Comparison of Multi-Expert Ensemble Feature Selection with XAI Explainability for IoT Intrusion Detection on Edge Computing

Trang-Linh Le Thi[1], Minh-Tuan Dang[2], Minh-Hoang Nguyen[3], Trong-Minh Hoang[4*]
[1]Electric Power University, Hanoi, Vietnam
Email: linhltt@epu.edu.vn
[2]CMC University, Hanoi, Vietnam
Email: dmtuan@cmcu.edu.vn
[3]University of Engineering and Technology, VNU, Hanoi, Vietnam
Email: 22025003@vnu.edu.vn
[4]Posts and Telecommunications Institute of Technology, Hanoi, Vietnam
Email: hoangtrongminh@ptit.edu.vn

***Abstract*—This paper presents a comprehensive evaluation of four ensemble methods (Majority Voting Method, AdaBoost Algorithm, Stacking Ensemble Method, and Bayesian Decision Method) in multi-expert feature selection systems, constructed from five base experts including SelectKBest, Recursive Feature Elimination, Random Forest, L1-based Selection (Lasso), and Mutual Information, in the context of intrusion detection for Internet of Things (IoT) systems. We utilize the CIC IoT 2023 dataset, a large-scale real-world dataset that reflects diverse attack scenarios in IoT environments. The highlight of this research is the integration of Explainable Artificial Intelligence (XAI) methods to analyze the influence of each expert and feature on the ensemble method results. Additionally, the study conducts multi-criteria comparisons including performance metrics (accuracy, per-class precision, recall, F1-score), computational efficiency (training time, response time, memory usage), and comprehensive classification performance indicators (ROC-AUC, PR-AUC, false positive rate). The experimental results provide in-depth analyses of the advantages and limitations of each method, thereby offering recommendations for selecting the most appropriate ensemble method according to specific IoT system requirements. This research contributes to enhancing the effectiveness and transparency of intrusion detection systems in complex IoT environments.**

***Index Terms*—Internet of Things, Intrusion Detection System, CIC IoT 2023, Feature Selection, Multi-expert system, Ensemble Method**

I. Introduction

Contemporary Internet of Things (IoT) solutions have marked a significant advancement in the technology domain, enabling the integration of diverse devices into interconnected networks serving various aspects of human life [1]. However, alongside these apparent benefits, this diverse integration also presents security challenges due to the complexity, large scale, and critical role of IoT ecosystems [2]. Particularly, attacks and security breaches in IoT systems have severe impacts, directly affecting both digital and physical domains [3]. Therefore, implementing highly effective security solutions has increasingly become a crucial component in modern network solutions [4]. The massive volume and diversity of IoT data necessitate stringent criteria for feature selection. In edge-deployed intrusion detection systems (IDS), feature selection enhances detection accuracy, minimizes computational costs, and facilitates real-time processing [5]. Nevertheless, practical IDS solutions are limited by the attributes of intricate and decentralized IoT systems, needing to ongoing investigation into more efficient and versatile feature selection techniques.

Feature Selection (FS) is an important preprocessing phase in machine learning, designed to select representative subsets from high-dimensional datasets. Filter, wrapper, embedded, and hybrid methodologies each possess distinct advantages; nonetheless, their efficacy is significantly influenced by data attributes, model selection, and available system resources [6]. Multi-expert FS systems have thus arisen to use the complementary benefits of several methodologies. Nevertheless, the majority of ensemble processes continue to operate as "black boxes," exhibiting a lack of openness regarding the contributions of experts to the final outcomes. Thus, augmenting interpretability and reliability in these systems is vital.

This research addresses these shortcomings by performing a thorough comparative evaluation of aggregation techniques in multi-expert feature selection for IoT intrusion detection. Our primary contributions are:

- Utilising XAI to provide insight into result aggregation methods in multi-expert FS systems.
- Conducting a multi-criteria assessment of four ensemble techniques as Majority Voting, AdaBoost, Stacking, and Bayesian Decision on the CIC IoT 2023 dataset.

II. Related Work

Feature selection (FS) is a key component of intrusion detection systems (IDS) since it reduces dimensionality, gets rid of unnecessary features, and makes classification more

accurate. Recent studies have utilized diverse feature selection techniques on novel IoT datasets to improve attack detection [7]. Nonetheless, no singular strategy is universally superior; their efficacy is contingent upon data attributes, assessment standards, and application goals [8].

To overcome these constraints, ensemble methodologies have been extensively utilized to enhance the stability and performance of Intrusion Detection Systems (IDS) and File Systems (FS) using Internet of Things (IoT) data. Ensembles mitigate bias inherent in individual methods by capitalizing on the diversity among specialists and harnessing collective strength [9]. Some common strategies are Majority Voting, Bagging, Boosting (like AdaBoost), Stacking, and Bayes Optimal. Fusion mechanisms can work at the decision or feature level that enhance detection precision, flexibility to emerging threats, and overall scalability.

In [10], several aggregation strategies, such as union, intersection, rank aggregation, and majority voting, were evaluated, indicating that the optimal selection is contingent upon accuracy objectives and the dimensionality of the feature set. Other comparative studies show trade-offs: Stacking and AdaBoost often work best because they use the best weights [10]; Boosting lowers bias while Bagging lowers variance [11]; and Stacking works best for feature-based classification tasks [12]. However, the majority of research prioritizes accuracy and generalization over explainability. Recent efforts to include XAI into ensembles [13] mostly improve the transparency of individual models, yet fail to adequately tackle the "black box" characteristic of aggregation in FS for IDS. This gap drives our research to a thorough assessment of ensemble techniques integrated with XAI to examine expert and feature-level contributions on the CIC IoT 2023 dataset.

III. Methodology

Figure 1 illustrates the comparison system for result aggregation methods in multi-expert feature selection models for IoT attack detection on edge computing environments.

A. Ensemble Methods

1) Majority Voting Method: The majority voting decision algorithm is considered one of the most commonly used rules for combining decisions from multiple experts [16]. According to this principle, the accepted hypothesis is the one that receives more than half of the total expert votes. Specifically, if the number of experts is N, then the selected hypothesis must achieve at least $\lfloor N/2 \rfloor + 1$ votes. The popularity of this rule stems not only from its intuitive nature, ease of understanding, and compatibility with social decision-making models, but is also reinforced by solid mathematical foundations. Many studies have proven that if each expert in the system has an equal probability of making correct decisions greater than 0.5, then as the number of experts approaches infinity, the probability of the system achieving correct decisions will approach 1. This result can be extended to cases where experts have different correctness probabilities, provided all probabilities are greater than 0.5. In the proof,

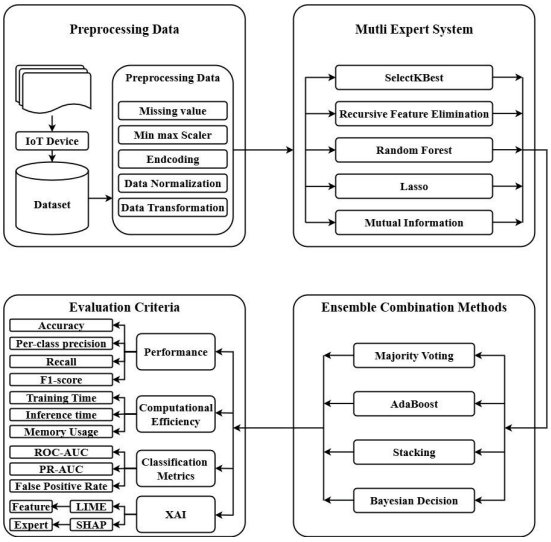


Fig. 1: Comparative system

different probabilities are replaced by the minimum value to ensure generality of results. Additionally, the majority voting algorithm has distinct characteristics when the number of experts is even or odd. These characteristics lead to differences in handling tie situations or absolute advantage, which have been analyzed in detail in [17].

2) AdaBoost Algorithm: The AdaBoost Algorithm (Adaptive Boosting) is used as an aggregation mechanism to optimize decision-making processes in multi-expert systems. Instead of considering each expert as equally reliable as in majority voting methods, AdaBoost learns to assign different weights to experts based on classification performance. Using AdaBoost allows the system to focus more on high-accuracy experts while reducing the influence of experts that frequently make incorrect predictions. Specifically, at each iteration, the algorithm calculates the classification error ε_t of expert t and assigns weight α_t according to the formula [18],

$$\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \varepsilon_t}{\varepsilon_t} \right). \tag{1}$$

These weights are used to update the influence of each expert on the aggregated decision. The iterative process continues until reaching a predetermined number of iterations or when the error converges. The final decision of the multi-expert system is made according to the classification function.

$$H(x) = \text{sign} \left(\sum_{t=1}^{T} \alpha_t h_t(x) \right), \tag{2}$$

where $h_t(x)$ is the prediction of expert t with x being the input feature vector.

Applying AdaBoost effectively exploits differences in reliability between experts while enhancing overall system accuracy compared to traditional aggregation methods like voting.

3) Stacking Ensemble Method: The Stacking Ensemble Method uses a second-level machine learning model (meta-learner) to learn optimal coordination mechanisms between component predictions. In this architecture, each base model f_i generates a probability prediction vector $p_i = f_i(x)$ with x being the input feature vector and $p_i \in \mathbb{R}^C$, where C is the number of classification classes. Vectors from N experts are concatenated to create a meta feature vector [19]:

$$z = \text{concat}(f_1(x), f_2(x), \ldots, f_N(x)) \in \mathbb{R}^{N \cdot C} \tag{3}$$

This vector z is then fed into a meta classifier $g(\cdot)$ to produce the final **classification label**:

$$y = g(z) = g(f_1(x), f_2(x), \ldots, f_N(x)) \tag{4}$$

This method allows maximum exploitation of the complementary potential between experts, especially when each model has different tendencies or biases. The meta model can learn complex nonlinear relationships between experts, thereby significantly improving performance compared to traditional aggregation methods like averaging or simple voting.

4) Bayesian Decision Method: One of the popular decision fusion methods in multi-expert systems is the Bayes algorithm, built on Bayes' formula [20]. Assuming a finite number of classes (hypotheses) M and a set of statistically independent decision-making experts, the basic Bayes formula has the form:

$$P(H_m|E) = \frac{P(E|H_m) \cdot P(H_m)}{P(E)} \tag{5}$$

Where:

- $P(H_m|E)$: conditional probability for the system to accept hypothesis H_m given decision E
- $P(H_m)$: prior probability of class H_m occurrence
- $P(E|H_m)$: probability that experts make decision E when the signal belongs to class H_m

If assuming expert independence, we have:

$$P(H_m|E) = \prod_{j=1}^{N} P(e_j|H_m) \tag{6}$$

The decision function is then constructed as:

$$g_m(E) = P(H_m) \cdot \prod_{j=1}^{N} P(e_j|H_m) \tag{7}$$

The input signal is assigned to class H_m such that $g_m(E)$ achieves maximum value.

In practice, probability $P(e_j|H_m)$ is estimated through the confusion matrices of each expert. However, the Bayes method has two important limitations: (i) the statistical independence assumption between experts is often not satisfied in practice; (ii) results may be biased when the occurrence probability of a class is very small, leading to system bias toward classes with higher prior probabilities, even when all experts choose different classes.

B. Explaining Multi-Expert Ensemble Decisions with LIME and SHAP

To enhance transparency and interpretability of decision fusion systems from multiple experts, this research integrates an explanation module based on two XAI techniques: LIME and SHAP. These two methods provide complementary explanation capabilities at two levels: LIME operates at the input feature level, allowing analysis of each feature's influence, while SHAP evaluates each expert's contribution to the final fusion result.

Feature Influence Explanation using LIME: LIME approximates the model decision boundary using locally weighted regression [21]:

$$g(x) = \beta_0 + \sum_{i=1}^{m} \beta_i x_i + \varepsilon \tag{8}$$

where β are weights learned from perturbed samples. LIME provides explanations in the form of weights assigned to input features, reflecting the influence level of each feature on the final aggregated result. By identifying features with the greatest impact, this method allows deeper analysis of the input data's role in decision-making processes, while enhancing system transparency and verifiability.

Expert-Level Explanation with SHAP: SHAP assigns an importance score S_i for each expert X_i, calculated according to the formula [22]:

$$S_i = \sum_{S \subseteq F \setminus \{X_i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} [f(S \cup \{X_i\}) - f(S)] \tag{9}$$

where F is the full expert set and $f(S)$ is the model output with subset S.

SHAP (SHapley Additive exPlanations) is a game theory-based method that allows quantitative measurement of each expert's contribution to the final aggregated result in multi-expert systems. By decomposing the final prediction value into components corresponding to each expert, this method provides detailed information about individual expert impacts in specific cases. SHAP analysis not only ensures transparency of expert coordination mechanisms but also helps detect bias issues or excessive dependence on certain experts. These analyses contribute to enhancing model reliability and interpretability while providing a scientific basis for optimizing structure and expert combination strategies in complex systems.

IV. Experiments and Results

A. An Overview of the CIC IoT 2023 Dataset

To validate the proposed approach, we employed the CICIoT2023 dataset provided by the Canadian Institute for Cybersecurity (CIC) [15]. This benchmark dataset is widely adopted in IoT security research due to its large scale and realistic attack scenarios. The data was generated from a testbed that integrates 105 heterogeneous IoT devices, thereby emulating the complexity of modern IoT deployments. Among them, 67 devices were directly involved in attack scenarios,

while the other 38 operated with Zigbee and Z-Wave protocols across five control centers. CICIoT2023 includes 46 network flow features and 33 attack categories grouped into seven families, ensuring broad coverage of network architectures, communication standards, and behavioral patterns of critical IoT devices.

B. Data Processing

Before training, data underwent preprocessing to enhance quality and reliability. Missing values were replaced with mean or median to ensure integrity, while duplicate or non-informative records were removed to reduce noise. Subsequently, data was normalized to synchronize ranges between features, facilitating model training. As a result, the final dataset comprised 466,866 samples with 46 attributes, maintaining the original distribution of 8 attack types, ensuring representativeness of IoT environments. Finally, stratified sampling techniques were applied to divide data into training and testing sets, preserving class proportions and helping models better recognize rare attack types.

C. Experimental Parameters

Parameter settings for the 5 feature selection methods are presented in Table I:

TABLE I: Parameter Settings for Feature Selection Methods

Expert Name	Scikit-learn Class	Main Parameters
SelectKBest	SelectKBest	score_func=f_classif, k=10
RFE	RFE	estimator=LR(max_iter=10(n_features_to_select=10, step=1
Random Forest	RandomForestClassifier	n_estimators=100, criterion='gini', max_depth=None, min_samples_split=2
Lasso	Lasso + SelectFromModel	alpha=chosenby cross-validation, max_iter=1000, threshold='mean'
Mutual Information	mutual_info_classif	discrete_features='auto', n_neighbors=3, copy=True

Parameter settings for the 4 result aggregation methods are presented in Table II:

TABLE II: Parameter Settings for Ensemble Methods

Method	Parameters
Majority Voting	voting="soft", weights=[1, 2, 1]
AdaBoost	n_estimators=200, learning_rate=0.5
Stacking	final_estimator=LogisticRegression()
Bayesian Decision	var_smoothing=1e-9

To evaluate the result aggregation methods, we used an MLP model with the following parameters: The MLP was trained for 50 epochs using the Adam optimizer (learning rate 0.001, batch size 128), with sparse_categorical_crossentropy loss and callbacks such as ModelCheckpoint to retain the best validation accuracy.

D. Explaining Ensemble Results using XAI

1) Feature Influence Explanation using LIME: From the analysis results, it can be seen that Majority Voting and Stacking Ensemble methods identify relatively similar influential feature sets, with prominent features such as Weight, Number, rst_count, Variance, HTTPS, Max. These features reflect the influence of traffic factors, protocols, and network ratios on attack detection capability, especially when feature values are relatively uniform.

For AdaBoost, the order and type of features change significantly: Header_Length, ack_flag_number, flow_duration become more important. This reflects the tendency to recognize special patterns based on header information and session duration, thereby diversifying detection capabilities.

Meanwhile, Bayesian Decision Method, besides using familiar quantitative features, also combines distribution-describing features (Covariance, Radius, Std), helping the system become more sensitive to abnormal behaviors in terms of scale and statistical variation that other models find difficult to recognize.

Notably, the feature group Weight, Number, rst_count, Variance appears consistently across all methods, playing the role of core features providing decisive information to distinguish abnormal states. This stability ensures consistency and increases model transferability in changing environments.

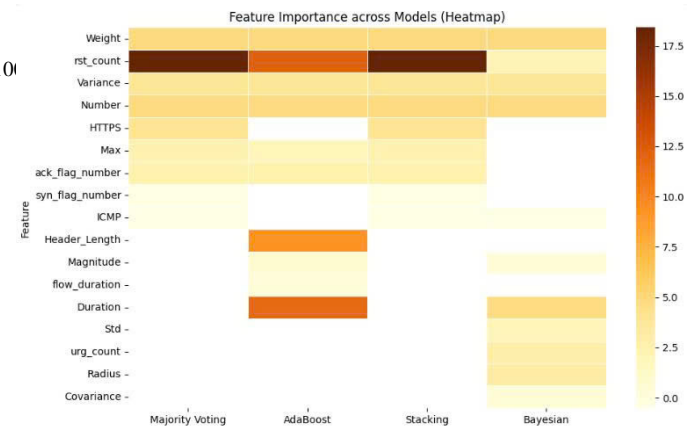


Fig. 2: LIME value plots for four different ensemble/decision methods

2) Expert Influence Explanation using SHAP: SHAP analysis allows a clear explanation of each feature selection expert's influence in the four result aggregation mechanisms. With Majority Voting, the system prioritizes stable experts like Lasso and RandomForest, while Mutual Information contributes only marginally. This helps maintain balance and reliability, suitable for contexts requiring stability.

AdaBoost distributes contributions fairly evenly among multiple experts, including SelectKBest, Mutual Information, RandomForest, and RFE. This is a fundamental characteristic of Boosting: combining both strong and weak experts to reduce bias, but increasing computational cost and latency.

Stacking shows heavy dependence on Mutual Information, while Lasso and SelectKBest play supporting roles. This bias helps increase Recall but simultaneously poses risks of increased false alarms if the main expert is corrupted.

Bayesian Decision emphasizes SelectKBest and RFE with SHAP distributions spread across both positive and negative directions. This shows that Bayes is suitable for linearly structured data, both increasing interpretability and supporting anomaly pattern detection.

E. Performance Criteria Comparison

TABLE III: Performance Metrics Comparison

Method	Accuracy	Precision	Recall	F1-score
Majority Voting	0.996444	0.893249	0.964026	0.927289
AdaBoost	0.995149	0.881067	0.917577	0.898952
Stacking	0.995224	0.835249	0.992714	0.907199
Bayesian Decision	0.704382	0.073665	0.999545	0.137218

Analysis of Table III shows that each aggregation mechanism exhibits different performance configurations. Majority Voting achieves the highest F1-score (0.927289) by balancing Precision (0.893249) and Recall (0.964026), maintaining classification effectiveness while controlling false alarms. AdaBoost achieves competitive performance with metrics close to Majority Voting, suitable for scenarios requiring balance between correct detection and error limitation.

Stacking optimizes detection capability with a Recall of 0.992714, but low Precision (0.835249) reduces the F1-score to 0.907199; this method is suitable when maximum coverage of attack cases is needed, but requires adjustment to limit false alarms. Conversely, Bayesian achieves near-absolute Recall (0.999545) but very low Precision (0.073665), pulling F1-score down to 0.137218, showing difficulty in controlling false alarms in noisy edge environments.

F. Computational Efficiency Comparison

TABLE IV: Computational Efficiency Comparison

Method	Training Time (s)	Inference Time (s)	Memory Usage (MB)
Majority Voting	117.212786	0.367319	0.832127
AdaBoost	157.774939	3.255955	0.754509
Stacking	385.066334	0.383666	0.830904
Bayesian Decision	0.158375	0.040807	0.758259

Experimental results indicate significant differences in computational efficiency. Bayesian Decision has training time of only 0.158 seconds and an inference time of 0.041 seconds, faster by two to three orders of magnitude compared to other methods. Majority Voting and Stacking have similar inference times (0.37–0.38 seconds), but Stacking requires much longer training time (385 seconds vs. 117 seconds). AdaBoost has inference time up to 3.26 seconds, nearly 80 times higher than Bayesian. Memory usage ranges 0.75–0.83 MB, relatively uniform.

TABLE V: Classification Performance Indicators

Method	ROC-AUC	PR-AUC	False Positive Rate
Majority Voting	0.999483	0.981397	0.002775
AdaBoost	0.999193	0.963783	0.002983
Stacking	0.999472	0.999472	0.004716
Bayesian Decision	0.983261	0.983261	0.302727

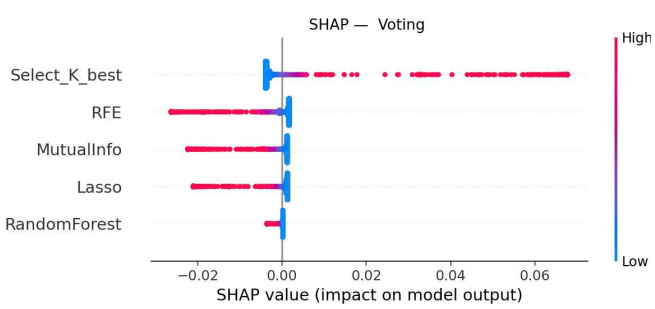
G. Classification Performance Comparison

Majority Voting, AdaBoost, and Stacking all achieve near-saturated ROC-AUC ($\geq$ 0.9991), so ROC-AUC is no longer the main distinguishing criterion. Differences are clearer in PR-AUC and false positive rate (FPR). Stacking achieves the highest PR-AUC (0.999472) but also higher FPR (0.004716). Majority Voting has the lowest FPR (0.002775) with PR-AUC 0.981397, suitable for systems prioritizing false alarm control. AdaBoost achieves PR-AUC 0.999193 and FPR 0.002983, balancing accurate detection and error control. Bayesian Decision only achieves ROC-AUC and PR-AUC 0.983261 with high FPR 0.302727, not meeting edge environment requirements.

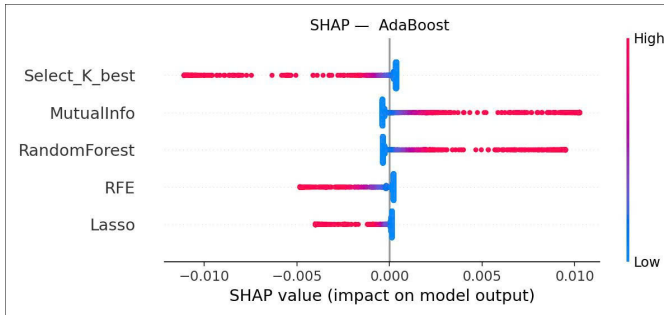
V. Conclusion

Multi-dimensional comparison demonstrates that no aggregation mechanism is preferable; each method uses different expert selection and feature set procedures for different deployment circumstances. Majority Voting balances and stabilizes, AdaBoost utilizes diversity, Stacking enhances detection but increases false alarms, and Bayesian is computationally efficient but inaccurate. This research showed that explainable AI (XAI) analysis indicates that multi-expert system success depends on combination mechanisms and how each technique utilizes and weights component experts. Consistent properties like Weight, Number, rst_count, and Variance establish stability and model transferability across operational conditions. This conclusion suggests creating XAI-based dynamic expert selection techniques to enable systems to adapt to data characteristics and operational conditions rather than employing a fixed architecture. Experimental results have evident practical relevance. Bayesian Decision Method is ideal for resource-constrained edge devices due to its fast processing speed and optimal memory usage; Stacking is best for gateways or high-configuration devices; and Majority Voting and AdaBoost balance performance and reliability in standard scenarios. Selecting aggregation algorithms affects energy optimization, heat reduction, and edge device hardware lifespan, which are crucial for continuously functioning systems.

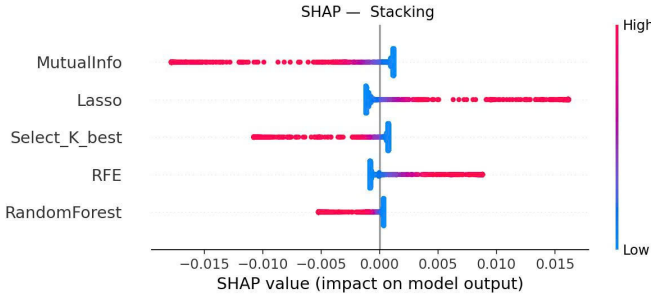
Our future work will focus on feature selection systems and feature compression methods that reduce computational costs while maintaining model quality for resource-constrained IoT edge devices. XAI-guided expert pruning mechanisms automatically remove redundant experts and features, improving balance and self-adjustment according to each device layer's resource status and processing requirements in edge computing systems.



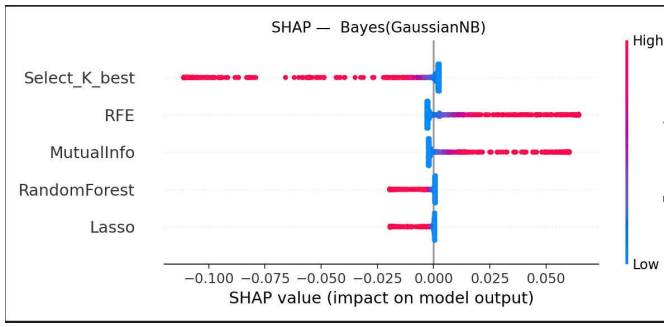
(a) Majority Voting Method



(b) AdaBoost Algorithm



(c) Stacking Ensemble Method



(d) Bayesian Decision Method

Fig. 3: SHAP value plots for four different ensemble/decision methods.

References

[1] Samonte, M.J.C., et al.: Securing IoT Ecosystems: Integration Challenges and Architectural Solutions. In: 2024 IEEE 7th International Conference on Computer and Communication Engineering Technology (CCET). IEEE (2024)

[2] El Hajla, S., Ennaji, E.M., Maleh, Y., Mounir, S.: Security Challenges and Solutions in IoT. Advances in Computational Intelligence and Robotics Book Series, pp. 25–50 (2024)

[3] Khan, N.A., Awang, A., Karim, S.A.A.: Security in Internet of Things: A review. IEEE Access 10, 104649–104670 (2022)

[4] Vallabhaneni, R.: Effects of Data Breaches on Internet of Things (IoT) Devices within the Proliferation of Daily-Life Integrated Devices. Engineering And Technology Journal 9(7), 4439–4442 (2024)

[5] Li, J., et al.: Optimizing IoT intrusion detection system: feature selection versus feature extraction in machine learning. Journal of Big Data 11(1), 36 (2024)

[6] Almohaimeed, M., Albalwy, F.: Enhancing IoT Network Security Using Feature Selection for Intrusion Detection Systems. Applied Sciences 14(24) (2024)

[7] Gheni, H., Oleiwi, W., Albarmani, Z., Alabdali, M.: Optimizing Feature Selection for Intrusion Detection: A Hybrid Approach Using Cuckoo Search and Particle Swarm Optimization. International Journal of Safety and Security Engineering 14, 1907–1912 (2024)

[8] Wolpert, D.H., Macready, W.G.: No free lunch theorems for optimization. IEEE Transactions on Evolutionary Computation 1(1), 67–82 (1997)

[9] Raza, M.S., Sheikh, M.N.A., Hwang, I.-S.: Ensemble Learning-Based DDOS Attack Recognition in IoT Networks. Computer Networks and Communications 3(2), 73–83 (2025)

[10] Bolón-Canedo, V., Alonso-Betanzos, A.: Ensembles for textwidth: A review and future trends. Information Fusion 52, 1–12 (2018)

[11] Şevgin, H.: A comparative study of ensemble methods in the field of education: Bagging and Boosting algorithms. International Journal of Assessment Tools in Education 10, 544–562 (2023)

[12] Chandran, S., Ahuja, C., Elango, S.: A Comparative Study of Feature Selection and Machine Learning Methods for Sentiment Classification on Movie Data Set. In: Advances in Intelligent Systems and Computing, vol. 343, pp. 367–379. Springer (2015)

[13] Kondamudi, M., Sahoo, S.R.: Integrating Explainable AI with Enhanced Ensemble Models for Accurate and Transparent Fake News Detection in OSN's. Procedia Computer Science 258, 1081–1090 (2025)

[14] HryniewskaGuzik, W., Sawicki, B., Biecek, P.: NormEnsembleXAI: Unveiling the strengths and weaknesses of XAI ensemble techniques. arXiv preprint arXiv:2401.17200 (2024)

[15] Canadian Institute for Cybersecurity (CIC): CICIoT2023 Dataset. University of New Brunswick. https://www.unb.ca/cic/datasets/iotdataset-2023.html (2023)

[16] Kuncheva, L.I.: Combining Pattern Classifiers: Methods and Algorithms. John Wiley & Sons, Inc., Hoboken, New Jersey (2004)

[17] Avedyan, E.D., Le Thi Trang Linh: Intrusion detection in mobile networks using hybrid deep learning. Informatization and Communication 2020(6), 7–14 (2020)

[18] Freund, Y., Schapire, R.E.: A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. Journal of Computer and System Sciences 55(1), 119–139 (1997)

[19] Sill, J., Takács, G., Mackey, L., Lin, D.: Feature-Weighted Linear Stacking. arXiv preprint arXiv:0911.0460 (2009)

[20] Gnedenko, B.V.: A Course in the Theory of Probability, 6th edn. Nauka, Main Editorial Board of Physical and Mathematical Literature, Moscow, Russia (1988)

[21] Ribeiro, M.T., Singh, S., Guestrin, C.: "Why should I trust you?" Explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1135–1144 (2016)

[22] Lundberg, S.M., Lee, S.-I.: A unified approach to interpreting model predictions. In: Advances in Neural Information Processing Systems, vol. 30 (2017)