhancing interpretability and robustness of IoT IDS solutions.

Evaluated on the CICIoT2023 dataset, our framework achieved good results with a precision of 99.95% and a recall of 98.59%. It significantly reduced false positives by 73-97% and false negatives by 9-30% compared to individual models, highlighting its effectiveness in maintaining reliable security.

Future works. We believe a natural extension of our framework for long-term adaptation is to integrate online learning, where models and feature importances are updated incrementally on-the-fly on streaming data.

References

[1] Statista, "Number of connected iot devices worldwide from 2019 to 2030," 2025, available: https://www.statista.com/statistics/1194688/iot-connected-devices-communications-technology/, Accessed: June 2025.

[2] I. Butun, P. Österberg, and H. Song, "Security of the internet of things: Vulnerabilities, attacks, and countermeasures," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 616–644, 2019.

[3] S. Hajiheidari, K. Wakil, M. Badri, and N. J. Navimipour, "Intrusion detection systems in the internet of things: A comprehensive investigation," *Computer Networks*, vol. 160, pp. 165–191, 2019.

[4] S. M. Tahsien, H. Karimipour, and P. Spachos, "Machine learning based solutions for security of internet of things (iot): A survey," *Journal of Network and Computer Applications*, vol. 161, p. 102630, 2020.

[5] J. B. Awotunde, S. O. Folorunso, A. L. Imoize, J. O. Odunuga, C.-C. Lee, C.-T. Li, and D.-T. Do, "An ensemble tree-based model for intrusion detection in industrial internet of things networks," *Applied Sciences*, vol. 13, no. 4, p. 2479, 2023.

[6] B. Sharma, L. Sharma, C. Lal, and S. Roy, "Explainable artificial intelligence for intrusion detection in iot networks: A deep learning based approach," *Expert Systems with Applications*, vol. 238, p. 121751, 2024.

[7] M. Mohy-Eddine, A. Guezzaz, S. Benkirane, and M. Azrour, "An efficient network intrusion detection model for iot security using k-nn classifier and feature selection," *Multimedia Tools and Applications*, vol. 82, no. 15, pp. 23 615–23 633, 2023.

[8] A. H. Farooqi, S. Akhtar, H. Rahman, T. Sadiq, and W. Abbass, "Enhancing network intrusion detection using an ensemble voting classifier for internet of things," *Sensors*, vol. 24, no. 1, p. 127, 2023.

[9] R. Ahmad, I. Alsmadi, W. Alhamdani, and L. Tawalbeh, "A deep learning ensemble approach to detecting unknown network attacks," *Journal of Information Security and Applications*, vol. 67, p. 103196, 2022.

[10] J. Mijalkovic and A. Spognardi, "Reducing the false negative rate in deep learning based network intrusion detection systems," *Algorithms*, vol. 15, no. 8, p. 258, 2022.

[11] P. Kumar, G. P. Gupta, and R. Tripathi, "An ensemble learning and fog-cloud architecture-driven cyber-attack detection framework for iomt networks," *Computer Communications*, vol. 166, pp. 110–124, 2021.

[12] M. S. Alshehri, O. Saidani, F. S. Alrayes, S. F. Abbasi, and J. Ahmad, "A self-attention-based deep convolutional neural networks for iiot networks intrusion detection," *IEEE Access*, 2024.

[13] E. C. P. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu, and A. A. Ghorbani, "Ciciot2023: A real-time dataset and benchmark for large-scale attacks in iot environment," *Sensors*, vol. 23, no. 13, p. 5941, 2023.

[14] Y. Alotaibi and M. Ilyas, "Ensemble-learning framework for intrusion detection to enhance internet of things' devices security," *Sensors*, vol. 23, no. 12, p. 5568, 2023.

[15] M. A. Hossain and M. S. Islam, "Ensuring network security with a robust intrusion detection system using ensemble-based machine learning," *Array*, vol. 19, p. 100306, 2023.

[16] A. Verma and V. Ranga, "Elnids: Ensemble learning based network intrusion detection system for rpl based internet of things," in *2019 4th International conference on Internet of Things: Smart innovation and usages (IoT-SIU)*. IEEE, 2019, pp. 1–6.

[17] A. A. Wardana, G. Kołaczek, A. Warzyński, and P. Sukarno, "Collaborative intrusion detection using weighted ensemble averaging deep neural network for coordinated attack detection in heterogeneous network," *International Journal of Information Security*, vol. 23, no. 5, pp. 3329–3349, 2024.

[18] A. Maheshwari, B. Mehraj, M. S. Khan, and M. S. Idrisi, "An optimized weighted voting based ensemble model for ddos attack detection and mitigation in sdn environment," *Microprocessors and Microsystems*, vol. 89, p. 104412, 2022.

[19] V. Dutta, M. Choraś, M. Pawlicki, and R. Kozik, "A deep learning ensemble for network anomaly and cyber-attack detection," *Sensors*, vol. 20, no. 16, p. 4583, 2020.

[20] A. Thakkar and R. Lohiya, "Attack classification of imbalanced intrusion data for iot network using ensemble-learning-based deep neural network," *IEEE Internet of Things Journal*, vol. 10, no. 13, pp. 11 888–11 895, 2023.

[21] P. Verma, A. Dumka, R. Singh, A. Ashok, A. Gehlot, P. K. Malik, G. S. Gaba, and M. Hedabou, "A novel intrusion detection approach using machine learning ensemble for iot environments," *Applied Sciences*, vol. 11, no. 21, p. 10268, 2021.