

*Research Article*

## Comprehensive Exploration of Ensemble Machine Learning Techniques for IoT Cybersecurity Across Multi-Class and Binary Classification Tasks

Aziz Çetin <sup>1,\*</sup> and Sıtkı Öztürk <sup>2</sup>

<sup>1</sup> Department Electronics and Communication Engineering, Institute of Science, Kocaeli University, Kocaeli 41100, Turkey; e-mail : aziz.cetin4145@outlook.com

<sup>2</sup> Department Electronics and Communication Engineering, Faculty of Engineering, Kocaeli University, Kocaeli 41100, Turkey; e-mail : sozturk@kocaeli.edu.tr

\* Corresponding Author : Aziz Çetin

**Abstract:** This study aimed to predict and detect cyberattacks using hybrid machine-learning models. The CICIOT2023 dataset was utilized for attack prediction and detection, and model performance was evaluated separately by performing thirty-four class (33+1), eight class (7+1), and binary (1+1) classifications according to the types of attacks in the dataset. Voting and stacking hybrid machine learning models were employed in this study, with Logistic Regression (LR), Gaussian Naive Bayes (GNB), and Random Forest (RF) algorithms selected as sub-models. Data preprocessing steps were applied to enhance model performance and hyperparameter optimization was performed. As a result, this study achieved an accuracy of 98% in thirty-four class classifications, 88% in eight class classifications, and 99% in binary classifications with the Voting hybrid machine learning model. In contrast, the Stacking hybrid machine learning model reached an accuracy of 98% in both thirty-four class and eight class classifications and 99% in binary classifications. This study presents a significant innovation in the cybersecurity field by introducing an innovative approach to eliminating the disadvantages of single-model methods.

**Keywords:** Attack Detection; CICIOT2023 Dataset; Cyber Security; Hybrid Model; IoT Security; Machine Learning.

### 1. Introduction

Due to rapid technological advancements and the increasingly complex nature of the digital world, cybersecurity has become a critical research area. Developing effective methods, particularly in network traffic analysis and anomaly detection, is essential for identifying and preventing malicious activities. The main research problem addressed in this study is the inadequacy of existing approaches in the face of increasing cyberattacks on IoT devices. It is estimated that the number of IoT devices worldwide will exceed 20 billion by 2025 and nearly double 40 billion by 2033[1]. These devices often have minimal security measures, making them highly vulnerable to attacks.

The unique characteristics of IoT devices, such as heterogeneity, limited resources, and large-scale deployment, render conventional approaches less effective in such environments. Furthermore, IoT devices can be exploited as attack surfaces to compromise larger networks, necessitating an integrated approach to address both network security and IoT cybersecurity[2].

Existing literature predominantly employs single machine learning models, often failing to achieve robust classification and prediction accuracy for diverse attack types. Therefore, there is a clear need for methods that can provide higher accuracy and broader generalization capabilities[3], [4].

Received: November, 22<sup>nd</sup> 2024

Revised: January, 25<sup>th</sup> 2025

Accepted: January, 26<sup>th</sup> 2025

Published: February, 1<sup>st</sup> 2025

Curr. Ver.: February, 1<sup>st</sup> 2025



Copyright: © 2025 by the authors.  
Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY SA) license (<https://creativecommons.org/licenses/by-sa/4.0/>)

This study chose Voting and Stacking methods as hybrid machine learning models. The primary rationale behind this choice lies in their ability to optimize performance by combining the strengths of different sub-models. The Voting model achieves balanced predictions by aggregating the outputs of base models, while the Stacking model leverages meta-learning to capture complex relationships within the data. These methods are particularly effective for large and diverse datasets as they can accommodate linear and nonlinear patterns.

The CICIoT2023 dataset was selected as the primary data source for this study due to its comprehensive representation of IoT-specific attack scenarios and its real-world applicability[5]. This dataset includes 33 distinct attack types across seven main categories, broadening the scope of the research and enabling a thorough evaluation of model performance across multi-class classification levels.

In the study's first phase, preliminary checks and preprocessing steps were applied to the CICIoT2023 dataset. This process aims to enhance data quality and make it suitable for model training. After completing the preprocessing steps, the resulting data frame was divided into training and test data. Subsequently, various submachine learning models were defined, and hyperparameter optimization was performed for some models.

Following the hyperparameter optimization process, hybrid models were defined and selected. After defining the models, they were trained, and predictions were made. The performance of the models was evaluated using metrics such as accuracy, recall, precision, and F1 score.

In the final phase of this study, the types of attacks in the dataset were revisited through eight-class and binary classifications, and the steps following the preprocessing phase were reapplied to these classifications. The resulting predictions were compared and evaluated to examine the impact of different classification approaches on performance.

This study aims to contribute significantly to the literature by demonstrating the effectiveness of machine learning and hybrid models in cyber-attack detection. The findings of this study can provide valuable insights to network security experts and researchers and serve as a foundation for future studies. The contributions of this paper to the literature can be summarized as follows.

- Applying various preprocessing steps to enhance data quality.
- A hybrid model combining Logistic Regression (LR), an efficient and fast algorithm for linear relationships, Random Forest (RF), which is effective in understanding non-linear relationships and enhancing classification performance, and Gaussian Naive Bayes (GNB), which performs well with high-dimensional data, has been employed. This approach eliminates the disadvantages of single-model methods and achieves higher accuracy rates.
- The systematic evaluation of how Voting and Stacking hybrid models perform across different classification levels (33+1, 7+1, and 1+1) has been presented.
- Comparisons between multi-class classification (33+1 and 7+1) and binary classification (1+1) highlight which metrics are more critical at each classification level.
- Testing the proposed methods on large datasets with multiple classes provides a significant foundation for better understanding the vulnerabilities of IoT devices and developing solutions tailored to the modern threat landscape.

The subsequent section of the paper will discuss similar studies previously conducted in the literature, along with summarized information about these works. Following this, the proposed method, its implementation steps, and the results obtained from its application will be compared with the outcomes of similar studies in the literature. The discussion will conclude with an in-depth analysis of how the hybrid models used in this study have significantly improved the model's predictive capabilities compared to the single-model approaches employed in previous research.

## 2. Related Works

Numerous studies have been conducted in the context of machine learning-based cyberattack classification. Previous studies have generally proposed intrusion detection system (IDS) methodologies using single-level machine learning techniques. These approaches analyze packet-level traffic parameters such as payload size, protocol distribution, and communication frequency to distinguish between malicious and benign traffic.

Sharmin and Kiekintveld [6] analyzed reconnaissance attacks to enhance the security of IoT devices. This study addresses the potential impacts of such attacks on device security by focusing on reconnaissance attacks. The objective was to predict and analyze reconnaissance attacks using flags and time-based attributes. Flags are features used to mark specific activities in network traffic and serve as critical information sources for detecting potential attacks. In addition, time-based attributes refer to data that can be used to analyze attack behavior within a specific time frame. This study aims to optimize the process of detecting and analyzing reconnaissance attacks using these features to enhance the security of IoT devices.

In their study, Sahin O. and Uludag S. [7] proposed an approach to enhance the efficiency of the detection of cyberattacks. This study proposes selecting and using fewer but more significant features instead of utilizing all traditionally used features. Embracing the "less is more" philosophy, this method advocates careful selection and focusing on data features used to detect cyberattacks. By focusing on feature selection for cyberattack detection, this study aims to enhance the effectiveness of cybersecurity systems. In this context, a methodology was developed to identify and select the most important features for detecting cyberattacks using data mining and machine learning techniques. The proposed method attempts to improve the performance of the employed models and cybersecurity systems by ensuring that unnecessary features are not considered in the detection process.

Almaleh et al. [8] attempted to detect API calls made by malware using a hybrid approach that combines traditional LR and Recurrent Neural Network (RNN) models. This study presents an experimental evaluation of this hybrid model using real-world data to assess its effectiveness. The results demonstrate that the proposed method provides higher malware detection accuracy than existing methods. Therefore, this study highlights the effectiveness of combining traditional and deep learning techniques in malware detection.

Wang et al. [9], a lightweight cyberattack detection method based on IoT was examined. This method uses deep learning techniques to learn and detect normal and abnormal behaviors in IoT devices. In addition, by employing dynamic optimization techniques, this study aims to reduce the size of the deep learning model, thereby creating a lightweight cyberattack detection system. Various experimental studies were conducted to evaluate the effectiveness of the proposed method. These experiments were performed on real-world IoT data, and the results demonstrate that the proposed method provides a faster and less resource-intensive cyberattack detection system than other methods.

Haque et al. [10] systematically reviewed data-driven attack detection trends in IoT. This study aims to analyze the development of data-driven attack detection methods in IoT environments and examine the trends in techniques used by reviewing existing research in the literature. Based on the current research in the literature, this study provides a general assessment of data-driven attack detection trends in the IoT. This study adopted a single-model approach, utilizing machine learning techniques such as RF, AdaBoost, Perceptron, LR, and DNN. The CICIoT2023 dataset was employed, with the highest accuracy achieved using the RF model, while the lowest accuracy was observed with the AdaBoost model.

Atay et al. [11] investigated attack detection using single-level and two-level machine learning methods, and they compared the two methods to the CSE-CIC-IDS2018 dataset. In addition, they employed the SMOTE method to address the imbalance in the dataset and compared its impact on performance. The single-level method used the Convolutional Neural Network (CNN), RF, and Light Gradient-Boosting Machine (LGBM) models separately. In contrast, hybrid models such as CNN+RF, LGBM+RF, and RF+RF were used in the two-level method. The best results in the two-level study were obtained with the combined use of CNN and RF models, achieving 98% accuracy and an F1 score of 0.86.

Calp and Bütüner [12] focused on detecting cyberattacks targeting IoT devices using machine learning. They used the DARPA, KDD99, and NSL-KDD datasets. The machine learning models employed in a single-level approach included the Artificial Neural Network (ANN), RF, Naive Bayes (NB), K-Nearest Neighbors (KNN), and LR. The highest performance was achieved by the RF model with a success rate of 99.6%, while the lowest performance was obtained by the NB model with a success rate of 91.7%.

Keskin and Okatan [13] conducted attack detection research using the CSE-CIC-IDS2018 dataset, employing machine learning techniques such as Decision Trees (DT), Extreme Gradient Boosting (XGBoost), Extra Trees (ET), and RF. In this study, where each machine learning model was used separately, the XGBoost algorithm achieved the highest performance with an accuracy rate of 98.18%.

Setiadi et al. [14] propose an effective approach for phishing detection using the Bi-directional Gated Recurrent Units (BiGRU) model combined with feature selection techniques. The research utilizes the Phishing Websites dataset from the UCI Machine Learning Repository, incorporating various preprocessing steps such as data cleaning, normalization, and optimal feature selection. The BiGRU model stands out for its ability to capture temporal dependencies in the data. Experiments conducted with five-fold cross-validation demonstrate exceptional performance, achieving high accuracy, precision, recall, F1 score, and AUC across all metrics. This work highlights the significance of advanced deep learning methods and feature selection in enhancing phishing detection systems.

Dhahir [15] proposes a hybrid approach combining Clustering-Based Local Outlier Factor (CBLOF) and XGBoost for detecting DDoS attacks in network traffic. CBLOF was utilized as a preprocessing step for anomaly detection, while XGBoost effectively categorized these anomalies. The research was conducted on CI-CIDS2017 and CICIDS2018 datasets, achieving exceptional results with 99.99% accuracy, 100% precision, recall, and F1 scores. The hybrid model's success lies in combining CBLOF's anomaly detection capabilities with XGBoost's robust classification power. Additionally, SHAP analysis provided insights into feature contributions, enhancing the interpretability of the decision-making process. This approach sets a new standard for real-time DDoS detection and offers a reliable framework for advancing network security.

Okpor et al. [16] focused on phishing detection in the context of social engineering attacks. They introduced a novel approach utilizing the tree-based Random Forest (RF) algorithm to address data imbalance issues. Various data balancing techniques, such as SMOTE and SMOTEEN, were applied, and their impact on model performance was analyzed. Data collected from social media platforms were classified as phishing or trustworthy content. The RF model demonstrated exceptional performance, achieving 99.73% accuracy and an F1 score of 98.98% with SMOTEEN. These findings highlight the significance of data balancing and feature selection in enhancing the model's generalization capability. Additionally, the study concludes that SMOTEEN outperforms other data balancing techniques in improving model performance.

Htwe et al. propose a machine learning-based system to detect cyberattacks targeting IoT devices [17]. Utilizing the IoT-23 dataset, the research develops a solution tailored to the modern threat landscape. The proposed system employs the Pearson correlation algorithm for feature selection, optimizing feature impact to reduce processing time and improve detection accuracy. Classification and Regression Trees (CART) were applied for attack detection and compared with the Naive Bayes model. The results indicate that the CART algorithm achieved 100% accuracy and outperformed Naive Bayes regarding processing time. This study presents an efficient and lightweight detection system for IoT device security and provides a foundation for future attack classifications.

Nguyen et al. [18] explores the potential applications of Blockchain technology to enhance the security of Wireless Sensor Networks (WSNs). Due to the resource-constrained nature of WSNs, challenges such as data security, integrity, and authentication are addressed. The study discusses how Blockchain's advantages, including de-centralization, transparency, and immutability, can be leveraged to prevent potential attacks in WSNs. Additionally, the effects of Blockchain-based encryption, attack detection mechanisms, consensus algorithms, and smart contracts are examined in detail. The findings demonstrate that Blockchain is an effective tool for mitigating security vulnerabilities in WSNs. This research provides a significant framework for enhancing the reliability of WSNs against modern security threats.

Akazue et al. [19] proposes a hybrid credit card fraud detection method by combining the Random Forest algorithm with three feature selection techniques (Recursive Feature Elimination, Information Gain, and Chi-Squared). Using a dataset from Kaggle, feature selection optimized the model by reducing irrelevant data and accelerating the training process. The proposed model demonstrated excellent performance with 99.6% accuracy, 100% precision, a 99.6% F1 score, and a 95.83% ROC value. The model's reduced training time makes it suitable for real-time fraud detection. This research provides an effective framework for reducing error rates and enhancing the generalization capacity of fraud detection systems.

The existing literature predominantly focuses on single-model approaches, which often demonstrate limited generalization capabilities in complex threat environments. Furthermore, most studies center on general network traffic, with limited research addressing IoT devices' heterogeneous nature and unique threat vectors. Many of these works emphasize binary

classification, prioritizing the distinction between attack and normal traffic rather than conducting a detailed analysis of attack types. Additionally, the datasets commonly used are often low-dimensional and feature limited class diversity, failing to adequately represent the broad spectrum of threats present in IoT environments. This study addresses these gaps by providing a more comprehensive and effective framework for enhancing IoT device security.

### 3. Proposed Method

The successful execution of this study requires various methods, applications, and research efforts. The key element driving the study's success was the hybrid use of different machine-learning models. The diversity of the selected dataset in terms of attack types and classes, the accuracy of the data, the choice of a hybrid model, and the selection of machine learning or deep learning models employed in the hybrid model constitute the main dynamics of this study. Critical steps such as preprocessing based on data accuracy, the accuracy of binary or multi-class classifications, model selection, the ratio of training to test data, and hyperparameter optimization applied to the sub-models are critical for determining the study's accuracy and performance of this study.

Figure 1 illustrates the application steps used in this study. In the first step, the dataset to be used was determined. The next step involved applying preliminary checks and preprocessing to the dataset.

After the preprocessing steps, the resulting new data frame was split into training and test data. Sub-machine learning models were defined following this split, and hyperparameter optimization was applied to some sub-models (e.g., RF-LR). After these processes, hybrid model definitions and selections were made. Following the model definition steps, the model was trained, and predictions were made. Metrics such as accuracy, sensitivity, precision, and F1 score were calculated to evaluate the model's performance. Subsequently, the attack types in the dataset were classified into eight categories and later reduced to binary categories. After the preprocessing stage, the steps were sequentially reapplied, and the obtained predictions were compared and evaluated.

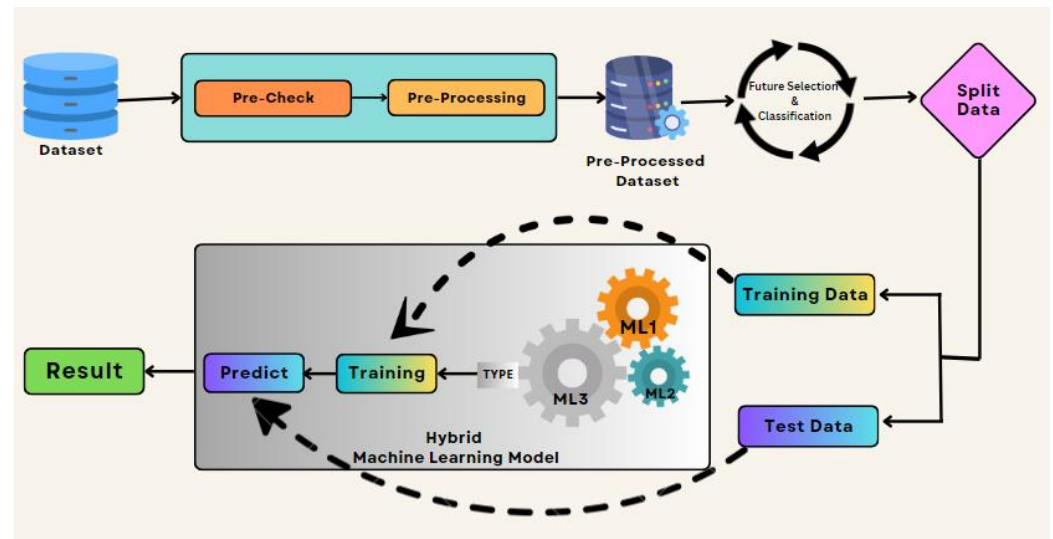


Figure 1. Proposed Method Illustration

#### 3.1. Dataset

The reviewed literature shows that different datasets are typically used, which are relatively limited in terms of attack diversity and classification. Therefore, this study utilized the CICIoT2023 dataset, published in 2023 by the Canadian Institute for Cyber-security. This dataset is one of the most up-to-date and comprehensive datasets specifically designed for IoT devices. This dataset was obtained through tests conducted in a laboratory setup with 105 IoT devices. The dataset consists of observed data representing 33 distinct attack types across seven main attack categories, among the most common cybersecurity threats in the rapidly expanding IoT ecosystem. This makes the dataset suitable for both detailed analyses and generalizable attack classifications.

In contrast, other datasets often only provide two classes (attack and normal traffic), limiting the classification depth. Therefore, this dataset is more appropriate for IoT cybersecurity research. Table 1 presents the attack types included in the CICIoT2023 dataset and their corresponding classes.

**Table 1.** Attack types and classes included in the CICIoT2023 dataset

| Attack Classes | Attack Types                                                                                                                                                                  |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| WEB-BASED      | SQL-Injection; Command-Injection; Backdoor-Malware; Uploading-Attack; XSS; Browser-Hijacking                                                                                  |
| DDOS           | ACK-Fragmentation; UDP-Flood; Slow-Loris; ICMP-Flood; RSTFIN-Flood; PSHACK-Flood; HTTP-Flood; UDP-Fragmentation; ICMP-Fragmentation; TCP-Flood; SYN-Flood; Synonymous-IPFlood |
| MIRAI          | Greip-Flood; Greeth-Flood; UDPPlain                                                                                                                                           |
| BRUTE-FORCE    | Dictionary-Brute-Force                                                                                                                                                        |
| SPOOFING       | Arp-Spoofing; DNS-Spoofing                                                                                                                                                    |
| DOS            | HTTP-Flood; TCP-Flood; SYN-Flood; UDP-Flood                                                                                                                                   |
| RECON          | Ping-Sweep; OS-Scan; Vulnerability-Scan; Port-Scan; Host-Discovery                                                                                                            |

### 3.2. Pre-check and preprocessing

The pre-validation and preprocessing of data within the dataset are among the primary factors directly influencing the results and overall performance[20]. Accordingly, in this study, several pre-validation checks and preprocessing steps were applied to the dataset. These steps included detecting erroneous data, checking for missing data content, identifying duplicate entries, verifying the accuracy of data types, detecting highly correlated data, identifying outliers, and implementing preprocessing methods such as elimination[3] or normalization[21], [22] based on these checks. Figure 2 illustrates the density plots of selected features before and after outlier detection and preprocessing.

### 3.3. Feature selection and classification

In this study, three different attack classifications were considered according to the dataset: The attack classes and types are labeled within the dataset. The data can be divided using these labels into 33+1, 7+1, and 1+1 classes. The 33+1 classification categorizes all types of attacks in the dataset along with the label information for non-attack (benign traffic). The 7+1 classification groups categorize attack types into their respective attack categories and classify them alongside non-attack (benign traffic) label information. The 1+1 classification determines whether the traffic was an attack or a non-attack. Feature selection was performed for each classification and applied to the hybrid machine learning model. Separate results were obtained and compared for each classification.

Feature selection is a preprocessing step applied in machine learning models to eliminate unnecessary or low-importance features, thereby improving the model's efficiency. The CICIoT2023 dataset initially contains 47 features, including statistical network traffic attributes, protocol-related details, and time-based features.

This study chose feature importance ranking, variance thresholding, and mutual information-based selection methods. The number of selected features was initially 47 but was reduced to 38 after feature selection. These methods effectively eliminated unnecessary noise, enhancing the model's generalization capability and overall performance.

### 3.4. Split data

The dataset was divided into two main parts, training and test data, to accurately evaluate the model's training and testing performance. To ensure the reliability of the results, care was taken to maintain the balance of the dataset and preserve equal representation of class distributions. Seventy percent of the dataset was allocated for training, enabling the model to learn the parameters necessary for making predictions. The remaining 30% was reserved for testing, allowing the evaluation of the model's predictive power on previously unseen samples to simulate real-world scenarios. This 70-30 split is a widely adopted standard practice in the literature, as it helps prevent issues such as overfitting or underfitting[23]. The training and

test data were selected randomly while maintaining balanced class distributions, facilitating a robust evaluation of performance metrics such as accuracy, sensitivity, precision, and F1 score.

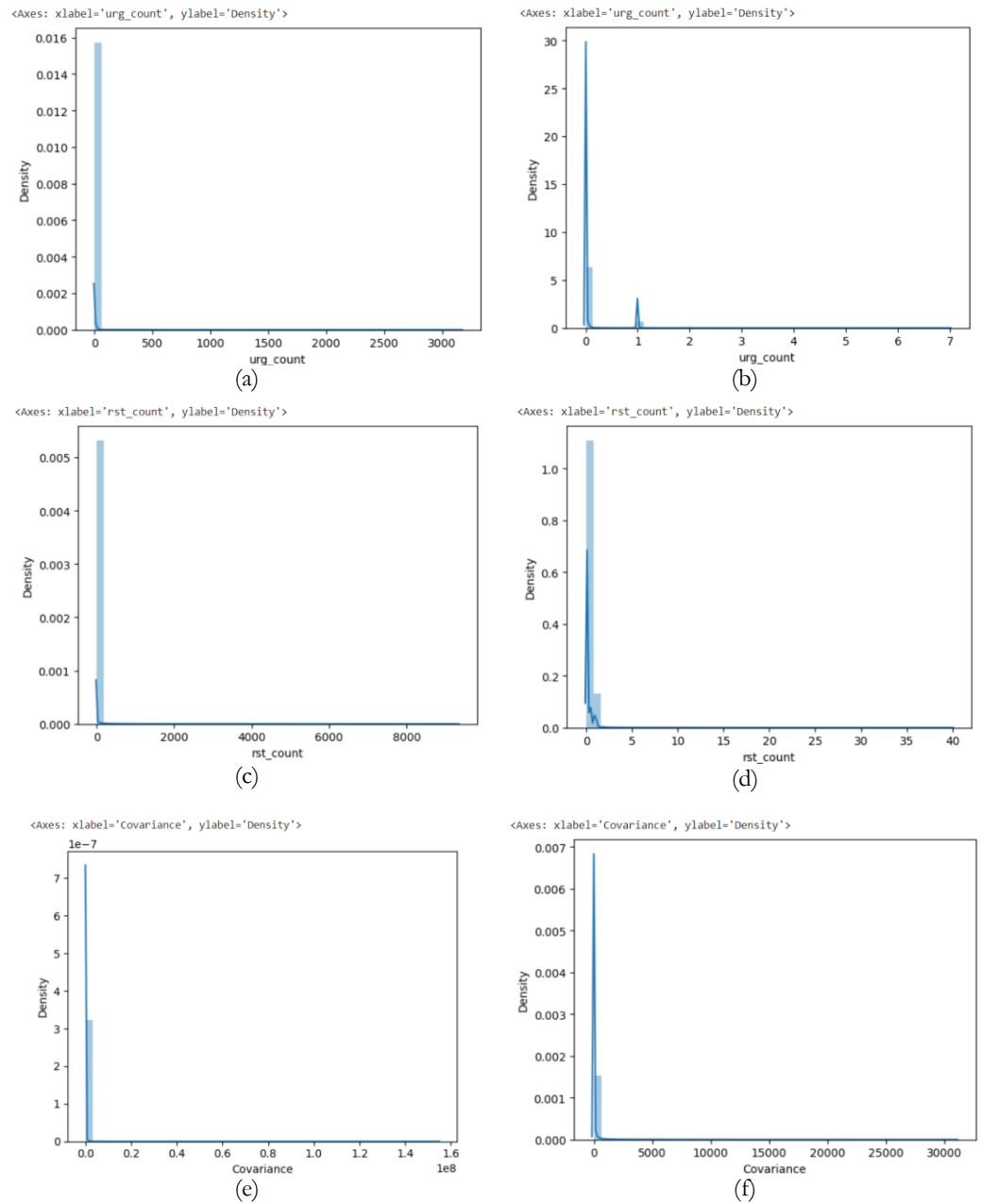


**Figure 2.** Density plot of a feature before preprocessing (a) `urg_count`; (c) `rst_count`; (e) covariance; Density plot of a feature after preprocessing (b) `urg_count`; (d) `rst_count`; (f) covariance.

### 3.5. Hybrid model design and implementation

The reviewed literature observed that different machine learning models were used individually and demonstrated low performance. Therefore, more effective and robust techniques are required to classify and detect cyberattacks. The proposed hybrid machine learning model successfully achieved higher accuracy and reliability by combining different machine learning algorithms and techniques. This study utilized the hybrid machine learning models “Voting Classifier” and “Stacking Classifier”. These models aim to enhance overall performance by combining predictions from multiple sub-models[24], [25].

The Voting model has three different types of classification prediction: “Hard Voting,” “Soft Voting,” and “Weighted Voting.” In this study, the “Soft Voting” method, which

provides better performance, was used. In this method, each sub-model makes predictions based on class probabilities, which enhances the model's ability to generalize, particularly in cases of dataset imbalance. The soft voting method's ability to assess the contributions of sub-models equitably, combined with its significant improvement in accuracy metrics, has established it as the most appropriate choice for this study[26]. Figure 3 illustrates the soft voting model's working principle and prediction mechanism.

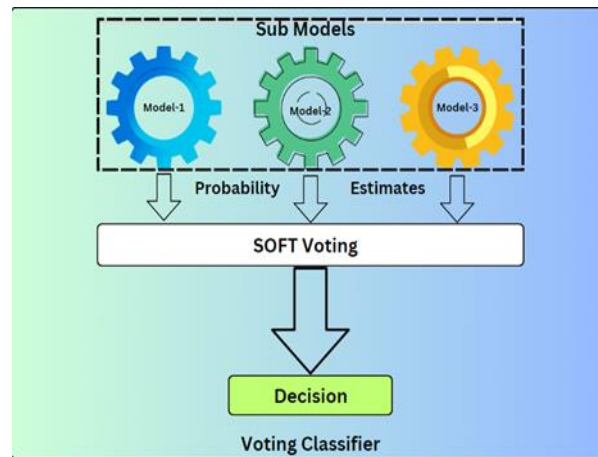


**Figure 3.** Hybrid Soft Voting Classifier Model

“Stacking” or “stacked generalization” is a machine learning model that combines predictions from multiple base classifiers or regression models using a meta-model to solve classification or regression problems. As shown in Figure 4, the predictions from the base models serve as inputs to the meta-model. The meta-model is then trained on these inputs to make the final prediction.

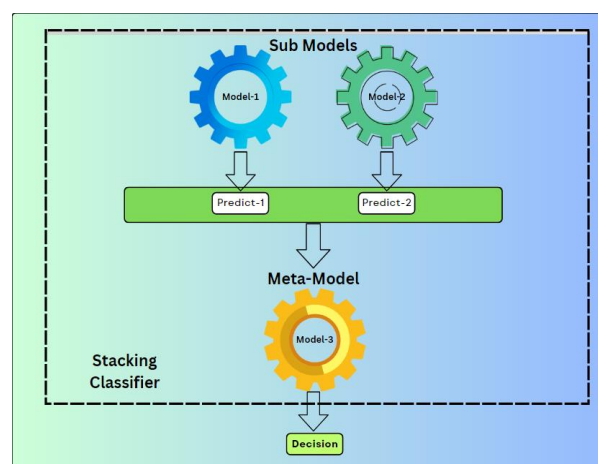


**Figure 4.** Hybrid Stacking Classifier Model

The study used GNB, LR, and RF as sub-models to construct the voting and stacking hybrid machine learning models. Each sub-model was selected for its specific strengths and contributions. Logistic Regression is a fast and lightweight algorithm that excels in identifying linear relationships and performs effectively with linearly separable data. On the other hand, Random Forest is highly effective in capturing non-linear relationships and enhancing classification performance while reducing overfitting. Gaussian Naive Bayes is particularly suited for high-dimensional datasets, offering efficiency in evaluating probabilistic characteristics of classes for classification. Together, these models bring complementary strengths; LR captures linear patterns, RF learns complex non-linear relationships, and GNB efficiently assesses the statistical properties of the dataset. This combination provides a well-balanced set of sub-models that addresses linear and non-linear relationships, thereby improving the system's

generalization capability. The parameter configurations for the ML models are provided in Table 2.

**Table 2.** ML Models Parameter Configuration

| Model                | Configuration                                                                   |
|----------------------|---------------------------------------------------------------------------------|
| Random Forest        | Using random seed=42, n_estimator=100, max_depth=20, min_sample_split=5         |
| Logistic Regression  | Using random seed=42, penalty=l1 & l2, Regularization Strength C=1, solver=saga |
| Gaussian Naive Bayes | Features are fitted, assuming a normal distribution.                            |
| Voting               | The estimation method is soft                                                   |
| Stacking             | Final estimator is GNB                                                          |

### 3.6. Performance metric

The model's performance was measured using evaluation metrics such as accuracy, recall, precision, and the F1 score. The parameters used in the evaluation and their corresponding calculation formulas are described below [27].

The accuracy score is a common metric used to evaluate the overall performance of a classification model. It calculates the ratio of correctly classified instances to the total number of instances in the dataset, which is expressed as a percentage. This metric provides a straightforward measure of how well a model performs. The mathematical formula for accuracy is given in Equation (1).

$$acc = \frac{TP + TN}{TS} \quad (1)$$

The recall score, often called sensitivity or the true positive rate, measures the model's ability to identify true positive cases correctly. It evaluates how effectively the model captures relevant positive instances within the dataset. The recall score can be computed using the formula provided in Equation (2).

$$recall = \frac{TP}{TP + FN} \quad (2)$$

Precision, another critical evaluation metric, assesses how many predicted positive instances are correct. This metric is particularly useful in scenarios where minimizing false positives is important. The calculation for precision is shown in Equation (3).

$$prec = \frac{TP}{TP + FP} \quad (3)$$

Finally, the F1 score serves as a comprehensive metric by combining both precision and recall into a single value. It is the harmonic mean of precision and recall, making it a balanced measure for evaluating the trade-offs between these two metrics. The formula for the F1 score is defined in Equation (4).

$$F1 = \frac{2TP}{2TP + FP + FN} \quad (4)$$

In these equations, FP refers to false positives, FN to false negatives, TP to true positives, TN to true negatives, and TS to total samples.

## 4. Results and Discussion

In this section, the proposed method is evaluated using soft voting and stacking hybrid machine learning models, which were selected for their ability to enhance classification performance. The Soft Voting model combines the predictions of individual sub-models by leveraging class probabilities, which contributes to improved generalization and balanced outcomes. On the other hand, the Stacking model optimizes classification accuracy by integrating sub-model predictions through a meta-model. This approach is particularly effective when the sub-models exhibit diversity, enabling the meta-model to perform better.

The selection of these hybrid methods aligns with the study's objective of improving overall classification accuracy and addressing a broader spectrum of attack scenarios. Both methods deliver robust and reliable classification results by integrating the complementary strengths of the sub-models. The outcomes of the classification process, represented as (33+1, 7+1, and 1+1), are detailed in Figure 5 and Figure 6, demonstrating the efficacy of the proposed approach.

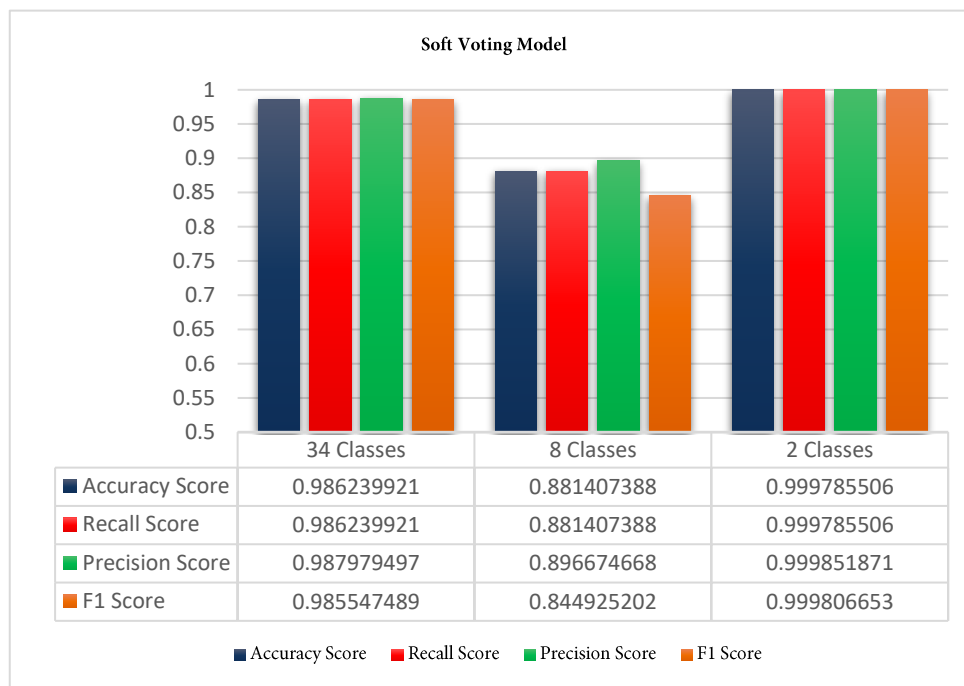


**Figure 5.** Results of Soft Voting Model for Three Different Classifications

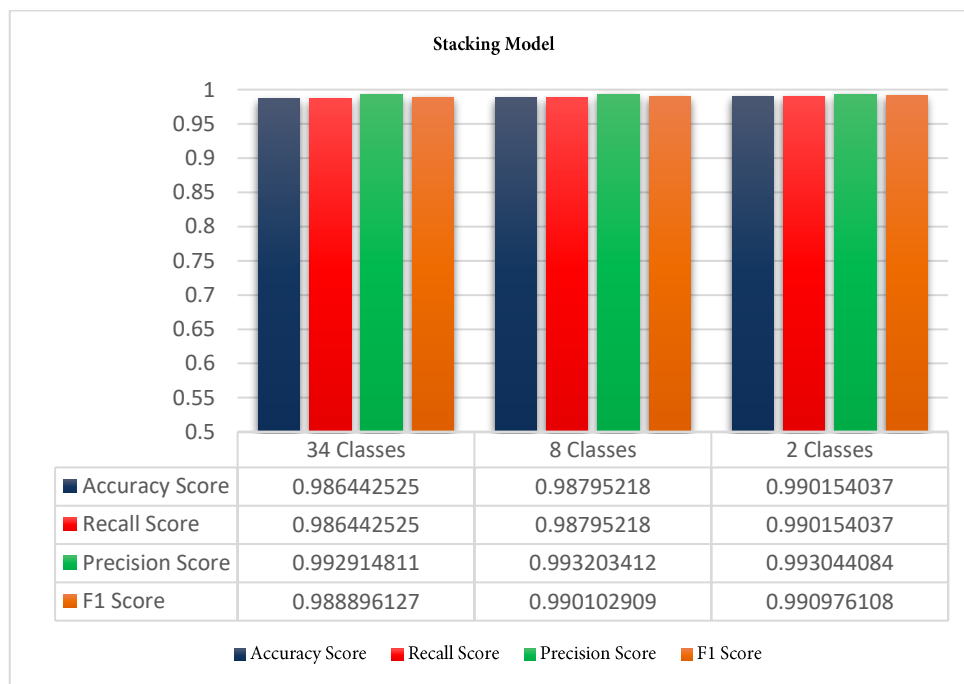


**Figure 6.** Results of Stacking Model for Three Different Classifications

This study analyzed the results obtained using Voting and Stacking models for the 33+1, 7+1, and 1+1 classification levels on the CICIOT2023 dataset. These classification levels highlight various dataset characteristics, each requiring different evaluation metrics. The 33+1

classification, the most complex scenario, involves separating each attack type from non-attack traffic. Accuracy is crucial for assessing overall success at this level, while Precision and Recall metrics are more significant for evaluating specific performance aspects. The Voting model achieved an Accuracy of 98.62%, precision of 98.79%, recall of 98.62%, and an F1 score of 98.55%, indicating high overall performance. However, the Stacking model demonstrated a higher Precision of 99.29%, showcasing its strength in managing false positives more effectively.

For the 7+1 classification, where attacks are grouped into seven main categories, the significance of Recall and F1 metrics increases alongside accuracy, as they evaluate the model's ability to identify and classify broader attack groups. The Voting model achieved 88.14% Accuracy, 89.67% Precision, 88.14% Recall, and an F1 score of 84.49%. In comparison, the Stacking model outperformed with 98.79% Accuracy and 99.31% Precision, demonstrating its effectiveness in reducing false positives and improving accuracy. The simplest classification, 1+1, distinguishes attack traffic from normal traffic (BenignTraffic). Here, the Voting model excelled with near-perfect results, achieving 99.97% Accuracy, 99.98% Precision, 99.97% Recall, and a 99.98% F1 score, while the Stacking model showed slightly lower metrics, achieving 99.01% Accuracy and a 99.09% F1 score. These results suggest that the Stacking model is more suited to complex multi-class scenarios, whereas the Voting model performs better in simpler binary classifications. The confusion matrix results presented in Figure 7 further validate these findings.

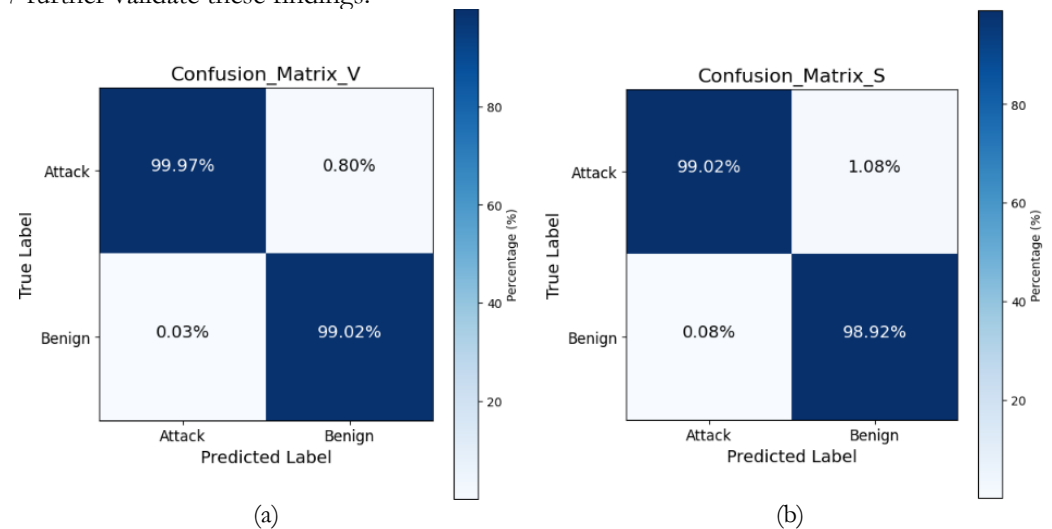


**Figure 7.** The confusion matrix results of the hybrid models, Voting (a) and Stacking (b), obtained through binary classification.

## 5. Comparison

In this section, the results obtained in this study were compared with similar studies in the literature, considering the datasets and machine learning models used. Table 3 presents the accuracy and F1 scores for comparison purposes.

As seen in Table 3, The results show that the hybrid approach using Voting and Stacking models provides significant advantages, especially in the context of precision and recall stability, as reflected by the better F1 score than the previous approach. This is especially relevant in the case of imbalanced datasets, such as CICIoT2023, where the F1 score metric provides a more representative picture than accuracy only.

In multi-class classification (33+1), although the accuracy of the Stacking model (98.6%) is slightly lower than Random Forest (RF) in the study of Neto et al. (99.1%), the Stacking F1 score reaches 98.8%, higher than the RF F1 score (71.4%). This shows that the hybrid model can maintain a balance between precision and recall, which is very important in dealing with uneven class distributions.

In group classification (7+1), the Stacking model again shows superior performance with an F1 score of 99.0%, surpassing RF from Neto et al. (71.9%) and deep learning methods such as BiLSTM and DL-BiLSTM from Wang et al., which each achieved an F1 score of around 91.9%. This stability reflects the hybrid model's ability to capture complex patterns and provide more reliable results in detecting attacks with a wider group.

**Table 3.** Study Results and Performance Comparisons with Literature.

| Method          | Dataset              | Model      | Accuracy     | F1 score     |
|-----------------|----------------------|------------|--------------|--------------|
| Neto et al. [5] | CICIoT2023<br>(33+1) | LR         | 0.802        | 0.493        |
|                 |                      | Perceptron | 0.819        | 0.447        |
|                 |                      | Adaboost   | 0.607        | 0.473        |
|                 |                      | RF         | <b>0.991</b> | 0.714        |
|                 |                      | DNN        | 0.986        | 0.672        |
| Ours            | CICIoT2023<br>(33+1) | Voting     | 0.986        | 0.985        |
|                 |                      | Stacking   | 0.986        | <b>0.988</b> |
| Neto et al.[5]  | CICIoT2023<br>(7+1)  | LR         | 0.831        | 0.539        |
|                 |                      | Perceptron | 0.866        | 0.555        |
|                 |                      | Adaboost   | 0.351        | 0.368        |
|                 |                      | RF         | <b>0.994</b> | 0.719        |
|                 |                      | DNN        | 0.991        | 0.697        |
| Wang et al.[9]  | CICIoT2023<br>(7+1)  | CNN        | 0.9221       | 0.9126       |
|                 |                      | RNN        | 0.9273       | 0.9150       |
|                 |                      | LSTM       | 0.9275       | 0.9152       |
|                 |                      | BiLSTM     | 0.9305       | 0.9173       |
|                 |                      | DL-BiLSTM  | 0.9313       | 0.9194       |
| Ours            | CICIoT2023<br>(7+1)  | Voting     | 0.881        | 0.844        |
|                 |                      | Stacking   | 0.987        | <b>0.990</b> |
| Neto et al.[5]  | CICIoT2023<br>(1+1)  | LR         | 0.989        | 0.876        |
|                 |                      | Perceptron | 0.981        | 0.810        |
|                 |                      | Adaboost   | 0.995        | 0.956        |
|                 |                      | RF         | 0.966        | 0.965        |
|                 |                      | DNN        | 0.994        | 0.940        |
| Ours            | CICIoT2023<br>(1+1)  | Voting     | <b>0.999</b> | <b>0.999</b> |
|                 |                      | Stacking   | 0.990        | 0.990        |

In binary classification (1+1), the Voting model produces almost perfect results with an F1 score of 99.9%, indicating an optimal ability to distinguish between attack and non-attack traffic. Although the RF accuracy of Neto et al. is close to the same result (99.4%), the Voting F1 score shows an advantage in maintaining a balance between precision and recall, making it more relevant in this case. In summary, although accuracy is not the best metric in some scenarios, the stability of precision and recall shown through the F1 score makes the hybrid Voting and Stacking approach more effective in dealing with the challenge of imbalanced datasets. This advantage not only reflects the model's ability to better detect various types of attacks but also makes a significant contribution to modern IoT security solutions.

## 6. Conclusions

The preliminary checks and preprocessing steps applied based on these checks on the selected machine learning models and their use within a hybrid application framework have resulted in higher accuracy values and more reliable outcomes compared to similar studies with the same aim. The experimental results demonstrate that using a hybrid machine learning model for attack classification, prediction, and detection provides a reliable approach that meets industry standards. The proposed method achieves both higher accuracy and more precise results.

The findings indicate that a disciplined approach, robust security strategy, and data analytics integration significantly enhance the performance of attack classification, prediction, and detection. Hybrid machine learning models stand out as an industry-leading approach that effectively addresses the disadvantages of widely used single-model methods. The results of this study underscore the potential of hybrid models to serve as advanced solutions in cybersecurity, setting a new standard for reliability and accuracy in the field.

**Author Contributions:** Conceptualization: A.Ç. and S.Ö.; Methodology: A.Ç.; Software: A.Ç.; Validation: A.Ç. and S.Ö.; Formal analysis: A.Ç. and S.Ö.; Investigation: A.Ç.; Resources: A.Ç. and S.Ö.; Data curation: A.Ç. and S.Ö.; Writing—original draft preparation: A.Ç.; Writing—review and editing: S.Ö.; Visualization: A.Ç.; Supervision: S.Ö.; Project administration: A.Ç.

**Funding:** This research received no external funding.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

- [1] L. S. Vailshery, “Number of IoT connected devices worldwide from 2022 to 2023, with forecast from 2024 to 2033” *Statista*. <https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/>
- [2] A. R. Khan, M. Kashif, R. H. Jhaveri, R. Raut, T. Saba, and S. A. Bahaj, “Deep Learning for Intrusion Detection and Security of Internet of Things (IoT): Current Analysis, Challenges, and Possible Solutions,” *Secur. Commun. Networks*, vol. 2022, pp. 1–13, Jul. 2022, doi: 10.1155/2022/4016073.
- [3] F. Özger, “Makine öğrenmesi algoritmalarının hibrit yaklaşımı ile ağ anomalisi tespiti,” *Sakarya University of Applied Sciences Institutional Repository*. 2023. [Online]. Available: <https://acikerisim.subu.edu.tr/xmlui/handle/20.500.14002/2440>
- [4] J. P. Ntayagabiri, Y. Bentaleb, J. Ndikumagenge, and H. EL Makhtoum, “A Comprehensive Approach to Protocols and Security in Internet of Things Technology,” *J. Comput. Theor. Appl.*, vol. 2, no. 3, pp. 324–341, Dec. 2024, doi: 10.62411/jcta.11660.
- [5] E. C. P. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu, and A. A. Ghorbani, “CICIOT2023: A Real-Time Dataset and Benchmark for Large-Scale Attacks in IoT Environment,” *Sensors*, vol. 23, no. 13, p. 5941, Jun. 2023, doi: 10.3390/s23135941.
- [6] N. Sharmin and C. Kiekintveld, “Enhancing IoT Device Security: Predicting and Analyzing Reconnaissance Attacks Using Flags and Time-Based Attributes,” in *2023 10th International Conference on Internet of Things: Systems, Management and Security (IOTSMS)*, Oct. 2023, pp. 23–30. doi: 10.1109/IOTSMS59855.2023.10325811.
- [7] O. Sahin and S. Uludag, “Advancing Intrusion Detection Efficiency: A ‘Less is More’ Approach via Feature Selection.” Oct. 09, 2023. doi: 10.21203/rs.3.rs-3398752/v1.
- [8] A. Almaleh, R. Almushabb, and R. Ogran, “Malware API Calls Detection Using Hybrid Logistic Regression and RNN Model,” *Appl. Sci.*, vol. 13, no. 9, p. 5439, Apr. 2023, doi: 10.3390/app13095439.
- [9] Z. Wang, H. Chen, S. Yang, X. Luo, D. Li, and J. Wang, “A lightweight intrusion detection method for IoT based on deep learning and dynamic quantization,” *PeerJ Comput. Sci.*, vol. 9, p. e1569, Sep. 2023, doi: 10.7717/peerj-cs.1569.
- [10] S. Haque, F. El-Moussa, N. Komninos, and R. Muttukrishnan, “A Systematic Review of Data-Driven Attack Detection Trends in IoT,” *Sensors*, vol. 23, no. 16, p. 7191, Aug. 2023, doi: 10.3390/s23167191.
- [11] R. Atay, D. E. Odabaş, and M. Kurt Pehlivanoglu, “İki Seviyeli Hibrit Makine Öğrenmesi Yöntemi ile Saldırı Tespiti,” *Gazi J. Eng. Sci.*, vol. 5, no. 3, pp. 258–272, Dec. 2019, doi: 10.30855/gmbd.2019.03.07.
- [12] M. H. Calp and R. Bütünler, “Detecting the Cyber Attacks on IoT-Based Network Devices Using Machine Learning Algorithms,” *Politekn. Derg.*, vol. 27, no. 5, pp. 1971–1989, Oct. 2024, doi: 10.2339/politeknik.1340515.
- [13] S. Keskin and E. Okatan, “Machine Learning Methods for Intrusion Detection in Computer Networks: A Comparative Analysis,” *Int. J. Eng. Innov. Res.*, vol. 5, no. 3, pp. 268–279, Oct. 2023, doi: 10.47933/ijeir.1360141.
- [14] D. R. I. M. Setiadi, S. Widiono, A. N. Safriandono, and S. Budi, “Phishing Website Detection Using Bidirectional Gated Recurrent Unit Model and Feature Selection,” *J. Futur. Artif. Intell. Technol.*, vol. 2, no. 1, pp. 75–83, 2024, doi: 10.62411/faith.2024-15.
- [15] Z. S. Dhahir, “A Hybrid Approach for Efficient DDoS Detection in Network Traffic Using CBLof-Based Feature Engineering and XGBoost,” *J. Futur. Artif. Intell. Technol.*, vol. 1, no. 2, pp. 174–190, Sep. 2024, doi: 10.62411/faith.2024-33.
- [16] M. D. Okpor *et al.*, “Pilot Study on Enhanced Detection of Cues over Malicious Sites Using Data Balancing on the Random Forest Ensemble,” *J. Futur. Artif. Intell. Technol.*, vol. 1, no. 2, pp. 109–123, Sep. 2024, doi: 10.62411/faith.2024-14.
- [17] C. S. Htwe, Z. T. T. Myint, and Y. M. Thant, “IoT Security Using Machine Learning Methods with Features Correlation,” *J. Comput. Theor. Appl.*, vol. 2, no. 2, pp. 151–163, Aug. 2024, doi: 10.62411/jcta.11179.
- [18] M. D. Nguyen, M. T. Nguyen, T. C. Vu, T. M. Ta, Q. A. Tran, and D. T. Nguyen, “A Comprehensive Study on Applications of Blockchain in Wireless Sensor Networks for Security Purposes,” *J. Comput. Theor. Appl.*, vol. 2, no. 1, pp. 102–117, Jul. 2024, doi: 10.62411/jcta.10486.
- [19] M. I. Akazue, I. A. Debekeme, A. E. Edje, C. Asuai, and U. J. Osame, “UNMASKING FRAUDSTERS: Ensemble Features Selection to Enhance Random Forest Fraud Detection,” *J. Comput. Theor. Appl.*, vol. 1, no. 2, pp. 201–211, Dec. 2023, doi: 10.33633/jcta.v1i2.9462.
- [20] T. Johnson, A. J. Liu, S. Raza, and A. McGuire, “A Comparison of Modeling Preprocessing Techniques,” *arXiv*. Feb. 23, 2023. [Online]. Available: <http://arxiv.org/abs/2302.12042>
- [21] S. Abbas *et al.*, “A Novel Federated Edge Learning Approach for Detecting Cyberattacks in IoT Infrastructures,” *IEEE Access*, vol. 11, pp. 112189–112198, 2023, doi: 10.1109/ACCESS.2023.3318866.
- [22] K. R. Narayan, S. Mookherji, V. Odelu, R. Prasath, A. C. Turlapaty, and A. K. Das, “IIDS: Design of Intelligent Intrusion Detection System for Internet-of-Things Applications,” *arXiv*. Aug. 02, 2023. [Online]. Available: <http://arxiv.org/abs/2308.00943>
- [23] C. Karakuş, “Makine Öğrenmesi Algoritmaları,” *Dr. Cahit Karakuş*, 2023. [https://ckk.com.tr/Kitap/Mühendislik/ML\\_01\\_Makine\\_Oğrenmesi\\_Algoritmaları.html](https://ckk.com.tr/Kitap/Mühendislik/ML_01_Makine_Oğrenmesi_Algoritmaları.html)
- [24] Y. Zhou, G. Cheng, S. Jiang, and M. Dai, “Building an efficient intrusion detection system based on feature selection and ensemble classifier,” *Comput. Networks*, vol. 174, p. 107247, Jun. 2020, doi: 10.1016/j.comnet.2020.107247.

- [25] D. H. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, no. 2, pp. 241–259, Jan. 1992, doi: 10.1016/S0893-6080(05)80023-1.
- [26] L. Rokach, "Ensemble-based classifiers," *Artif. Intell. Rev.*, vol. 33, no. 1–2, pp. 1–39, Feb. 2010, doi: 10.1007/s10462-009-9124-7.
- [27] E. U. H. Qazi, A. Almorjan, and T. Zia, "A One-Dimensional Convolutional Neural Network (1D-CNN) Based Deep Learning System for Network Intrusion Detection," *Appl. Sci.*, vol. 12, no. 16, p. 7986, Aug. 2022, doi: 10.3390/app12167986.