SEMAR JOURNAL

Explainable Machine Learning for IoT Botnet-Related DDoS Detection Through Outbound Network Traffic Analysis

Zaydan Khalifa Al-Muhannadi[1]✉, Nayla Mariam Al-Kuwari[2], Aseel Haya Al-Nuaimi[3], Ramez Tahnoun bin Ahmed Al-Kaabi[4], Alya Reem bint Zayed Al-Shamsi[5]

[1-3]College of Arts and Sciences, Qatar University, Qatar
[4,5]Computer Information Science, Higher Colleges of Technology, United Arab Emirates

doi: 10.23917/semar.vxxx.xxxxx
Received: April 12th, 2025 | Revised: May 9th, 2025 | Accepted: May 18th, 2025
Available Online: May 22nd, 2025 | Published Regularly: May, 2025

Abstract

The increasing deployment of Internet of Things (IoT) devices has introduced significant cybersecurity challenges, particularly the threat of botnet-related Distributed Denial of Service (DDoS) attacks. Conventional Intrusion Detection Systems (IDSs) predominantly focus on inbound traffic analysis, potentially limiting their ability to identify suspicious communication originating from compromised internal devices. This study proposes a machine learning-based approach for detecting IoT botnet-related DDoS attacks through outbound network traffic analysis, integrating classification performance evaluation and Explainable Artificial Intelligence (XAI). The proposed approach employs Random Forest and Extreme Gradient Boosting (XGBoost), alongside an ensemble configuration combining both algorithms. Experiments were conducted using 50,000 network flow records derived from four scenarios of the IoT-23 dataset, comprising benign and DDoS attack traffic. Model performance was evaluated using Accuracy, Precision, Recall, F1-Score, and Stratified 10-Fold Cross-Validation. The results demonstrate that Random Forest achieved the best overall performance on the balanced dataset, with an accuracy of 97.52%, precision of 99.76%, recall of 95.27%, and F1-score of 97.46%. The reported cross-validation accuracy reached 97.57%. Furthermore, SHapley Additive exPlanations (SHAP) identified `orig_ip_bytes`, `proto_state_interaction`, and `orig_pkts_per_sec` as the most influential features in attack classification, highlighting the importance of data transmission volume, protocol interactions, and packet transmission intensity. A web-based prediction system was also developed to demonstrate the practical integration of traffic classification and feature-level explanations. These findings demonstrate the potential of combining machine learning and explainable AI to develop accurate and interpretable IoT network intrusion detection systems.

Keywords: Internet of Things; DDoS Detection; Outbound Traffic Analysis; Random Forest; XGBoost; Explainable Artificial Intelligence; SHAP.

This is an open access article under the CC-BY license.

✉**Corresponding Author:**
Zaydan Khalifa Al-Muhannadi, College of Arts and Sciences, Qatar University, Qatar
Email: 25081920@mail.qu.edu.qa

Introduction

The rapid proliferation of Internet of Things (IoT) technologies has fundamentally transformed the digital landscape, enabling interconnected devices to support a wide range of applications, from smart homes and industrial automation to healthcare systems and critical infrastructure. In Indonesia, the increasing adoption of IoT technologies has also expanded the potential attack surface of interconnected

networks, creating substantial cybersecurity challenges. According to the 2024 Cybersecurity Environment Report published by the National Cyber and Crypto Agency (Badan Siber dan Sandi Negara, BSSN), network traffic anomalies in Indonesia are predominantly associated with malware exploiting vulnerabilities in IoT devices, with hundreds of millions of cyberattacks detected annually (Badan Siber dan Sandi Negara [BSSN], 2025). This situation highlights the growing importance of strengthening security mechanisms within IoT environments, particularly as the number of connected devices continues to increase. Unlike conventional computing systems, many IoT devices are designed with limited computational resources, memory capacity, and energy consumption requirements. Consequently, security mechanisms are frequently constrained by hardware limitations and inadequate security configurations. Common weaknesses include the use of default credentials, insufficient authentication mechanisms, and the absence of reliable firmware update procedures (Hassija et al., 2019; Neshenko et al., 2019). These limitations make IoT devices particularly susceptible to malware infections and unauthorized access. Furthermore, the restricted computational capabilities of such devices often hinder the implementation of computationally intensive security mechanisms, including advanced encryption and conventional antivirus solutions (Sicari et al., 2015; Zarpelão et al., 2017). Therefore, developing effective and resource-conscious intrusion detection mechanisms has become an important research priority for securing increasingly interconnected IoT infrastructures (Al-Garadi et al., 2020).

Among the emerging cybersecurity threats, IoT-based botnets represent a particularly significant challenge because of their ability to compromise large numbers of vulnerable devices and coordinate malicious activities across distributed networks. A botnet is a collection of compromised devices infected with malicious software and remotely controlled by an attacker, commonly referred to as a botmaster, to execute unauthorized operations (Bertino & Islam, 2017; Silva et al., 2013). In IoT environments, botnets exploit weaknesses in device authentication, software configurations, and firmware security to establish persistent control over vulnerable devices. Malware families such as Mirai and Gafgyt demonstrate how compromised IoT devices can be recruited into large-scale botnet infrastructures (Antonakakis et al., 2017; Kolias et al., 2017). Once infected, these devices can communicate with command-and-control (C2) servers and receive instructions to perform coordinated malicious activities (Gu et al., 2008). One of the most damaging consequences of IoT botnet infections is their ability to initiate Distributed Denial of Service (DDoS) attacks by generating massive volumes of malicious network traffic. Such attacks can overwhelm network resources, disrupt service availability, and compromise the operational continuity of critical digital services (Mirkovic & Reiher, 2004; Zargar et al., 2013). Beyond their immediate operational impact, IoT botnets also present persistent security risks because compromised devices may remain undetected within internal networks while continuously communicating with external malicious infrastructure (Meidan et al., 2018). Accordingly, identifying infected devices before they participate in large-scale attacks is

essential for improving the resilience of IoT network security.

Intrusion Detection Systems (IDSs) have traditionally been developed to identify malicious activities by monitoring network traffic and detecting deviations from established behavioral patterns (Khraisat et al., 2019; Liao et al., 2013). However, a substantial proportion of existing IDS research emphasizes inbound traffic analysis, particularly the examination of incoming connections and malicious traffic directed toward servers or targeted network resources (Chaabouni et al., 2019; Mirkovic et al., 2002). Although this approach is useful for identifying attacks that have already reached their intended destinations, it provides limited opportunities for identifying compromised IoT devices at an earlier stage. In particular, inbound-oriented detection mechanisms may fail to recognize suspicious communication originating from infected devices within local networks. This limitation creates a significant detection gap because compromised IoT devices can establish outbound connections to external C2 servers before initiating coordinated attacks against other systems (Bilge et al., 2012). Consequently, monitoring outbound network traffic offers a complementary and potentially more proactive approach to identifying malicious activities. By examining communication patterns generated by internal devices, outbound traffic analysis can reveal abnormal connection frequencies, unusual communication destinations, irregular packet characteristics, and other behavioral indicators associated with malware infections. More importantly, identifying suspicious outbound communications may enable the detection of C2 activities before compromised devices are instructed to execute subsequent malicious operations. Recent studies have demonstrated the potential of outbound traffic analysis and machine learning techniques to improve the identification of anomalous network behavior and strengthen early detection capabilities in IoT environments (Doshi et al., 2018; Kumar & Lim, 2019; Nguyen et al., 2019). Nevertheless, further investigation is required to develop detection mechanisms that effectively combine outbound traffic monitoring, reliable attack classification, and interpretable decision-making.

Machine learning and deep learning approaches have increasingly become important components of modern network intrusion detection because of their ability to identify complex patterns in high-dimensional network traffic data (Buczak & Guven, 2016; Ferrag et al., 2020). Traditional machine learning algorithms, particularly Random Forest and Extreme Gradient Boosting (XGBoost), have attracted considerable attention due to their ability to model nonlinear relationships, handle heterogeneous network traffic features, and provide robust classification performance (Dhaliwal et al., 2018; Resende & Drummond, 2018). Random Forest employs an ensemble of decision trees constructed through bootstrap sampling and randomized feature selection, thereby reducing overfitting and improving prediction stability (Breiman, 2001). In contrast, XGBoost adopts a gradient boosting strategy that sequentially optimizes decision trees to minimize classification errors and capture complex relationships within the dataset (Chen & Guestrin, 2016; Friedman, 2001). These complementary learning mechanisms provide opportunities to improve the discrimination of normal and malicious network traffic,

particularly in scenarios involving diverse attack characteristics and imbalanced class distributions. Meanwhile, deep learning architectures, including Long Short-Term Memory (LSTM) networks and Convolutional Neural Networks (CNNs), have also been applied to network intrusion detection to capture temporal dependencies and complex feature representations in network traffic data (Shone et al., 2018; Vinayakumar et al., 2019; Yin et al., 2017). Despite their promising detection capabilities, these approaches may introduce computational complexity and interpretation challenges, particularly when their predictions are generated through complex internal model structures (Sommer & Paxson, 2010). Furthermore, high classification accuracy alone is insufficient to guarantee the practical reliability of an intrusion detection system, as security analysts also require an understanding of why a particular network connection or device is classified as malicious (Capuano et al., 2022). This issue is particularly relevant in operational cybersecurity environments, where detection decisions must be evaluated, investigated, and translated into appropriate mitigation strategies.

The interpretability of machine learning-based intrusion detection systems has therefore emerged as an important consideration in the development of trustworthy cybersecurity solutions. Many advanced classification models operate as black-box systems, producing predictions without providing sufficiently transparent explanations of the underlying decision-making processes. This limitation can reduce the ability of security analysts to identify the network characteristics associated with malicious behavior, assess the credibility of detection results, and determine appropriate responses to potential threats (Adadi & Berrada, 2018; Rudin, 2019). Explainable Artificial Intelligence (XAI) offers a systematic approach to addressing this challenge by providing mechanisms for interpreting model predictions and identifying the contribution of individual input features (Barredo Arrieta et al., 2020). Among the available XAI techniques, SHapley Additive exPlanations (SHAP) provides a theoretically grounded framework based on Shapley values from cooperative game theory (Lundberg & Lee, 2017; Shapley, 1953). SHAP quantifies the contribution of individual features to a model's prediction, enabling the identification of network traffic characteristics that have the greatest influence on attack classification. Through both global and local explanations, SHAP can provide insights into the overall importance of network features as well as the specific factors underlying individual detection decisions (Lundberg et al., 2020). In the context of IoT botnet detection, this capability is particularly valuable because it allows security analysts to investigate which outbound traffic characteristics contribute to the identification of suspicious communication patterns, thereby improving the transparency and practical applicability of machine learning-based detection systems (Wang et al., 2020; Zhang et al., 2022).

Despite advances in machine learning-based intrusion detection and explainable cybersecurity analytics, the integration of outbound traffic-oriented detection, ensemble classification, and feature-level interpretability remains an important research direction (Le et al., 2022; Oseni et al., 2023). In particular, the relationship between outbound communication characteristics, the classification capabilities of

different ensemble learning algorithms, and the explainability of their predictions requires further systematic investigation. Addressing this issue is important for developing detection systems that not only distinguish malicious network activities from legitimate traffic but also provide meaningful explanations that can support security analysis and incident response.

To address these challenges, this study proposes an outbound traffic-based intrusion detection approach for the early identification and classification of IoT botnet-related DDoS activities. The proposed approach employs Random Forest and XGBoost ensemble learning algorithms to classify network traffic based on its behavioral characteristics, with particular emphasis on identifying suspicious outbound communication patterns. Furthermore, SHAP-based Explainable Artificial Intelligence is integrated to interpret the classification results and determine the relative contributions of network traffic features to the model's predictions. By combining ensemble-based classification with explainability analysis, this study aims to evaluate detection performance while providing greater transparency regarding the factors influencing model decisions. The main contribution of this research lies in integrating outbound traffic analysis, comparative ensemble learning, and SHAP-based interpretability within an IoT-oriented intrusion detection framework. This integration is expected to support earlier identification of compromised IoT devices, improve the understanding of botnet-related communication patterns, and provide more interpretable evidence for cybersecurity monitoring and mitigation decisions.

Method

This study adopts the Cross-Industry Standard Process for Data Mining (CRISP-DM) as the methodological framework for developing and evaluating the proposed machine learning-based intrusion detection approach. CRISP-DM consists of six interconnected stages: Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation, and Deployment (Schröer et al., 2021; Wirth & Hipp, 2000). These stages provide a systematic process for identifying research problems, preparing datasets, developing classification models, and evaluating their performance. The overall research workflow is illustrated in Figure 1.

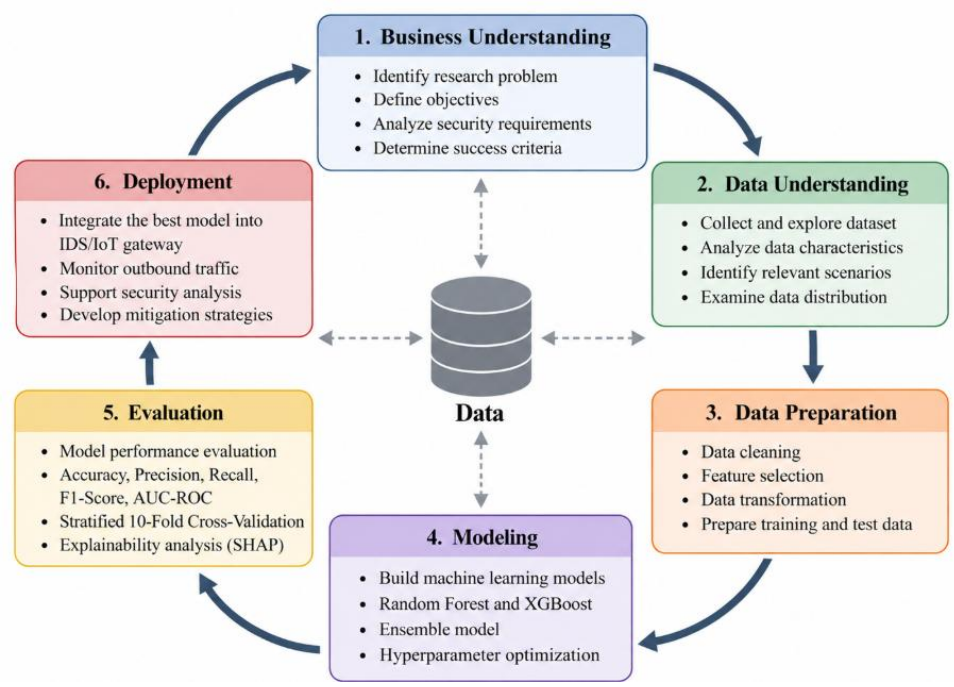

Figure 1. Cross-Industry Standard Process for Data Mining (CRISP-DM)

a. Business Understanding

The Business Understanding stage identifies the cybersecurity challenges associated with the increasing adoption of Internet of Things (IoT) devices. Their limited security mechanisms make them vulnerable to malware infections and exploitation as part of botnet networks, which can subsequently launch Distributed Denial of Service (DDoS) attacks. Conventional Intrusion Detection Systems (IDSs) predominantly focus on inbound traffic analysis, limiting their ability to identify suspicious communication originating from compromised internal devices. Therefore, this study aims to develop a machine learning-based detection approach that analyzes outbound network traffic to identify suspicious activities associated with IoT botnet-related DDoS attacks. Random Forest and XGBoost are employed as classification algorithms, while SHAP is integrated to improve model interpretability.

b. Data Understanding

This study utilizes the IoT-23 dataset provided by the Stratosphere Intrusion Prevention System (IPS) Laboratory (Garcia et al., 2020). The dataset contains network traffic recordings representing benign and malicious activities in IoT environments (Abdalgawad et al., 2022). Four scenarios, namely CTU-7, CTU-34, CTU-35, and CTU-17, were selected because they contain botnet-related activities relevant to DDoS attack patterns (García et al., 2014).

The selected scenarios were combined, and a sampling process was performed to obtain 50,000 network flow records. The resulting dataset consists of 23,461 benign records and 26,539 DDoS attack records. The classification labels are stored in the `binary_label` attribute, where 0 represents benign traffic and 1 represents attack traffic. Sampling was conducted to reduce computational complexity while maintaining an approximately balanced class distribution and sufficient data representation for model development (He & Garcia, 2009).

c. Data Preparation

The Data Preparation stage involves data cleaning, feature selection, and dataset

preparation for model training. Data cleaning was performed to identify missing values, duplicate records, and inconsistencies that could affect classification performance.

Ten network traffic features were selected: `conn_state`, `orig_ip_bytes`, `duration_pkts_ratio`, `orig_pkts_per_sec`, `orig_bytes_per_pkt`, `orig_overhead_ratio`, `duration`, `orig_pkts`, `proto_state_interaction`, and `orig_payload`. These features represent communication characteristics, including connection states, packet frequency, data volume, transmission duration, and payload information (Paxson, 1999).

Feature selection was based on the relevance of each attribute to network traffic behavior and previous research on flow-based IoT botnet detection. Preliminary correlation analysis was also conducted to examine relationships among the selected features and identify potentially redundant information (Guyon & Elisseeff, 2003; Koroniotis et al., 2019). Particular attention was given to features associated with outbound communication patterns that may indicate suspicious activities generated by compromised IoT devices.

d. Modeling

The Modeling stage employs two tree-based ensemble learning algorithms: Random Forest and Extreme Gradient Boosting (XGBoost). Random Forest constructs multiple decision trees using bootstrap sampling and randomized feature selection, whereas XGBoost uses a gradient boosting mechanism to sequentially optimize decision trees and minimize classification errors. Both algorithms were selected because of their effectiveness in handling structured network traffic data and modeling nonlinear relationships (Hasan et al., 2019; Resende & Drummond, 2018).

In addition to evaluating individual models, this study investigates an ensemble model combining Random Forest and XGBoost to examine whether integrating their predictions can improve classification performance. The approach is motivated by the complementary characteristics of bagging-based and boosting-based algorithms (Dietterich, 2000). Furthermore, these algorithms offer practical advantages in processing tabular data compared with computationally intensive deep learning architectures (Grinsztajn et al., 2022; Shwartz-Ziv & Armon, 2022).

Hyperparameter optimization was performed using RandomizedSearchCV, which randomly explores predefined parameter configurations to identify suitable model settings. This procedure aims to improve classification performance while reducing the computational requirements associated with exhaustive parameter searches (Bergstra & Bengio, 2012; Pedregosa et al., 2011).

e. Evaluation

The Evaluation stage assesses model performance using Accuracy, Precision, Recall, F1-Score, and Area Under the Receiver Operating Characteristic Curve (AUC-ROC). These metrics provide complementary measurements of classification effectiveness (Sokolova & Lapalme, 2009).

The confusion matrix consists of four components: True Positive (TP), representing correctly detected attacks; True Negative (TN), representing correctly classified benign traffic; False Positive (FP), representing benign traffic incorrectly classified as attacks; and False Negative (FN), representing attacks that remain undetected.

The evaluation metrics are calculated using the following equations:

$$\text{Accuracy} = \left(\frac{TP+TN}{TP+TN+FP+FN}\right) \quad (1)$$

$$\text{Precision} = \left(\frac{TP}{TP+FP}\right) \quad (2)$$

$$\text{Recall} = \left(\frac{TP}{TP+FN}\right) \quad (3)$$

$$\text{F1-Score} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (4)$$

Accuracy measures the proportion of correct predictions, while Precision evaluates the reliability of attack predictions. Recall measures the model's ability to identify actual attacks, and F1-Score provides a balanced assessment of Precision and Recall. Meanwhile, AUC-ROC evaluates the model's ability to distinguish benign and malicious traffic across different classification thresholds (Fawcett, 2006).

Model evaluation employs Stratified 10-Fold Cross-Validation to maintain approximately consistent class distributions across the ten folds. In each iteration, nine folds are used for training and the remaining fold for validation. The process is repeated ten times, and the final performance is calculated by averaging the results across all folds (Kohavi, 1995).

f. Explainability Analysis Using SHAP

This study incorporates Explainable Artificial Intelligence (XAI) using the SHapley Additive exPlanations (SHAP) method to improve the interpretability of machine learning predictions. SHAP quantifies the contribution of individual features to model predictions based on Shapley values from cooperative game theory.

The analysis identifies network traffic features that contribute most significantly to distinguishing benign traffic from DDoS attack traffic. Global explanations are used to examine overall feature importance, while local explanations provide insights into individual classification decisions. This approach enables security analysts to understand the network characteristics influencing attack detection and supports more transparent cybersecurity decision-making (Lundberg et al., 2020; Wang et al., 2020).

g. Deployment

The Deployment stage examines the potential application of the developed models in practical network security environments. The best-performing model can be integrated into an Intrusion Detection System (IDS) or IoT gateway security mechanism to monitor outbound traffic and identify suspicious communication patterns associated with compromised devices.

Additionally, SHAP analysis provides interpretable information regarding the features contributing to attack detection, which can support security analysts in developing appropriate network monitoring policies and mitigation strategies. This integration is expected to strengthen the practical applicability of machine learning-based intrusion detection in IoT environments (Doshi et al., 2018; Kumar & Lim, 2019).

Results and Discussion

This section presents and discusses the experimental findings of the proposed machine learning-based approach for detecting IoT botnet-related Distributed Denial of Service (DDoS) attacks through network traffic analysis. The experiments were conducted using the IoT-23 dataset to evaluate the classification performance of Random Forest, XGBoost, and their ensemble model. Particular attention was given to examining the effects of dataset balancing and cross-validation on classification performance, as well as identifying the network traffic characteristics that influence model predictions. Furthermore, Explainable Artificial Intelligence (XAI) using SHapley Additive exPlanations (SHAP) was employed to provide insights into the contribution of individual features to attack detection. The practical

applicability of the proposed approach was also demonstrated through a web-based prediction system. Accordingly, the results and discussion are organized into three main subsections: (1) Model Performance Evaluation and the Impact of Dataset Balancing, (2) Explainable AI-Based Feature Contribution and Interpretation Using SHAP, and (3) Web-Based Implementation and Practical Implications for IoT Network Security. These subsections collectively examine the predictive effectiveness, interpretability, and potential practical application of the proposed approach in IoT network security environments.

a. Model Performance Evaluation and the Impact of Dataset Balancing

The classification performance of the proposed machine learning models was evaluated using the IoT-23 dataset, which comprises network traffic records from four selected scenarios: CTU-7, CTU-34, CTU-35, and CTU-17. These scenarios were selected because they contain botnet-related activities associated with Distributed Denial of Service (DDoS) attack patterns. Following the data integration and sampling procedures, a total of 50,000 network flow records were obtained, consisting of 23,461 benign traffic records and 26,539 DDoS attack records. The classification process utilized ten selected network traffic features representing communication characteristics, including packet frequency, transmitted data volume, connection duration, and protocol interactions. The target variable was represented by the `binary_label` attribute, where 0 indicates benign traffic and 1 represents attack traffic. The dataset characteristics are summarized in Table 1.

Table 1. Characteristics of the IoT-23 Dataset Used in the Experiments

Characteristics	Description	Value	Function
Dataset source	IoT-23	Stratosphere IPS Laboratory	Data source
Dataset scenarios	CTU-7, CTU-34, CTU-35, CTU-17	4 scenarios	Scenario selection
Total observations	Network flow records	50	Classification dataset
Benign records	Normal network traffic	23,461 (46.92%)	Class 0
DDoS records	Malicious network traffic	26,539 (53.08%)	Class 1
Number of features	Selected network attributes	10	Model input
Target variable	`binary_label`	Binary (0 and 1)	Classification output

As presented in Table 1, the dataset contains a relatively balanced distribution between benign and DDoS attack traffic, with 46.92% and 53.08% of observations, respectively. This distribution provides an appropriate basis for evaluating the classification capabilities of the proposed models without substantial dominance by either class. The use of four different IoT-23 scenarios also provides variations in network communication characteristics for the classification process. Furthermore, the selection of ten network traffic features enables the models to examine multiple dimensions of communication behavior. These characteristics are particularly relevant to identifying abnormal network patterns associated with compromised IoT devices. Nevertheless, the dataset composition alone does not establish the ability to detect attacks at an early stage; this capability must be examined

through the characteristics and temporal availability of the analyzed traffic records.

The initial experiments were conducted using the original dataset without applying any class distribution adjustment. Three classification configurations were evaluated: Random Forest, XGBoost, and an ensemble model combining both algorithms. The experimental results indicate that all three configurations achieved relatively high classification performance, with accuracy values exceeding 97%. Random Forest obtained the highest accuracy of 0.9727, followed by XGBoost and the ensemble model, both of which achieved 0.9724. A comparison of their classification performance is presented in Table 2.

Table 2. Model Performance Comparison on the Original and Balanced Datasets

Algorithm	Dataset	Accuracy	Precision	Recall	F1-Score
Random Forest	Original	0.9727	0.9968	0.9516	0.9737
XGBoost	Original	0.9724	0.9961	0.9518	0.9734
Ensemble (RF+XGB)	Original	0.9724	0.9961	0.9518	0.9734
Random Forest	Balanced	0.9752	0.9976	0.9527	0.9746
XGBoost	Balanced	0.9749	0.9969	0.9527	0.9743
Ensemble (RF+XGB)	Balanced	0.9751	0.9973	0.9527	0.9745

Table 2 demonstrates that Random Forest consistently achieved the highest classification performance across both dataset configurations. On the original dataset, Random Forest obtained an accuracy of 0.9727, precision of 0.9968, recall of 0.9516, and F1-score of 0.9737. These results indicate that the model can identify attack traffic with high precision while maintaining a relatively strong ability to detect actual attacks. Following dataset balancing, its accuracy increased to 0.9752, while precision and F1-score improved to 0.9976 and 0.9746, respectively. Although the performance differences among the three configurations are relatively small, the results indicate that Random Forest provides the most consistent classification performance in the reported experiments. Notably, the ensemble configuration did not outperform the individual Random Forest model, suggesting that combining two algorithms does not necessarily produce better results without an effective integration strategy.

To further investigate the effect of class distribution on classification performance, an undersampling technique was applied to the original dataset. This procedure adjusted the number of observations in each class to produce a more balanced distribution. Following this process, the accuracy of Random Forest increased from 0.9727 to 0.9752, while XGBoost improved from 0.9724 to 0.9749. The ensemble model also demonstrated an improvement, increasing from 0.9724 to 0.9751. These results suggest that adjusting the class distribution can contribute to improved classification performance, although the magnitude of improvement remains relatively modest (He & Garcia, 2009).

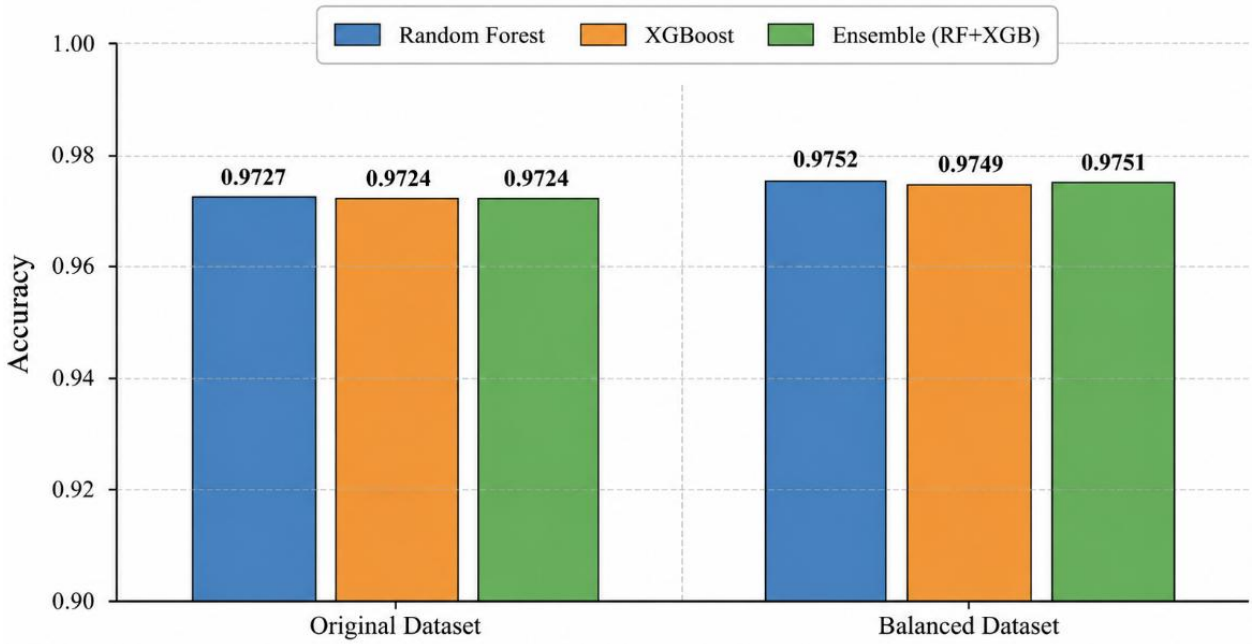

Figure 2. Accuracy Comparison of Random Forest, XGBoost, and Ensemble Models on Original and Balanced Datasets

The accuracy comparison illustrates Figure 2 a consistent improvement across all three classification configurations after dataset balancing. Random Forest demonstrates the highest accuracy in both experimental conditions, followed closely by the ensemble and XGBoost models. The relatively small differences between the models indicate that all three approaches can effectively capture the classification patterns represented in the selected network traffic features. However, the improvement observed after undersampling should be interpreted in relation to the sampling procedure, since removing observations may also affect the diversity of network traffic patterns available during training. Therefore, the performance improvement alone does not establish that undersampling universally enhances model generalization. Further validation using independent test data would provide stronger evidence regarding the robustness of these findings (Arp et al., 2022).

Furthermore, the reliability of the classification models was examined using Stratified 10-Fold Cross-Validation. This evaluation strategy divides the dataset into ten subsets while maintaining approximately consistent class proportions across the folds. During each iteration, nine subsets are used for training, while the remaining subset is used for validation. This process is repeated until every subset has served as the validation data. The reported validation results indicate that Random Forest achieved an accuracy of 0.9757, whereas XGBoost obtained 0.9753. These results are summarized in Table 3.

Table 3. Classification Accuracy Based on Stratified 10-Fold Cross-Validation

Algorithm	Validation Method	Number of Folds	Accuracy	Performance Observation
Random Forest	Stratified K-Fold	10	0.9757	Higher reported accuracy
XGBoost	Stratified K-Fold	10	0.9753	Comparable performance
Ensemble (RF+XGB)	Stratified K-Fold	10	Not reported	Requires additional evaluation

As shown in Table 3, Random Forest achieved a slightly higher accuracy than XGBoost, with a difference of 0.0004. Although this difference is relatively small, it is consistent with the comparative results obtained from the original and balanced datasets, where Random Forest also demonstrated the highest accuracy. The relatively close validation results suggest that both algorithms exhibit comparable classification capabilities under the reported evaluation procedure. Nevertheless, the available results do not provide fold-level performance distributions, standard deviations, or confidence intervals. Therefore, the stability of the models across individual folds and the statistical significance of the observed differences cannot yet be established (Dietterich, 1998). In addition, the cross-validation results for the ensemble model should be reported to enable a complete comparison among the three classification configurations.

Overall, the experimental findings demonstrate that Random Forest provides the highest reported classification performance among the evaluated models, achieving an accuracy of 0.9752 on the balanced dataset and 0.9757 in the reported 10-Fold Cross-Validation results. The improvements observed following undersampling indicate that dataset preparation plays an important role in classification performance. However, the marginal differences between Random Forest, XGBoost, and their ensemble suggest that model selection should not rely exclusively on accuracy. Precision, Recall, and F1-Score must also be considered, particularly because undetected attacks may have significant consequences for IoT network security (Chicco & Jurman, 2020). These findings provide an empirical foundation for selecting Random Forest as the primary model for subsequent SHAP-based interpretability analysis. By examining the features influencing its predictions, the following subsection investigates how network traffic characteristics contribute to distinguishing benign communication from DDoS-related activities.

b. Explainable AI-Based Feature Contribution and Interpretation Using SHAP

Beyond evaluating classification performance, this study employs Explainable Artificial Intelligence (XAI) to investigate the interpretability of the developed machine learning model. Specifically, the SHapley Additive exPlanations (SHAP) method is applied to the Random Forest model to identify the contribution of individual network traffic features to DDoS attack detection. This analysis is important because high classification accuracy does not necessarily explain the factors underlying a model's predictions. In cybersecurity applications, understanding the characteristics that influence attack classification can provide additional insights for security analysts when investigating suspicious network communication and identifying potential botnet activities. SHAP addresses this issue by quantifying the contribution of individual features to model predictions, thereby providing a more transparent interpretation of the classification process (Le et al., 2022; Lundberg & Lee, 2017).

The SHAP analysis begins with the examination of the Summary Plot, which illustrates the distribution and contribution of individual features across the evaluated network traffic records. Each observation is represented by a point, while its horizontal position indicates the corresponding SHAP value. Positive SHAP values indicate that a feature contributes toward the prediction of attack traffic, whereas negative values indicate contributions toward benign traffic classification. The color gradient represents the original feature values, with blue indicating relatively low values and red representing relatively high values. The visualization enables the examination of both the magnitude and distribution of feature

contributions, providing an overview of the network characteristics that influence the Random Forest model's decisions.

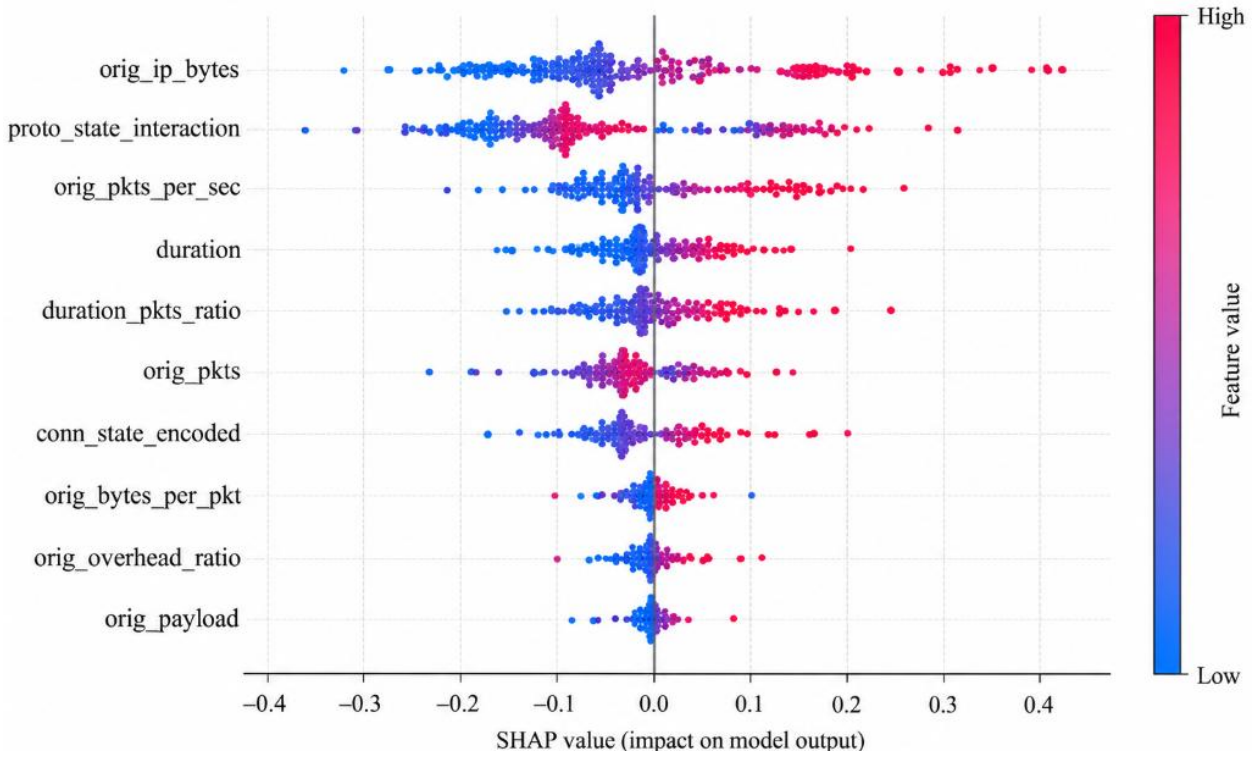

Figure 3. SHAP Summary Plot of the Random Forest Model

Figure 3 illustrates that `orig_ip_bytes` and `proto_state_interaction` exhibit substantial contributions to the model's predictions. Their SHAP value distributions indicate that these features have considerable influence on distinguishing benign and malicious network traffic. The `orig_ip_bytes` feature represents the volume of transmitted data, while `proto_state_interaction` captures characteristics associated with network protocol and connection-state interactions. Their importance suggests that data transmission volume and communication behavior provide meaningful information for identifying suspicious network activities. Furthermore, `orig_pkts_per_sec` also demonstrates a notable contribution, indicating that packet transmission intensity is another relevant characteristic in the classification process. These findings are consistent with the characteristics of DDoS-related communication, in which variations in traffic volume and packet transmission intensity may provide indicators of malicious network behavior.

To provide a more detailed understanding of the SHAP results, the features can be examined according to their relative contribution to the classification process. Based on the feature distribution presented in Figure 2 and the feature ranking in Figure 3, the ten selected attributes demonstrate different levels of influence on the Random Forest model. Their relative importance is summarized in Table 4.

Table 4. Ranking of Network Traffic Features Based on SHAP Analysis

Rank	Feature	Relative Contribution	Interpretation
1	orig_ip_bytes	Very high	Volume of transmitted network data
2	proto_state_interaction	High	Network protocol and connection-state interactions
3	orig_pkts_per_sec	Moderate	Packet transmission intensity per second
4	duration	Moderate	Duration of network communication
5	duration_pkts_ratio	Moderate	Relationship between communication duration and packet count
6	orig_pkts	Moderate	Number of packets transmitted
7	conn_state_encoded	Lower	Encoded network connection states
8	orig_bytes_per_pkt	Lower	Average transmitted bytes per packet
9	orig_overhead_ratio	Relatively low	Network communication overhead
10	orig_payload	Relatively low	Characteristics of transmitted payload

As presented in Table 4, orig_ip_bytes is the most influential feature in the Random Forest classification model, followed by proto_state_interaction and orig_pkts_per_sec. This ranking indicates that network communication characteristics associated with transmitted data volume, protocol interactions, and packet transmission intensity play important roles in distinguishing benign traffic from DDoS attack traffic. Meanwhile, features such as duration, duration_pkts_ratio, and orig_pkts provide additional information regarding communication behavior. In contrast, orig_payload and orig_overhead_ratio exhibit relatively smaller contributions. These findings suggest that flow-level communication characteristics provide more influential information for the classification process than payload-related characteristics in the dataset examined. The ranking also provides a basis for prioritizing network features when investigating suspicious communication patterns.

Furthermore, the overall contribution of each feature was evaluated using the mean absolute SHAP value. This measurement represents the average magnitude of a feature's contribution to model predictions, irrespective of whether its influence increases or decreases the predicted probability of an attack. The results are visualized in Figure 4 through a feature importance bar chart, in which features are arranged according to their average absolute SHAP values.

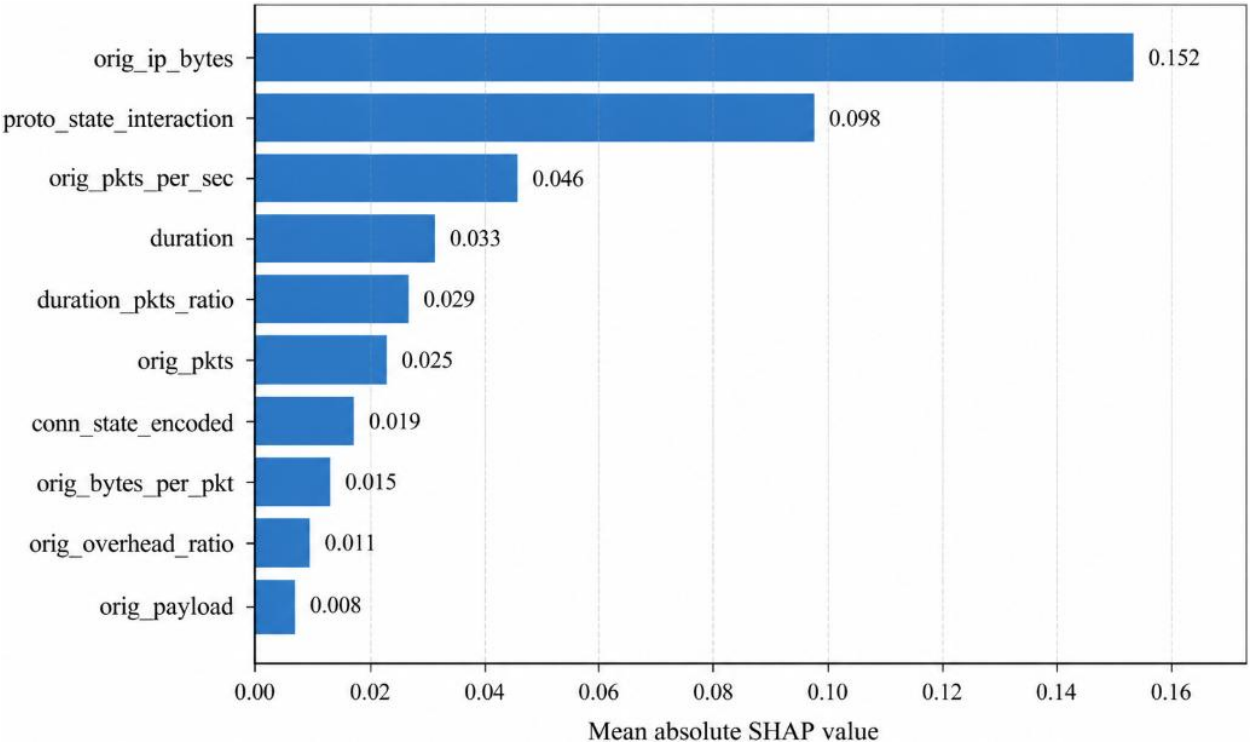

Figure 4. SHAP-Based Global Feature Importance of the Random Forest Model

Figure 4 provides a complementary perspective to the Summary Plot by emphasizing the overall magnitude of each feature's contribution. The visualization shows that `orig_ip_bytes` has the highest mean absolute SHAP value, followed by `proto_state_interaction`, with a noticeable difference compared with the remaining features. This indicates that these two attributes contribute substantially to the model's classification decisions across the evaluated observations. Meanwhile, `orig_pkts_per_sec`, `duration`, and `duration_pkts_ratio` demonstrate smaller but still relevant contributions. The lower importance of `orig_payload` and `orig_overhead_ratio` suggests that these attributes have a comparatively limited influence on the model's overall predictions. Together, the two SHAP visualizations demonstrate that feature importance should be interpreted not only through its ranking but also through the distribution and magnitude of individual feature contributions.

To further interpret these findings from a network security perspective, the selected features can be grouped according to their respective communication characteristics. This grouping helps explain how different dimensions of network traffic contribute to the identification of suspicious activities. Table 5 presents the relationship between the principal feature groups, their associated attributes, and their implications for IoT botnet-related DDoS detection.

Table 5. Interpretation of SHAP Features and Their Implications for IoT DDoS Detection

Feature Category	Associated Features	Observed Importance	Security Interpretation
Data transmission volume	`orig_ip_bytes`	Highest	Represents transmitted data volume that contributes strongly to attack classification

Feature Category	Associated Features	Observed Importance	Security Interpretation
Protocol interaction	`proto_state_interaction`, `conn_state_encoded`	High to lower	Represents communication and connection-state characteristics
Packet transmission intensity	`orig_pkts_per_sec`, `orig_pkts`	Moderate	Describes packet transmission frequency and volume
Communication duration	`duration`, `duration_pkts_ratio`	Moderate	Represents temporal characteristics of network communication
Packet-level data characteristics	`orig_bytes_per_pkt`, `orig_payload`	Lower	Represents transmitted data size and payload-related characteristics
Network overhead	`orig_overhead_ratio`	Relatively low	Represents the proportion of communication overhead

Table 5 demonstrates that the features contributing to the classification process represent several complementary dimensions of network communication. Data transmission volume emerges as the most influential characteristic, while protocol interaction and packet transmission intensity provide additional information for distinguishing suspicious traffic. Temporal characteristics, including communication duration and packet-to-duration relationships, also contribute to the model's predictions. These findings indicate that IoT botnet-related DDoS detection is influenced by multiple network flow characteristics rather than a single traffic attribute. Nevertheless, the relative importance reported by SHAP reflects the model's learned predictive relationships and should not be interpreted as direct evidence of causal relationships between individual features and attack occurrence (Rudin, 2019).

From a cybersecurity perspective, the dominance of `orig_ip_bytes` suggests that the volume of transmitted data is an important indicator in distinguishing normal and malicious network communication. Similarly, the contribution of `orig_pkts_per_sec` highlights the relevance of packet transmission intensity in identifying abnormal traffic behavior. These characteristics are particularly relevant to DDoS-related activities, which may involve substantial changes in network traffic volume and packet transmission patterns. Meanwhile, `proto_state_interaction` provides additional information about communication behavior that may help distinguish legitimate network connections from suspicious interactions associated with compromised devices. However, these features should be interpreted collectively because individual traffic characteristics may overlap between benign and malicious communication.

Overall, the SHAP findings demonstrate that Random Forest provides not only strong classification performance but also interpretable information regarding the network traffic characteristics influencing its predictions. The identification of `orig_ip_bytes`, `proto_state_interaction`, and `orig_pkts_per_sec` as the three most influential features provides useful insights into the characteristics associated with IoT botnet-related DDoS traffic. These findings are consistent with previous studies demonstrating the effectiveness of machine learning approaches, particularly Random Forest and XGBoost, in network intrusion detection (Dhaliwal et al., 2018; Hasan et al., 2019; Resende & Drummond, 2018). The integration of SHAP further strengthens the analytical value of the proposed approach by enabling security analysts to understand the factors underlying classification decisions. Consequently, combining ensemble-based classification with explainable AI provides an additional foundation for

developing more transparent and informative IoT network intrusion detection systems.

c. Web-Based Implementation and Practical Implications for IoT Network Security

Following the evaluation of classification performance and SHAP-based feature interpretation, the developed Random Forest model was implemented in a web-based prediction system to demonstrate its potential application in IoT network security monitoring. This implementation aims to translate the experimental findings into a practical interface that enables users to analyze network traffic and obtain classification results without directly interacting with the underlying machine learning implementation. The system provides an interface through which users can input network flow parameters and obtain predictions indicating whether the observed traffic is classified as benign or associated with a DDoS attack. In addition to the classification results, the system presents the corresponding probabilities for both traffic categories and information regarding the features influencing the prediction.

The web-based implementation represents an important component of the research because it demonstrates how machine learning-based classification can be incorporated into an accessible analytical environment. Rather than presenting prediction results solely as numerical evaluation metrics, the system translates model outputs into information that can be interpreted by users. This approach is particularly relevant to network security analysis, where the ability to identify suspicious traffic and understand the factors underlying classification decisions is essential for supporting further investigation.

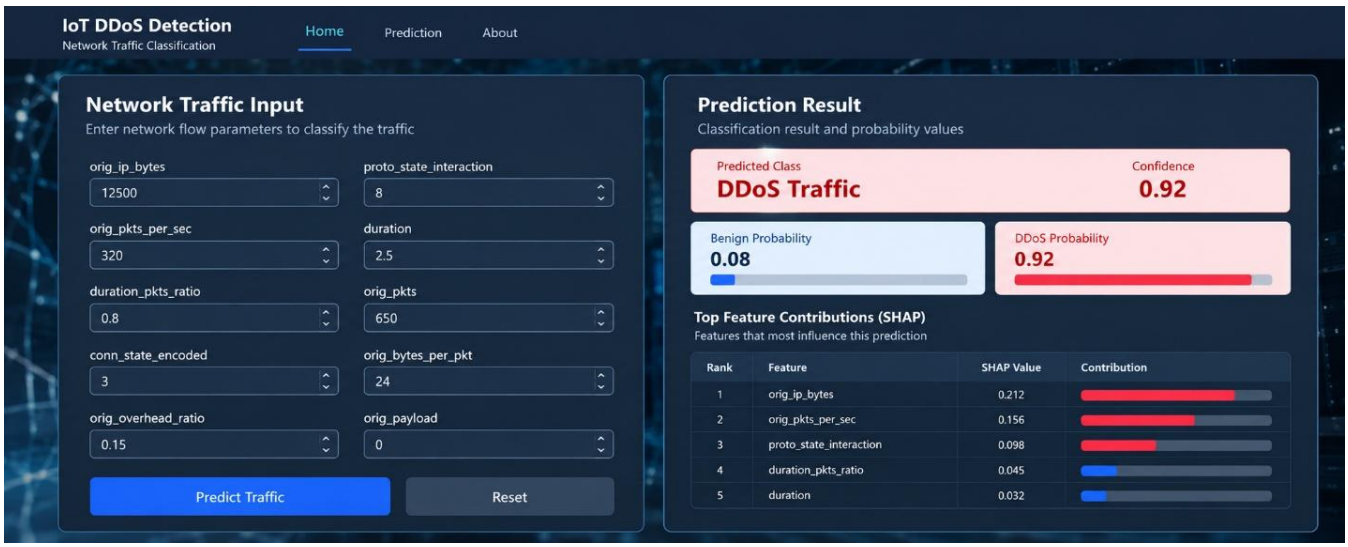

Figure 5. Web-Based Implementation of the IoT DDoS Traffic Prediction System

Figure 5 illustrates the implemented web-based prediction interface, which consists of a network traffic input section and a prediction result display. Users can provide network flow parameters, after which the system processes the input using the trained classification model. The resulting output presents the predicted traffic category and the associated probabilities of benign and DDoS traffic. The interface also provides information about feature contributions, enabling users to examine the characteristics influencing the prediction. This implementation demonstrates the integration of machine learning classification and explainability within a

single application environment. Consequently, the system functions not only as a traffic classification tool but also as an analytical interface for supporting the interpretation of potentially malicious network activities.

To provide a clearer understanding of the system's operational components, the main functionalities of the web-based application are summarized in Table 6.

Table 6. Main Functional Components of the Web-Based Prediction System

Component	Input/Output	Main Function	Expected User Benefit
Network traffic input	Network flow parameters	Accepts traffic characteristics from users	Facilitates traffic analysis
Prediction processing	Input network features	Processes inputs using the trained Random Forest model	Automates traffic classification
Traffic classification	Benign or DDoS label	Identifies the predicted traffic category	Supports attack identification
Probability estimation	Benign and DDoS probabilities	Presents prediction probabilities	Provides additional prediction information
Feature contribution analysis	Influential network features	Displays information about factors influencing predictions	Improves prediction interpretability
Result visualization	Classification and analytical outputs	Presents prediction results through the web interface	Simplifies result interpretation

As summarized in Table 6, the system integrates several functional components that support the transformation of network traffic inputs into interpretable classification outputs. The prediction processing component serves as the core mechanism by applying the trained Random Forest model to the submitted network flow parameters. Meanwhile, the classification and probability estimation components communicate the prediction results in a form that is easier to understand. The inclusion of feature contribution information provides an additional analytical dimension by allowing users to examine the network characteristics associated with the model's predictions. These functionalities demonstrate that the application extends beyond conventional binary classification by incorporating information that supports the interpretation of machine learning decisions.

Furthermore, the integration of classification probabilities and feature contribution information provides a more comprehensive representation of prediction outcomes. In conventional classification systems, users generally receive a predicted class without sufficient information about the factors influencing the decision. In contrast, the developed application presents both the predicted category and its associated probabilities, allowing users to examine the model's assessment of the submitted network traffic. The probability information is particularly useful for understanding how the model distinguishes between benign and malicious traffic, although it should not be interpreted as a direct measurement of the actual likelihood of an attack without appropriate probability calibration (Niculescu-Mizil & Caruana, 2005).

The classification outputs provided by the application can be further described in terms of their analytical functions, as presented in Table 7.

Table 7. Classification Outputs and Their Interpretation in the Web-Based System

Output Component	Description	Interpretation	Security Relevance
Predicted class	Benign or DDoS	Indicates the model's classification decision	Identifies potentially malicious traffic
Benign probability	Probability associated with benign traffic	Represents the model's support for normal traffic classification	Helps assess normal traffic predictions
DDoS probability	Probability associated with DDoS traffic	Represents the model's support for attack classification	Helps identify suspicious traffic
Feature contribution	Information about influential features	Explains factors influencing the prediction	Supports investigation of traffic characteristics
Combined prediction result	Classification and probability information	Provides an overall representation of model output	Facilitates security analysis

Table 7 highlights the importance of presenting classification results through multiple complementary outputs. The predicted class provides the primary detection result, whereas the benign and DDoS probabilities offer additional information regarding the model's classification assessment. Feature contribution information complements these outputs by providing insights into the network characteristics associated with the prediction. This combination is particularly relevant to cybersecurity applications because a classification result should ideally be accompanied by sufficient information to support further examination. For example, when network traffic is classified as DDoS, analysts can examine the contributing features to understand whether the prediction is associated with data transmission volume, packet transmission intensity, or protocol interaction patterns. Such information can help prioritize suspicious traffic for additional investigation, although the final security response should also consider other network evidence.

From an operational perspective, the developed application provides an initial foundation for integrating machine learning-based traffic classification into IoT network security monitoring. Its functionality can be understood through three interconnected activities: network traffic input, automated classification, and interpretation of prediction results. Figure 6 is proposed to illustrate this operational relationship, emphasizing how network flow information is transformed into classification outputs and explanatory information.

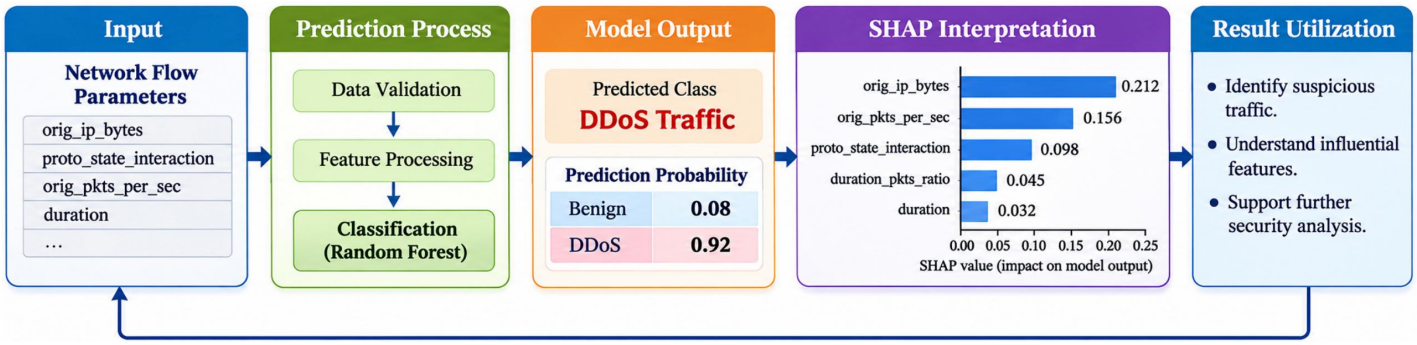

Figure 6. Operational Workflow of the Web-Based IoT DDoS Traffic Prediction System

The operational workflow provides a conceptual representation of how the application processes network traffic information. It begins with the submission of network flow parameters, followed by processing through the trained Random Forest model. The model subsequently produces a predicted traffic category and associated probabilities. SHAP-based feature contribution information is then presented to support the interpretation of the prediction. This workflow illustrates the relationship between predictive classification and explainability, emphasizing that both components contribute to the analytical functionality of the application. However, the workflow represents the system's reported prediction process rather than demonstrating automated real-time network monitoring or direct integration with operational network infrastructure.

To further clarify the relationship between the application's functionality and its potential contribution to network security analysis, Table 8 presents the main operational activities and their corresponding implications.

Table 8. Operational Functions and Practical Implications of the Proposed System

Operational Activity	System Capability	Analytical Contribution	Practical Implication
Traffic submission	Accepts network flow parameters	Provides data for classification	Enables user-driven traffic analysis
Automated prediction	Applies the trained Random Forest model	Distinguishes benign and DDoS traffic	Supports initial attack identification
Probability presentation	Displays benign and DDoS probabilities	Provides additional prediction information	Supports assessment of model outputs
Feature interpretation	Presents feature contribution information	Identifies influential traffic characteristics	Assists suspicious traffic investigation
Result examination	Combines prediction and explanation outputs	Provides interpretable classification information	Supports security-related decision-making

Table 8 demonstrates how the application's operational activities contribute to the overall analytical process. The traffic submission and automated prediction stages establish the basic classification mechanism, while probability presentation and feature interpretation provide additional information for examining the results. The integration of these functions allows users to move beyond simply identifying a predicted attack category toward understanding the network characteristics associated with that classification. From a practical perspective, this functionality may support preliminary security investigations and assist analysts in identifying traffic records that require further examination. Nevertheless, the system should be considered a decision-support component rather than a fully autonomous security mechanism, since the current implementation relies on user-submitted network flow parameters.

In addition to the operational workflow, the integration of prediction results and feature explanations represents an important aspect of the proposed system. The relationship between these outputs can be further illustrated through a conceptual diagram showing how classification results are connected to SHAP-based interpretation. Figure 7 is proposed to demonstrate this relationship, particularly how the predicted traffic category, associated probabilities, and influential network features can collectively support the interpretation of suspicious network activities.

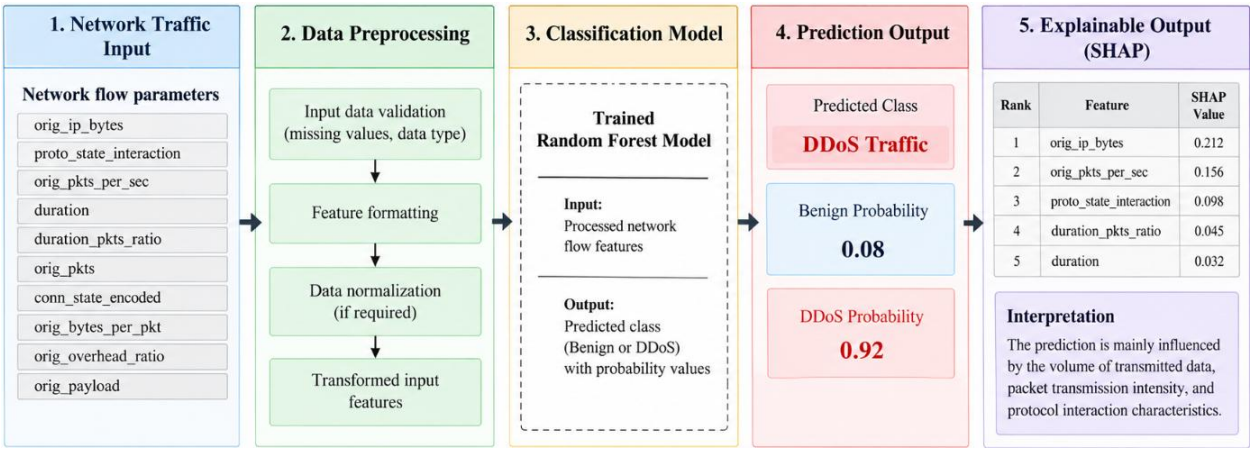

Figure 7. Integration of Machine Learning Prediction and SHAP-Based Explainability in the Web-Based IoT Security System

Figure 7 is intended to emphasize the complementary roles of machine learning classification and explainable artificial intelligence. The Random Forest model provides the classification output, while SHAP offers information regarding the contribution of individual network traffic features to the prediction. By presenting these outputs together, the system provides a more informative representation of the classification process. This integration is particularly relevant when examining suspicious network traffic because security analysts require not only information about the predicted attack category but also an understanding of the characteristics that influence the model's decision. Consequently, the combination of prediction and explanation can improve the usefulness of machine learning outputs in supporting network security analysis.

Overall, the web-based implementation demonstrates the practical potential of integrating machine learning classification with SHAP-based explainability for IoT network traffic analysis. The system provides an accessible interface for submitting network flow parameters, obtaining benign and DDoS traffic predictions, and examining the features influencing classification outcomes. These capabilities are consistent with previous studies highlighting the effectiveness of Random Forest and XGBoost in network intrusion detection, while the integration of SHAP provides an additional dimension of interpretability (Abdalgawad et al., 2022; Le et al., 2022; Oseni et al., 2023). Nevertheless, the current implementation demonstrates user-driven prediction based on submitted network flow parameters and does not yet establish continuous network monitoring, automated traffic interception, or real-time prevention of DDoS attacks. Further development involving integration with network gateways, automated traffic collection, and real-time evaluation would be necessary to establish its operational effectiveness in actual IoT infrastructures.

Conclusion

This study demonstrates the potential of integrating machine learning and Explainable Artificial Intelligence (XAI) for detecting IoT botnet-related Distributed Denial of Service (DDoS) attacks through network traffic analysis. Using 50,000 network flow records from four IoT-23 scenarios, the experimental results indicate that Random Forest achieved the best

overall classification performance, with an accuracy of 0.9752, precision of 0.9976, recall of 0.9527, and F1-score of 0.9746 on the balanced dataset. Furthermore, the reported Stratified 10-Fold Cross-Validation results demonstrate comparable classification performance, with Random Forest achieving an accuracy of 0.9757. The SHAP analysis identified `orig_ip_bytes`, `proto_state_interaction`, and `orig_pkts_per_sec` as the most influential features, highlighting the importance of data transmission volume, protocol interactions, and packet transmission intensity in distinguishing benign traffic from DDoS-related activities. In addition, the implementation of a web-based prediction system demonstrates the practical potential of combining automated traffic classification with interpretable feature contributions to support network security analysis. Overall, the findings highlight that integrating ensemble-based machine learning classification with SHAP-based explainability provides not only reliable attack detection performance but also greater transparency in understanding model predictions. This approach offers a foundation for developing more interpretable IoT intrusion detection systems, while further investigation involving real-time traffic monitoring, deployment at IoT gateways, and evaluation using independent network environments is necessary to establish its effectiveness in operational cybersecurity infrastructures.

References

Abdalgawad, N., Sajun, A., Kaddoura, Y., Zualkernan, I. A., & Aloul, F. (2022). Generative deep learning to detect cyberattacks for the IoT-23 dataset. *IEEE Access, 10*, 6430–6441. https://doi.org/10.1109/ACCESS.2021.3140015

Adadi, A., & Berrada, M. (2018). Peeking inside the black-box: A survey on explainable artificial intelligence (XAI). *IEEE Access, 6*, 52138–52160. https://doi.org/10.1109/ACCESS.2018.2870052

Al-Garadi, M. A., Mohamed, A., Al-Ali, A. K., Du, X., Ali, I., & Guizani, M. (2020). A survey of machine and deep learning methods for Internet of Things (IoT) security. *IEEE Communications Surveys & Tutorials, 22*(3), 1646–1685. https://doi.org/10.1109/COMST.2020.2988293

Antonakakis, M., April, T., Bailey, M., Bernhard, M., Bursztein, E., Cochran, J., Durumeric, Z., Halderman, J. A., Invernizzi, L., Kallitsis, M., Kumar, D., Lever, C., Ma, Z., Mason, J., Menscher, D., Seaman, C., Sullivan, N., Thomas, K., & Zhou, Y. (2017). Understanding the Mirai botnet. In *Proceedings of the 26th USENIX Security Symposium* (pp. 1093–1110). USENIX Association.

Arp, D., Quiring, E., Pendlebury, F., Warnecke, A., Pierazzi, F., Wressnegger, C., Cavallaro, L., & Rieck, K. (2022). Dos and don'ts of machine learning in computer security. In *Proceedings of the 31st USENIX Security Symposium* (pp. 3971–3988). USENIX Association.

Badan Siber dan Sandi Negara. (2025). *Lanskap keamanan siber Indonesia 2024*. Badan Siber dan Sandi Negara.

Barredo Arrieta, A., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., Garcia, S., Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R., & Herrera, F. (2020). Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities

and challenges toward responsible AI. *Information Fusion, 58*, 82–115. https://doi.org/10.1016/j.inffus.2019.12.012

Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research, 13*, 281–305.

Bertino, E., & Islam, N. (2017). Botnets and Internet of Things security. *Computer, 50*(2), 76–79. https://doi.org/10.1109/MC.2017.62

Bilge, L., Balzarotti, D., Robertson, W., Kirda, E., & Kruegel, C. (2012). Disclosure: Detecting botnet command and control servers through large-scale NetFlow analysis. In *Proceedings of the 28th Annual Computer Security Applications Conference* (pp. 129–138). ACM. https://doi.org/10.1145/2420950.2420969

Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

Buczak, A. L., & Guven, E. (2016). A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications Surveys & Tutorials, 18*(2), 1153–1176. https://doi.org/10.1109/COMST.2015.2494502

Capuano, N., Fenza, G., Loia, V., & Stanzione, C. (2022). Explainable artificial intelligence in cybersecurity: A survey. *IEEE Access, 10*, 93575–93600. https://doi.org/10.1109/ACCESS.2022.3204171

Chaabouni, N., Mosbah, M., Zemmari, A., Sauvignac, C., & Faruki, P. (2019). Network intrusion detection for IoT security based on learning techniques. *IEEE Communications Surveys & Tutorials, 21*(3), 2671–2701. https://doi.org/10.1109/COMST.2019.2896380

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794). ACM. https://doi.org/10.1145/2939672.2939785

Chicco, D., & Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics, 21*, Article 6. https://doi.org/10.1186/s12864-019-6413-7

Dhaliwal, S. S., Nahid, A.-A., & Abbas, R. (2018). Effective intrusion detection system using XGBoost. *Information, 9*(7), Article 149. https://doi.org/10.3390/info9070149

Dietterich, T. G. (1998). Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation, 10*(7), 1895–1923. https://doi.org/10.1162/089976698300017197

Dietterich, T. G. (2000). Ensemble methods in machine learning. In *Multiple classifier systems* (Lecture Notes in Computer Science, Vol. 1857, pp. 1–15). Springer. https://doi.org/10.1007/3-540-45014-9_1

Doshi, R., Apthorpe, N., & Feamster, N. (2018). Machine learning DDoS detection for consumer Internet of Things devices. In *2018 IEEE Security and Privacy Workshops (SPW)* (pp. 29–35). IEEE. https://doi.org/10.1109/SPW.2018.00013

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters, 27*(8), 861–874. https://doi.org/10.1016/j.patrec.2005.10.010

Ferrag, M. A., Maglaras, L., Moschoyiannis, S., & Janicke, H. (2020). Deep learning for cyber

security intrusion detection: Approaches, datasets, and comparative study. *Journal of Information Security and Applications, 50*, Article 102419. https://doi.org/10.1016/j.jisa.2019.102419

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics, 29*(5), 1189–1232. https://doi.org/10.1214/aos/1013203451

García, S., Grill, M., Stiborek, J., & Zunino, A. (2014). An empirical comparison of botnet detection methods. *Computers & Security, 45*, 100–123. https://doi.org/10.1016/j.cose.2014.05.011

Garcia, S., Parmisano, A., & Erquiaga, M. J. (2020). *IoT-23: A labeled dataset with malicious and benign IoT network traffic* (Version 1.0.0) [Data set]. Zenodo. https://doi.org/10.5281/zenodo.4743746

Grinsztajn, L., Oyallon, E., & Varoquaux, G. (2022). Why do tree-based models still outperform deep learning on typical tabular data? In *Advances in Neural Information Processing Systems* (Vol. 35, pp. 507–520). Curran Associates.

Gu, G., Perdisci, R., Zhang, J., & Lee, W. (2008). BotMiner: Clustering analysis of network traffic for protocol- and structure-independent botnet detection. In *Proceedings of the 17th USENIX Security Symposium* (pp. 139–154). USENIX Association.

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research, 3*, 1157–1182.

Hasan, M., Islam, M. M., Zarif, M. I. I., & Hashem, M. M. A. (2019). Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches. *Internet of Things, 7*, Article 100059. https://doi.org/10.1016/j.iot.2019.100059

Hassija, V., Chamola, V., Saxena, V., Jain, D., Goyal, P., & Sikdar, B. (2019). A survey on IoT security: Application areas, security threats, and solution architectures. *IEEE Access, 7*, 82721–82743. https://doi.org/10.1109/ACCESS.2019.2924045

He, H., & Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering, 21*(9), 1263–1284. https://doi.org/10.1109/TKDE.2008.239

Khraisat, A., Gondal, I., Vamplew, P., & Kamruzzaman, J. (2019). Survey of intrusion detection systems: Techniques, datasets and challenges. *Cybersecurity, 2*, Article 20. https://doi.org/10.1186/s42400-019-0038-7

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence* (Vol. 2, pp. 1137–1145). Morgan Kaufmann.

Kolias, C., Kambourakis, G., Stavrou, A., & Voas, J. (2017). DDoS in the IoT: Mirai and other botnets. *Computer, 50*(7), 80–84. https://doi.org/10.1109/MC.2017.201

Koroniotis, N., Moustafa, N., Sitnikova, E., & Turnbull, B. (2019). Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset. *Future Generation Computer Systems, 100*, 779–796. https://doi.org/10.1016/j.future.2019.05.041

Kumar, A., & Lim, T. J. (2019). EDIMA: Early detection of IoT malware network activity using machine learning techniques. In *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)* (pp. 289–294). IEEE.

https://doi.org/10.1109/WF-IoT.2019.8767194

Le, T.-T.-H., Kim, H., Kang, H., & Kim, H. (2022). Classification and explanation for intrusion detection system based on ensemble trees and SHAP method. *Sensors, 22*(3), Article 1154. https://doi.org/10.3390/s22031154

Liao, H.-J., Lin, C.-H. R., Lin, Y.-C., & Tung, K.-Y. (2013). Intrusion detection system: A comprehensive review. *Journal of Network and Computer Applications, 36*(1), 16–24. https://doi.org/10.1016/j.jnca.2012.09.004

Lundberg, S. M., Erion, G., Chen, H., DeGrave, A., Prutkin, J. M., Nair, B., Katz, R., Himmelfarb, J., Bansal, N., & Lee, S.-I. (2020). From local explanations to global understanding with explainable AI for trees. *Nature Machine Intelligence, 2*(1), 56–67. https://doi.org/10.1038/s42256-019-0138-9

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems* (Vol. 30, pp. 4765–4774). Curran Associates.

Meidan, Y., Bohadana, M., Mathov, Y., Mirsky, Y., Shabtai, A., Breitenbacher, D., & Elovici, Y. (2018). N-BaIoT—Network-based detection of IoT botnet attacks using deep autoencoders. *IEEE Pervasive Computing, 17*(3), 12–22. https://doi.org/10.1109/MPRV.2018.03367731

Mirkovic, J., Prier, G., & Reiher, P. (2002). Attacking DDoS at the source. In *Proceedings of the 10th IEEE International Conference on Network Protocols* (pp. 312–321). IEEE.

Mirkovic, J., & Reiher, P. (2004). A taxonomy of DDoS attack and DDoS defense mechanisms. *ACM SIGCOMM Computer Communication Review, 34*(2), 39–53. https://doi.org/10.1145/997150.997156

Neshenko, N., Bou-Harb, E., Crichigno, J., Kaddoum, G., & Ghani, N. (2019). Demystifying IoT security: An exhaustive survey on IoT vulnerabilities and a first empirical look on Internet-scale IoT exploitations. *IEEE Communications Surveys & Tutorials, 21*(3), 2702–2733. https://doi.org/10.1109/COMST.2019.2910750

Nguyen, T. D., Marchal, S., Miettinen, M., Fereidooni, H., Asokan, N., & Sadeghi, A.-R. (2019). DÏoT: A federated self-learning anomaly detection system for IoT. In *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)* (pp. 756–767). IEEE. https://doi.org/10.1109/ICDCS.2019.00080

Niculescu-Mizil, A., & Caruana, R. (2005). Predicting good probabilities with supervised learning. In *Proceedings of the 22nd International Conference on Machine Learning* (pp. 625–632). ACM. https://doi.org/10.1145/1102351.1102430

Oseni, A., Moustafa, N., Creech, G., Sohrabi, N., Strelzoff, A., Tari, Z., & Linkov, I. (2023). An explainable deep learning framework for resilient intrusion detection in IoT-enabled transportation networks. *IEEE Transactions on Intelligent Transportation Systems, 24*(1), 1000–1014.

Paxson, V. (1999). Bro: A system for detecting network intruders in real-time. *Computer Networks, 31*(23–24), 2435–2463. https://doi.org/10.1016/S1389-1286(99)00112-7

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research, 12*, 2825–2830.

Resende, P. A. A., & Drummond, A. C. (2018). A survey of random forest based methods for intrusion detection systems. *ACM Computing Surveys, 51*(3), Article 48. https://doi.org/10.1145/3178582

Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence, 1*(5), 206–215. https://doi.org/10.1038/s42256-019-0048-x

Schröer, C., Kruse, F., & Gómez, J. M. (2021). A systematic literature review on applying CRISP-DM process model. *Procedia Computer Science, 181*, 526–534. https://doi.org/10.1016/j.procs.2021.01.199

Shapley, L. S. (1953). A value for n-person games. In H. W. Kuhn & A. W. Tucker (Eds.), *Contributions to the theory of games* (Vol. 2, pp. 307–317). Princeton University Press.

Shone, N., Ngoc, T. N., Phai, V. D., & Shi, Q. (2018). A deep learning approach to network intrusion detection. *IEEE Transactions on Emerging Topics in Computational Intelligence, 2*(1), 41–50. https://doi.org/10.1109/TETCI.2017.2772792

Shwartz-Ziv, R., & Armon, A. (2022). Tabular data: Deep learning is not all you need. *Information Fusion, 81*, 84–90. https://doi.org/10.1016/j.inffus.2021.11.011

Sicari, S., Rizzardi, A., Grieco, L. A., & Coen-Porisini, A. (2015). Security, privacy and trust in Internet of Things: The road ahead. *Computer Networks, 76*, 146–164. https://doi.org/10.1016/j.comnet.2014.11.008

Silva, S. S. C., Silva, R. M. P., Pinto, R. C. G., & Salles, R. M. (2013). Botnets: A survey. *Computer Networks, 57*(2), 378–403. https://doi.org/10.1016/j.comnet.2012.07.021

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management, 45*(4), 427–437. https://doi.org/10.1016/j.ipm.2009.03.002

Sommer, R., & Paxson, V. (2010). Outside the closed world: On using machine learning for network intrusion detection. In *2010 IEEE Symposium on Security and Privacy* (pp. 305–316). IEEE. https://doi.org/10.1109/SP.2010.25

Vinayakumar, R., Alazab, M., Soman, K. P., Poornachandran, P., Al-Nemrat, A., & Venkatraman, S. (2019). Deep learning approach for intelligent intrusion detection system. *IEEE Access, 7*, 41525–41550. https://doi.org/10.1109/ACCESS.2019.2895334

Wang, M., Zheng, K., Yang, Y., & Wang, X. (2020). An explainable machine learning framework for intrusion detection systems. *IEEE Access, 8*, 73127–73141. https://doi.org/10.1109/ACCESS.2020.2988359

Wirth, R., & Hipp, J. (2000). CRISP-DM: Towards a standard process model for data mining. In *Proceedings of the 4th International Conference on the Practical Applications of Knowledge Discovery and Data Mining* (pp. 29–39).

Yin, C., Zhu, Y., Fei, J., & He, X. (2017). A deep learning approach for intrusion detection using recurrent neural networks. *IEEE Access, 5*, 21954–21961. https://doi.org/10.1109/ACCESS.2017.2762418

Zargar, S. T., Joshi, J., & Tipper, D. (2013). A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks. *IEEE Communications Surveys &*

Tutorials, 15(4), 2046–2069. https://doi.org/10.1109/SURV.2013.031413.00127

Zarpelão, B. B., Miani, R. S., Kawakani, C. T., & de Alvarenga, S. C. (2017). A survey of intrusion detection in Internet of Things. *Journal of Network and Computer Applications, 84*, 25–37. https://doi.org/10.1016/j.jnca.2017.02.009

Zhang, Z., Al Hamadi, H., Damiani, E., Yeun, C. Y., & Taher, F. (2022). Explainable artificial intelligence applications in cyber security: State-of-the-art in research. *IEEE Access, 10*, 93104–93139. https://doi.org/10.1109/ACCESS.2022.3204051