Jurnal Elektronika dan Telekomunikasi (JET), Vol. 26, No. 1, August 2026, pp. 45-53
Accredited by KEMDIKTISAINTEK, Decree No: **295/C/C3/KPT/2026**
doi: 10.55981/jet.823

Explainable IoT Intrusion Detection Using Random Forest, SMOTE, and SHAP

Julfikar Mawansyah[a,*], Anik Nur Handayani[b], Aji Prasetya Wibawa[b], Triyanna Widiyaningtyas[b], Mokh. Sholihul Hadi[b], Fidyah Ajeng Wulandari[a]

[a] *Department of Information System and Technology*
Faculty of Science and Technology
Mbojo Bima University
Piere Tandean Street, Mande II, Mpunda District, Bima City
Bima City, Indonesia
[b] *Department of Electrical Engineering and Informatics*
Faculty of Engineering
State University of Malang
5 Semarang Street
Malang, Indonesia

Abstract

Class imbalance in Internet of Things (IoT) Intrusion Detection System (IDS) datasets is a major challenge that degrades detection performance on minority attacks and complicates model interpretability. This study investigates an IoT IDS based on Random Forest (RF) combined with the Synthetic Minority Over-sampling Technique (SMOTE) and explainable Artificial Intelligence (AI) using SHapley Additive exPlanations (SHAP) on the public IoT Network Intrusion Dataset (IoTID20). The pipeline comprises data preprocessing, a stratified train-test split (7,000 training, 3,000 test samples), training an RF baseline, applying SMOTE to balance the Denial of Service (DoS), Scan, Normal, and Man-in-the-Middle Address Resolution Protocol (MITM ARP) spoofing classes, and training an RF-SMOTE model. Both models are compared using per-class and macro-averaged accuracy, precision, recall, and F1-score, followed by SHAP-based global feature importance analysis. Results show the RF baseline achieves high performance (accuracy of about 0.99, macro F1 of approximately 0.984), while RF-SMOTE maintains the same accuracy with a macro F1 of around 0.983. SMOTE improves training-set class distribution but yields only minor differences in aggregate performance: RF-SMOTE slightly enhances sensitivity for some minority classes, while F1-score for MITM ARP spoofing decreases marginally versus baseline. SHAP analysis indicates that flow-related traffic features, including connection duration, packet counts, byte volume, and packet direction ratios, dominate both models, with RF-SMOTE showing greater relative contribution from features representing rare attack patterns, making minority-class explanations more prominent. Overall, the RF-SMOTE-SHAP combination preserves strong detection performance while improving class balance and interpretability, rather than outperforming baseline RF in aggregate predictive accuracy.

Keywords: Explainable AI, Intrusion Detection System, Internet of Things, Random Forest, SMOTE, SHAP.

I. INTRODUCTION

The Internet of Things (IoT) has grown rapidly and been widely adopted in various domains such as smart homes, healthcare, industry, smart transportation, and smart cities, resulting in millions of heterogeneous devices connected to each other through complex and dynamic networks [1] – [6]. This increase in connectivity brings significant benefits, but at the same time expands the attack surface and opens up opportunities for exploiting security vulnerabilities in IoT-based devices, communication protocols, and services [3], [7] – [9]. Various studies show that attacks such as Distributed Denial of Service (DDoS), botnets, spoofing, Routing Protocol for Low-Power and Lossy Networks (RPL) protocol manipulation, and malware targeting IoT and Internet of Medical Things (IoMT) environments can cause service disruptions, data theft, and compromised critical systems [2], [10] – [12]. In this context, Intrusion Detection Systems (IDS) specifically designed for IoT traffic characteristics, device heterogeneity, and computational resource limitations are key components for monitoring network traffic, detecting anomalies, and providing early warnings of threats [4], [13] – [15].

To address the complexity of attack patterns and the dynamics of IoT traffic, various modern IDS approaches utilize machine learning (ML) and deep learning (DL) to learn the differences between normal and malicious traffic in various scenarios, such as BoT–IoT, UNSW-NB15, smart home environments, Industrial IoT (IIoT), and smart vehicle networks [1], [16], [17]. Models such as Random Forest, Gradient Boosting, ensemble learning, hybrid deep neural networks, and feature engineering-based architectures have been proven to achieve high accuracy and F1 scores in detecting various types of attacks on IoT, IoMT, and Internet of Vehicles (IoV) [5], [18], [19]. Several works emphasize that decision tree-based ensembles and effective feature selection can provide competitive performance with lower complexity than some heavy DL models, making

* Corresponding Author.
Email: jmawansyah@gmail.com
Received: December 08, 2025 ; Revised: June 02, 2026
Accepted: June 17, 2026 ; Published: August 31, 2026

them suitable for implementation on edge devices or limited IoT infrastructure [20] – [22]. On the other hand, the federated learning approach has also begun to be used to train IDSs in a distributed manner without transferring raw data, thereby improving privacy and scalability in large-scale IoT environments [17], [23], [24].

However, the IoT IDS domain still faces fundamental challenges in the form of class imbalance in public and operational datasets, where normal classes and majority attacks (e.g., DDoS or generic botnet traffic) are far more dominant than minority attacks that are actually critical, such as low-rate attacks, protocol manipulation, or specific exploits [5], [21], [22]. In real-world scenarios and some datasets, certain types of attacks only appear in small proportions, causing models to be biased toward the majority class and ignore rare attacks even though aggregate metrics such as accuracy appear high [1], [10], [16]. To overcome this issue, various studies combine feature selection, feature engineering, and IDS architecture design that is more sensitive to attack pattern variations, for example, through special ensemble designs, parameter optimization, and feature engineering techniques for specific protocols, such as Message Queuing Telemetry Transport (MQTT) or Routing Protocol for Low-Power and Lossy Networks (RPL) [12], [25], [26]. A number of works also propose ensemble learning and multiclass classification-based frameworks to improve detection reliability for many types of IoT attacks, but class imbalance is often handled implicitly through feature selection, threshold adjustment, or emphasis on certain metrics, rather than through systematic analysis of oversampling strategies and their impact on the model [5], [21], [22].

In addition to the need for accuracy and the ability to detect rare attacks, the aspect of explainability is also increasingly seen as important for building trust in IDS used in critical IoT environments. Several studies have begun to integrate Explainable Artificial Intelligence (XAI) principles into the design of IoT IDS, for example, by utilizing a two-stage framework that combines DL models and interpretability techniques to explain the contribution of features to detection decisions [13], [14], [17].

This approach allows security operators to understand which features most influence the classification of traffic as an attack or normal, thereby facilitating forensic analysis and network security policy adjustments [17], [24]. On the other hand, there is also a growing approach based on data synthesis using generative models to improve the quality of IDS training data, including in the context of IoT-sensor networks, which utilize Conditional Tabular Generative Adversarial Network (CTGAN) to generate synthetic data while maintaining the utility and privacy of the original data [17], [27]. However, most of these XAI and data synthesis studies focus on complex DL architectures or federated learning frameworks, while the use of XAI, particularly techniques such as SHAP, to analyze relatively lightweight tree-based models such as Random Forest in the context of handling class imbalance in IoT IDS is still rarely reported explicitly [23], [24], [27].

Based on this gap, this study proposes an RF–SMOTE–SHAP approach on a public IoT-IDS dataset with two main objectives. First, evaluate the extent to which the application of SMOTE can improve Random Forest detection performance, especially for minority attack classes, compared to the baseline RF without explicit balancing, by considering metrics that are sensitive to class imbalance such as recall and F1-score per class. Second, to provide a comprehensive XAI analysis using SHAP to explain feature contributions both globally and locally, so that changes in feature importance patterns due to SMOTE can be observed systematically. By integrating Random Forest, SMOTE, and SHAP into a relatively lightweight and measurable pipeline, this study aims to bridge the gap between complex IoT IDS solutions and the practical need for accurate, efficient, and transparently explainable models in resource-constrained IoT environments [1], [5], [6], [10], [12], [13], [15] – [17], [20] – [22], [24] – [27].

II. METHODS

This section describes in detail the methodology employed to develop and evaluate the proposed explainable IoT intrusion detection framework based on Random Forest, SMOTE, and SHAP using the IoTID20 dataset. The main stages include the selection and description of the dataset, data preprocessing (covering cleaning, numerical and categorical feature encoding, and normalization), stratified train–test splitting, training a baseline Random Forest model on the imbalanced training set, and applying SMOTE to balance the class distribution in the training data, followed by training the RF–SMOTE model. The performance of both models is then compared using accuracy, precision, recall, and F1-score for each class as well as macro averages. Finally, SHAP is used to perform an explainable AI analysis by quantifying the relative contribution of each feature to the model's predictions, thereby providing a more transparent understanding of the behavior of the proposed IoT IDS.

A. Random Forest

In this study, the main model used is Random Forest (RF), which is a decision tree-based ensemble algorithm that builds a number of trees on randomly selected subsets of data and features, then combines the predictions of each tree through a majority voting scheme for classification. RF is widely used in IoT IDS because it can handle high-dimensional data, non-linear patterns and is relatively efficient in resource-limited environments [5], [28], [29].

In general, for a decision tree in an RF forest, split selection at each node is based on minimizing impurity, such as the $Gini$ index. Suppose a node contains samples from K classes with class probability p_k. The Gini impurity is then defined as in (1):

$$Gini = 1 - \sum_{k=1}^{K} p_k^2 \quad (1)$$

RF builds many trees independently by performing bootstrap sampling on the training data and feature subsampling at each split. The final prediction for an instance $\boldsymbol{x}$ is obtained by majority voting across all trees, as expressed in (2):

$$\hat{y} = mode\big(h_1(x),\, h_2(x),\, \ldots,\, h_M(x)\big) \qquad (2)$$

where h_m denotes the m-th decision tree in the ensemble, M is the total number of trees, and the predicted class label $\hat{y}$ is obtained by majority voting over all trees, as expressed in (2).

The advantages of RF in IoT IDS that are often reported include the following:

a. Stable against noise,
b. Can provide feature importance to support interpretability analysis,
c. Good performance on IDS datasets such as UNSW-NB15, CICIDS, and various other IoT datasets [5], [19], [28], [29].

In this study, the main RF hyperparameters that were adjusted included the number of trees (n_estimators), the maximum tree depth (max_depth), the minimum number of samples for splitting and leaves, and the number of features considered in each split (max_features). The values selected were taken from a combination of literature reviews and preliminary experiments that maintained a compromise between accuracy, overfitting, and computational cost. Feature importance for global interpretation was further assessed using mean-decrease-in-impurity (MDI), which served as the conventional baseline for the SHAP analysis described in Section II.C.

B. SMOTE

IoT intrusion detection datasets often exhibit substantial class imbalance, where majority classes dominate the training data while minority attack classes are underrepresented. This condition may bias the classifier toward majority classes and reduce its ability to recognize rare but important attacks [10], [28], [30]. To address this issue, this study applies the Synthetic Minority Oversampling Technique (SMOTE) to the training set.

Let $x_i \in \mathbb{R}^d$ denote a minority-class sample and let x_{nn} be one of its k-nearest minority neighbors. SMOTE generates a synthetic sample x_{new} by linear interpolation, as expressed in (3)

$$x_{new} = x_i + \lambda(x_{nn} - x_i) \qquad (3)$$

where $\boldsymbol{\lambda}$ is a random value drawn from a uniform distribution between 0 and 1. This formulation places the synthetic sample on the line segment between two minority-class instances, thereby expanding the minority-class region in feature space without simply duplicating existing observations.

In this study, SMOTE is applied only to the training data, while the test set is kept in its original imbalanced form to preserve realistic evaluation conditions. This strategy is intended to improve the representation of minority classes during model learning while avoiding information leakage into the evaluation stage. Previous IDS studies have shown that combining SMOTE with supervised learning models such as Random Forest and XGBoost can improve minority-class recall and F1-score in imbalanced intrusion detection scenarios [25], [29].

C. SHAP

To improve explainability, this study employs SHapley Additive exPlanations (SHAP) as the main XAI technique for interpreting the predictions of the random forest models. SHAP is grounded in Shapley value theory from cooperative game theory and explains a model prediction as the sum of feature-wise contributions [5], [12], [24], [28], [31].

For an input instance $x = [x_1, x_2, \ldots, x_n]$, the SHAP explanation model can be written as in (4):

$$f(x) = \varphi_0 + \sum_{i=1}^{N} \varphi_i \qquad (4)$$

where $f(x)$ is the model output for instance x, φ_0 is the base value or average model output over the dataset, and φ_i is the SHAP value representing the contribution of feature x_i to the deviation of the prediction from the base value. Conceptually, the Shapley value for feature i is defined as in (5)

$$\varphi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!\,(|N|-|S|-1)!}{|N|!} \big(f(S \cup \{i\}) - f(S)\big) \qquad (5)$$

where N is the set of all features, S is a subset of features that does not contain feature i, $f(S)$ denotes the model prediction based on subset S, and $f(S \cup \{i\}) - f(S)$ measures the marginal contribution of feature i.

Because the exact computation of Shapley values is computationally expensive, practical SHAP implementations use optimized algorithms. In this study, SHAP is applied to a tree-based model, so TreeSHAP is used to efficiently compute feature attributions for both the baseline RF and RF-SMOTE models. The resulting SHAP values are then analyzed to compare global feature-importance patterns and to assess whether class balancing changes the relative contribution of features in attack detection [12], [28].

D. Dataset

This study uses the IoTID20 dataset, a publicly available intrusion detection dataset for Internet of Things (IoT) environments obtained from Kaggle (https://www.kaggle.com/datasets/rohulaminlabid/iotid20-dataset/data). The dataset is organized in a flow-based tabular format, where each row represents a network traffic flow instance and each column corresponds to a traffic feature or class label. Such a representation is commonly adopted in IoT intrusion detection studies because it preserves statistical traffic characteristics that can be processed effectively by machine learning models [26]. The use of flow-based traffic records is also consistent with recent IDS studies in IoT environments that emphasize feature-driven detection of malicious and benign communication patterns [28], [29].

In this work, an experimental subset of 10,000 records from IoTID20 was used. The subset was obtained through stratified random sampling (without replacement) from the original IoTID20 distribution using a fixed random seed (random_state = 42) so that the relative proportions of the five classes (Mirai, DoS, Scan, Normal, and MITM ARP spoofing) were preserved and the experiment remained fully reproducible.

After preprocessing, the data were divided using a stratified train–test split into 7.000 training samples and 3.000 testing samples so that the class proportions remained nearly identical in both subsets [5], [10]. Based on the resulting distribution, the dataset used in this study consists of five classes, namely Mirai, DoS, Scan, Normal, and MITM ARP spoofing.

The Mirai class dominates both the training and testing sets, while Normal and MITM ARP spoofing appear as minority classes. This pattern indicates a clear class imbalance problem, which motivates the application of SMOTE in the training phase [25], [28], [29]. The class counts reported in the manuscript show 4,622/1,981 samples for Mirai, 669/287 for DoS, 849/364 for Scan, 436/186 for Normal, and 424/182 for MITM ARP spoofing in the training/testing sets, respectively.

To help readers understand the dataset structure more clearly, Table 1 presents an illustrative example of the IoTID20 tabular format used in this study. The table is intended to show the general organization of the data, in which traffic identifiers, source and destination information, protocol attributes, flow statistics, and class labels are recorded in a structured form. This type of representation is appropriate for preprocessing, supervised classification, and explainability analysis using SHAP [1], [10].

The example in Table 1 is provided only as a visual illustration of the dataset structure and does not represent the full number of features contained in IoTID20. Nevertheless, it helps clarify that the dataset used in this research is not image-based or text-based, but rather a structured network-flow dataset suitable for machine-learning-based IoT intrusion detection [26].

Because the IoTID20 dataset is multi-class, attack labels can be grouped into K classes (including normal). For each class $c \in \{1,2,\ldots,K\}$, the number of samples is denoted by N_c. One important characteristic analyzed in this study is the degree of class imbalance, which can be quantified using the imbalance ratio (IR) defined in (6)

$$IR_c = \frac{N_{max}}{N_c} \quad (6)$$

with:

- $N_{max} = \max_j N_j$ denotes the number of samples in the largest (majority) class, and
- N_c denotes the number of samples in class c.

If IR_c is high for some attack classes, it means that those classes are minority classes that may be difficult for the model to detect without special handling. The phenomenon of class imbalance with high IR has also been widely reported in IDS datasets such as NSL-KDD, UNSW-NB15, DS2OS, ToN_IoT, and Edge-IIoTset in recent IDS and XAI studies [28] – [30].

III. Result and Discussion

Figure 1 illustrates the RF–SMOTE–SHAP-based IoT IDS experimental workflow used in this study. First, the IoTID20 dataset from Kaggle was loaded and underwent pre-processing in the form of data cleaning, categorical feature encoding, and feature scale normalization, as commonly done in previous IoT IDS studies [1], [5].

The processed data is then divided into training and test sets in a stratified manner to maintain the proportion of normal and attack classes [5], [10]. From this point, the flow branches into two paths: (a) the baseline path, which is Random Forest training on the still unbalanced training set and evaluation on the test set followed by SHAP analysis, and (b) the RF–SMOTE path, which is the application of SMOTE on the training set to balance the classes, Random Forest training on the oversampled data, then evaluation and SHAP analysis on the model [28], [29].

In the final stage, the performance of both models and the SHAP explanation patterns (global and local) are compared to assess the effect of SMOTE on attack detection, particularly in minority classes, while also improving understanding of model behavior in IoT environments [5], [28], [29].

The proposed IDS IoT conceptual framework consists of five main layers. The bottom layer presents IoT data and environment, namely, network traffic from IoT devices and services modeled in the form of flow-based records as commonly used in various IDS IoT studies [5], [10], [28], [29]. Above it, the class imbalance handling layer uses SMOTE to balance the class distribution in the training set so that minority attacks are better represented without changing the original distribution in the test set [28], [29].

The next layer is the Random Forest model (baseline and RF–SMOTE), which is tasked with learning normal patterns and attacks from the processed data, in line with many works that emphasize the effectiveness of tree-based ensembles for IoT IDS [5], [19], [28]. Above the model, the explainability layer (SHAP) provides global and local explanations for RF decisions, as practiced in the XAI framework for modern IDS [5], [24], [28], [31]. The top layer, evaluation and interpretation, combines

TABLE 1.
ILLUSTRATIVE STRUCTURE OF THE IOTID20 DATASET

Flow ID	Source IP	Destination IP	Protocol	Flow Duration	Total Forward Packets	Total Backward Packets	Flow Bytes/s	Class Label
flow_001	192.168.0.13	192.168.0.24	TCP	24531	18	12	1450.7	Mirai
flow_002	192.168.0.15	192.168.0.16	TCP	1834	6	5	320.4	Normal
flow_003	192.168.0.20	192.168.0.13	UDP	9042	22	3	2781.9	DoS
flow_004	192.168.0.11	192.168.0.18	TCP	4120	9	4	954.2	Scan
flow_005	192.168.0.17	192.168.0.13	ARP	2310	7	6	611.8	MITM ARP spoofing

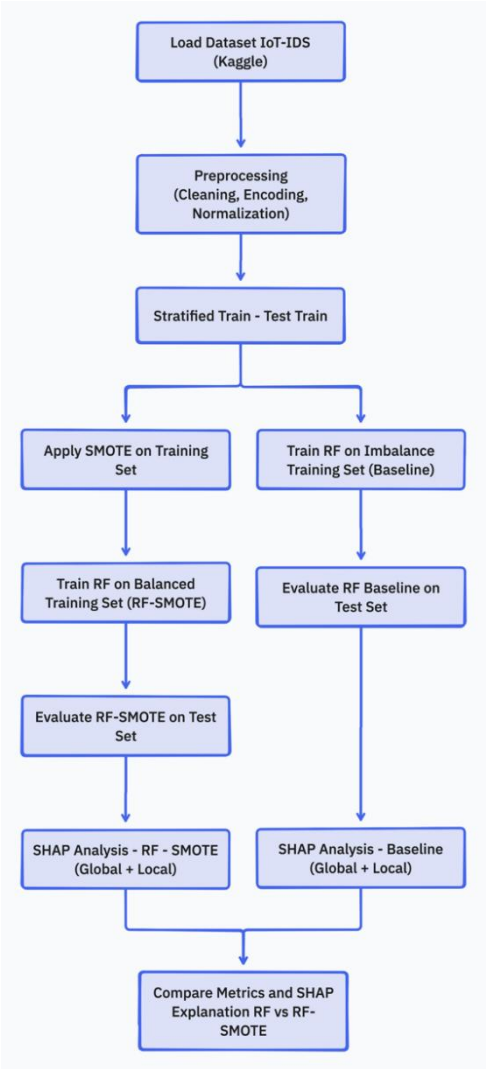

Figure 1. Experimental workflow of the proposed Random Forest with SMOTE and SHAP for intrusion detection on the IoTID20 dataset.

performance metrics (accuracy, precision, recall, F1, and macro-F1) and SHAP analysis results to assess and explain the effect of SMOTE on attack detection capabilities in IoT environments [5], [28], [29].

After preprocessing and stratified train–test split on 10,000 IoTID20 samples, 7,000 samples were obtained as training data and 3,000 samples as test data with class distributions as shown in Table 2. The Mirai class dominated both the training and test data, with 4,622 (66.03%) and 1,981 samples (66.03%), respectively. The DoS and Scan classes followed with proportions of approximately 9.56% and 12.13% in both data sets (669 and 287 samples for DoS; 849 and 364 samples for Scan in training and testing). Meanwhile, the Normal and MITM ARP spoofing classes had much lower frequencies, approximately 6.23% and 6.06% in the training data (436 and 424 samples) and test data (186 and 182 samples), respectively. This pattern indicates a fairly strong class imbalance, where most samples come from Mirai attacks, while the Normal and MITM ARP spoofing classes are relatively rare. Since the division was performed in a stratified manner, the percentage distribution between the training set and the test set is

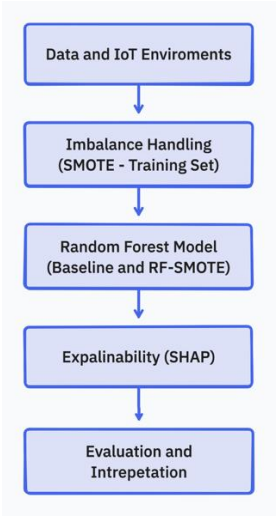

Figure 2. Conceptual framework of the proposed RF–SMOTE–SHAP-based IoT IDS.

almost identical, so the model evaluation still reflects the same imbalance characteristics as in the training data.

Table 3 and Figure 3 show the class distribution in the training set before and after applying SMOTE. Before SMOTE, the difference in sample size between the Mirai class and the class with the lowest frequency (e.g., MITM ARP spoofing) was quite large, indicating a significant imbalance ratio. After SMOTE was applied to the training set, the number of samples in each class became more balanced, with the number of samples in the minority class increasing to approach that of the majority class. This process was performed only on the training set, so that the distribution in the test set still reflected the actual imbalanced conditions in the IoT network. Conceptually, these results show that SMOTE successfully generated synthetic minority samples

TABLE 3.
TRAINING-SET CLASS DISTRIBUTION BEFORE AND AFTER SMOTE

Class Index	Class Name	Before SMOTE	After SMOTE	Before SMOTE (%)	After SMOTE (%)
0	Mirai	4622	4622	66.028571	27.806521
1	DoS	669	3000	9.557143	18.048370
2	Scan	849	3000	12.128571	18.048370
3	Normal	436	3000	6.228571	18.048370
4	MITM ARP spoofing	424	3000	6.057143	18.048370

TABLE 2.
TRAIN/TEST CLASS DISTRIBUTION OF THE IoTID20 EXPERIMENTAL SUBSET (10,000 SAMPLES)

Class Name	Train	Test	Train (%)	Test (%)
Mirai	4622	1981	66.028571	66.033333
DoS	669	287	9.557143	9.566667
Scan	849	364	12.128571	12.133333
Normal	436	186	6.228571	6.200000
MITM ARP spoofing	424	182	6.057143	6.066667

Figure 3. Bar chart of the training-set class distribution before and after applying SMOTE.

without changing the distribution of the test data so that he model had more examples of minority attacks at the training stage but was still evaluated in an imbalanced real-world scenario.

Table 4 presents a comparison of the performance of the Random Forest (RF) baseline and RF–SMOTE for each attack class and macro average metric. In general, both models show very high performance, with an accuracy of around 0.99 and precision and recall values above 0.94 for all classes. In the Mirai and DoS classes, both RF and RF–SMOTE achieved F1 values close to 1.00, indicating that both models were able to detect these attack patterns very well. For the Scan and Normal classes, the F1 values were also in the range of 0.97–0.99, with very small differences between the baseline model and the balanced model.

More noticeable changes appear in classes with relatively lower frequencies, namely, MITM ARP spoofing. In this class, the baseline RF obtained an F1 of around 0.963, while RF–SMOTE decreased slightly to around 0.957. However, when viewed from the macro F1, the baseline RF and RF–SMOTE obtained values of around 0.984 and 0.983, respectively. This shows that in the IoTID20 dataset subset used, SMOTE does not produce a significant improvement in aggregate performance, but it also does not drastically reduce performance. In other words, the RF model is already quite robust under the existing class imbalance, while RF–SMOTE is still able to maintain very high performance while providing a fairer basis for XAI analysis on minority classes.

Figure 4 shows a comparison of the macro F1 values between the baseline RF model and RF–SMOTE. Overall, both models show nearly identical performance, with macro F1 values of approximately 0.984 for RF and 0.983 for RF–SMOTE. This very small difference indicates that, on the IoTID20 subset used (7,000 training data and 3,000 test data), the application of SMOTE does not result in a significant improvement in cross-class average performance but also does not reduce the model's detection capability. Thus, the use of SMOTE in this study serves more to balance class representation in the training phase and support fairer XAI analysis in minority classes, rather than to pursue large increases in accuracy or macro F1 scores.

XAI analysis using SHAP was performed to understand which features contributed most to the RF model's decisions in the baseline and RF–SMOTE scenarios. Figures 5 and 6 show the global SHAP summary plots for each model, while Table 5 summarizes the ten features with the highest average absolute SHAP values. In general, both models show very similar feature importance patterns: features related to traffic flow characteristics, such as connection duration, number of packets, volume of bytes sent or received, and the ratio of packets between forward and backward directions, dominate the top positions. This is in line with findings in previous IoT IDS research, that attack behavior is largely reflected in the intensity and pattern of packet communication rather than solely in protocol attributes.

TABLE 4.
COMPARISON OF RF AND RF–SMOTE PERFORMANCE (PRECISION, RECALL, F1 PER CLASS, AND MACRO AVERAGE)

Class Index	Class Name	Baseline Precision	Baseline Recall	Baseline F1 Score	RF-SMOTE Precision	RF-SMOTE Recall	RF-SMOTE F1 Recall
0	Mirai	0.983127	1.000000	0.991491	0.983607	0.999495	0.991487
1	DoS	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
2	Scan	1.000000	0.947802	0.973202	0.997118	0.950549	0.973277
3	Normal	1.000000	0.989247	0.994595	1.000000	0.989247	0.994595
4	MITM ARP spoofing	1.000000	0.928571	0.962963	0.994083	0.923077	0.957265

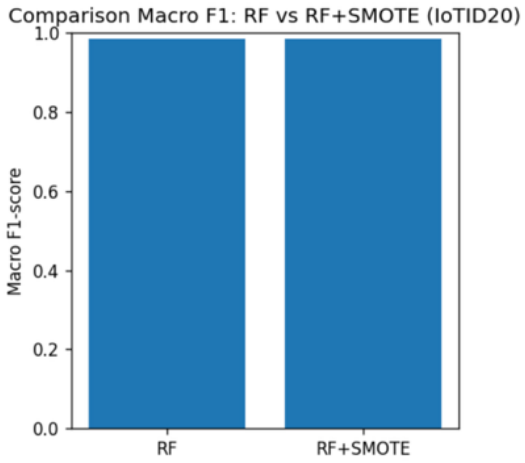

Figure 4. Bar chart comparison of macro F1: RF vs RF–SMOTE on the IoTID20 experimental subset.

A comparison of SHAP values between the RF baseline and RF–SMOTE in Table 5 shows that SMOTE does not drastically change the order of important features but slightly adjusts the relative contribution of some features. For some features related to infrequent attack activities (e.g., features describing typical MITM/ARP spoofing packet patterns), the SHAP values in the RF–SMOTE model tend to increase slightly. This indicates that the class balancing process makes the model more sensitive to traffic patterns that were previously underrepresented in the training data. Thus, the RF–SMOTE–SHAP combination not only maintains high detection performance but also provides a more balanced explanation of the factors that trigger attack detection, especially in minority classes.

IV. Conclusion

This study investigated an IoT intrusion detection approach based on Random Forest (RF), SMOTE, and SHAP using a subset of the IoTID20 dataset. The data were preprocessed through cleaning, feature encoding, normalization, and stratified train–test splitting into

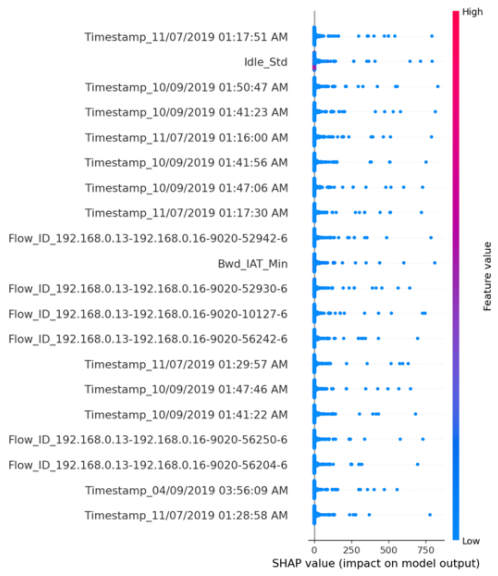

Figure 5. Global SHAP summary plot for the RF–SMOTE model.

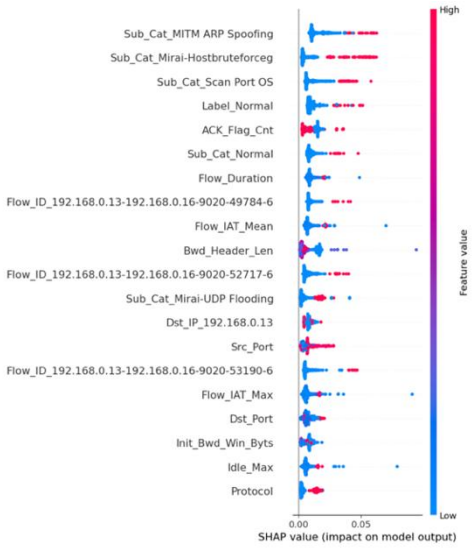

Figure 6. Global SHAP summary plot for the baseline RF model.

7,000 training samples and 3,000 testing samples. The class distribution analysis confirmed a substantial imbalance, with the Mirai class dominating the dataset, while Normal and MITM ARP spoofing were relatively underrepresented.

The experimental results show that the baseline RF model already achieved very strong classification performance, with an accuracy of approximately 0.99 and a macro F1-score of about 0.984. After applying SMOTE to the training data, the resulting RF–SMOTE model maintained a similarly high accuracy of approximately 0.99, with a macro F1-score of about 0.983. These results indicate that, for the IoTID20 subset used in this study, SMOTE did not lead to a substantial improvement in aggregate predictive performance. Instead, its main contribution lies in providing a more

TABLE 5.
TOP 10 FEATURES RANKED BY MEAN ABSOLUTE SHAP VALUE (BASELINE RF VS RF–SMOTE)

	Feature	Mean Absolute SHAP (Baseline)	Mean Absolute SHAP (RF-SMOTE)
0	Timestamp_11/07/2019 01:17:51 AM	24.635793	2.672009e-07
1	Idle_Std	23.729407	2.934833e-03
2	Timestamp_10/09/2019 01:50:47 AM	22.911690	3.313587e-06
3	Timestamp_10/09/2019 01:41:23 AM	22.288347	3.704555e-06
4	Timestamp_11/07/2019 01:16:00 AM	21.640055	7.006279e-07
5	Timestamp_10/09/2019 01:41:56 AM	20.869356	6.759197e-06
6	Timestamp_10/09/2019 01:47:06 AM	20.806216	7.701235e-04
7	Timestamp_11/07/2019 01:17:30 AM	20.761529	1.060994e-03
8	Flow_ID_192.168.0.13-192.168.0.16-9020-52942-6	20.127654	1.208703e-03
9	Bwd_IAT_Min	20.083773	4.937359e-03

balanced class representation during training, which is useful for analyzing minority classes more fairly.

The SHAP analysis further showed that both the baseline RF and RF–SMOTE models relied on broadly similar flow-related features, such as connection duration, packet counts, byte volume, and directional traffic characteristics. Although SMOTE did not markedly change the overall ranking of the most influential features, it slightly altered the relative contribution of some features associated with less frequent attack patterns. This suggests that the balanced training process may improve the interpretability of minority-class behavior, even when aggregate classification performance remains nearly unchanged.

Overall, the findings suggest that the RF–SMOTE–SHAP combination should be viewed not as a framework that significantly outperforms the baseline RF in accuracy, but rather as an approach that maintains competitive detection performance while improving class balance in training and supporting more transparent feature-level interpretation. Therefore, its main value in this study lies in enhancing the interpretability and fairness of analysis for minority attack classes, rather than in producing large gains in overall predictive accuracy.

DECLARATIONS

Conflict of Interest

The authors have declared that there are no competing interests.

CRediT Authorship Contribution

Julfikar Mawansyah: Conceptualization, Methodology, Software, Formal analysis, Data curation, Writing – Original draft preparation; Anik Nur Handayani: Supervision, Validation, Writing – Reviewing and Editing; Aji Prasetya Wibawa: Supervision, Validation, Writing – Reviewing and Editing; Triyanna Widiyaningtyas: Methodology, Validation, Writing – Reviewing and Editing; Mokh. Sholihul Hadi: Investigation, Resources, Writing – Reviewing and Editing; Fidyah Ajeng Wulandari: Investigation, Visualization, Writing – Reviewing and Editing.

Funding

The authors received no financial support for this article's research, authorship, and/or publication.

Acknowledgement

We would like to thank the Department of Electrical Engineering and Informatics, State University of Malang, for their support and facilitation of this research.

REFERENCES

[1] J. Ashraf, G. M. Raza, B.-S. Kim, A. Wahid, and H.-Y. Kim, “Making a Real-Time IoT Network Intrusion-Detection System (INIDS) Using a Realistic BoT–IoT Dataset with Multiple Machine-Learning Classifiers,” *Appl. Sci.*, vol. 15(4), Feb. 2025, doi: 10.3390/app15042043.

[2] E. G. Ribera, B. M. Alvarez, C. Samuel, P. P Ioulianou, and V. G. Vassilakis, “Intrusion Detection System for RPL-Based IoT Networks,” *Electronics*, vol. 11(23) 4041, 2022, doi: 10.3390/electronics11234041.

[3] A. Javed, et al., “Implementation of Lightweight Machine Learning-Based Intrusion Detection System on IoT Devices of Smart Homes,” *Future Internet*, vol. 16(6) 200, Jun. 2024, doi: 10.3390/fi16060200.

[4] V. Kelli et al., ”IDS for Industrial Applications: A Federated Learning Approach with Active Personalization,” *Sensors*, vol. 21(20) 6743, Oct 2021, doi: 10.3390/s21206743.

[5] Y. Alotaibi and M. Ilyas, “Ensemble-Learning Framework for Intrusion Detection to Enhance Internet of Things’ Devices Security,” *Sensors*, vol.23(12) 5568, Jun 2023, doi: 10.3390/s23125568.

[6] A. Javed et al., “Embedding Tree-Based Intrusion Detection System in Smart Thermostats for Enhanced IoT Security,” *Sensors*, vol. 24 no. 22, 7320, Nov. 2024, 10.3390/s24227320.

[7] G. Zachos, et al., “An Anomaly-Based Intrusion Detection System for Internet of Medical Things Networks,” *Electronics*, vol. 10(21), 2562, Oct. 2021, doi: 10.3390/electronics10212562.

[8] F. B. Saghezchi, G. Mantas, M. A. Violas, A. J. de Oliveira Duarte, and D. Guimarães, ”Machine Learning for DDoS Attack Detection in Industry 4.0 CPPSs,” *Electronics*, vol. 11(4) 602, Feb. 2022, doi :10.3390/electronics11040602.

[9] M. Alsharif and D. B. Rawat, “ Study of Machine Learning for Cloud Assisted IoT Security as a Service,” *Sensors*, vol. 21(4) 1034, Feb. 2021, doi: 10.3390/s21041034.

[10] A. Churcher et al., “An Experimental Analysis of Attack Classification Using Machine Learning in IoT Networks,” *Sensors*, vol. 21 (2) 446, Jan 2021, doi: 10.3390/s21020446.

[11] C. D. Morales-Molina et al., “A Dense Neural Network Approach for Detecting Clone ID Attacks on the RPL Protocol of the IoT,” *Sensors*, vol. 21(9) 3173, May 2021, doi: 10.3390/s21093173.

[12] S. Ullah et al., “HDL-IDS: A Hybrid Deep Learning Architecture for Intrusion Detection in the Internet of Vehicles,” *Sensors*, vol 22(4) 1340, Feb. 2022, doi: 10.3390/s22041340.

[13] S. K. Erskine, “Real-Time Large-Scale Intrusion Detection and Prevention Assessment Based on Deep Learning,” *Appl. Syst. Innov.*, vol. 8(2), April. 2025, doi: 10.3390/asi8020052.

[14] B. Sousa, N. Magaia, and S. Silva, “Intelligent Intrusion Detection System for 5G-Enabled Internet of Vehicles,” *Electronics*, vol. 12(8) 1757, 2023, doi: 10.3390/electronics12081757.

[15] I. Aliyu, S. Van Engelenburg, M. B. Mu'azu, J. Kim, and C. G. Lim, "Statistical detection of adversarial examples in blockchain-based federated forest in-vehicle network intrusion detection systems," *IEEE Access*, vol. 10, pp. 109366–109384, 2022, doi: 10.1109/ACCESS.2022.3212412.

[16] S. More, M. Idrissi, H. Mahmoud, and A. T. Asyhari, “Enhanced Intrusion Detection Systems Performance with UNSW-NB15 Data Analysis,” *Algorithms*, vol. 17(2), Feb. 2024, doi: 10.3390/a17020064.

[17] G. Abdelmoumin, D. B. Rawat, and M. A. Rahman, “Studying Imbalanced Learning for Anomaly-Based Intelligent IDS for Mission-Critical Internet of Things,” *J. Cybersecur. Priv.*, vol. 3(4), pp. 706–743, Oct. 2023, doi: 10.3390/jcp3040032.

[18] K. Harahsheh, R. Al-Naimat, and C.-H. Chen, ”Using Feature Selection Enhancement to Evaluate Attack Detection in the Internet of Things Environment,” *Electronics*, vol. 13(9) 1678, Apr. 2024, doi: 10.3390/electronics13091678.

[19] G. C. Amaizu, A. M. V. V. Sai, M. Siddula, and D.-S Kim, “Cost-Efficient Hybrid Filter-Based Parameter Selection Scheme for Intrusion Detection System in IoT,” *Electronics*, vol. 14(4) 726, 2025, doi: 10.3390/electronics14040726.

[20] K. Albulayhi, Q. Abu Al-Haija, S. A. Alsuhibany, and A. Jillepalli, “IoT Intrusion Detection using Machine Learning with a Novel High Performing Feature Selection Method,” *Appl. Sci.*, vol. 12(10), 5015, May. 2022, doi: 10.3390/app12105015.

[21] N. Abosata, S. Al-Rubaye, and G. Inalhan, “Customised Intrusion Detection for an Industrial IoT Heterogeneous Network Based on Machine Learning Algorithms Called FTL-CID,” *Sensors*. vol. 23(1) 321, Dec. 2022, doi: 10.3390/s23010321.

[22] A. Alrefaei and M. Ilyas, “Using Machine Learning Multiclass Classification Technique to Detect IoT Attacks in Real Time,” *Sensors*, vol 24(14) 4516, Jul. 2024, doi: 10.3390/s24144516.

[23] R. Lazzarini, H. Tianfield, and V. Charissis, “Federated Learning for IoT Intrusion Detection,” *AI*, vol. 4, pp. 509–530, 2023, doi: 10.3390/ai4030028.

[24] M. M. Mahmoud, Y. O. Youssef, and A. A. Abdel-Hamid, “ XI2S-IDS: An Explainable Intelligent 2-Stage Intrusion

Detection System," *Future Internet*, vol. 17(1) 25, Jan. 2025, doi: 10.3390/fi17010025.

[25] S. Ullah et al., "A New Intrusion Detection System for the Internet of Things via Deep Convolutional Neural Network and Feature Engineering," *Sensors*, vol. 22(10) 3607, May 2022, doi: 10.3390/s22103607.

[26] A. Al Hanif, and M. Ilyas, " Effective Feature Engineering Framework for Securing MQTT Protocol in IoT Environments," *Sensors,* vol. 24(6) 1782, Mar. 2024, doi: 10.3390/s24061782.

[27] S. Alabdulwahab, Y.-T. Kim, and Y. Son, "Privacy-Preserving Synthetic Data Generation Method for IoT-Sensor Network IDS Using CTGAN," *Sensors*, vol. 24(22) 7389, Nov. 2024, doi: 10.3390/s24227389.

[28] I. Ioannou *et al.*, "GEMLIDS-MIoT: A Green Effective Machine Learning Intrusion Detection System Based on Federated Learning for Medical IoT Network Security Hardening," *Comput. Commun.*, vol. 218, pp. 209–239, Mar. 2024, doi: 10.1016/j.comcom.2024.02.023.

[29] M. Roopak, G. Y. Tian, and J. Chambers, "Multi-Objective-Based Feature Selection for DDoS Attack Detection in IoT Networks," *IET Netw.,* vol. 9, no. 4, pp. 214–222, 2020, doi: 10.1049/iet-net.2018.5206.

[30] A. Zahoor, W. Abbasi, M. Z. Babar, and A. Aljohani, "Robust IoT Security Using Isolation Forest and One-Class SVM Algorithms," *Sci. Rep.*, 15, Oct. 2025, doi: 10.1038/s41598-025-20445-4.

[31] Z. Deng, A. Torim, S. Ben Yahia, and H. Bahsi, "Generative AI in Intrusion Detection Systems for Internet of Things: A Systematic Literature Review," *IEEE Open J. Commun. Soc.*, pp. 4689 – 4717, May 2025, doi: 10.1109/OJCOMS.2025.3573194.