Jurnal Nasional Teknik Elektro dan Teknologi Informasi
Volume 14 Number 3 August 2025

Translation of article 10.22146/jnteti.v14i3.19764

Attack Detection in IoT Networks Using Hybrid Feature Selection and Bayesian Optimization

Samsudiat[1], Kalamullah Ramli[1]

[1] Department of Electrical Engineering, Faculty of Engineering, Universitas Indonesia, Depok, Jawa Barat 16424, Indonesia

[Submitted: 6 March 2025, Revised: 29 May 2025, Accepted: 1 August 2025]
Corresponding Author: Kalamullah Ramli (email: kalamullah.ramli@ui.ac.id)

ABSTRACT — Machine learning (ML)-based attack detection is a promising alternative for addressing cybersecurity threats in Internet of things (IoT) networks. This approach can handle various emerging attack types. However, the growing volume of data and the reliance on default parameter values in ML algorithms have led to performance degradation. This study proposed a hybrid feature selection method combined with Bayesian optimization to improve the effectiveness and efficiency of attack detection models. The hybrid feature selection method integrated correlation-based filtering, which aimed to rapidly remove highly correlated features, and feature importance, which aimed to select the most influential features for the model. In addition, Bayesian optimization was employed to efficiently identify the optimal parameter values for lightweight and robust ML algorithms suitable for IoT networks, namely decision tree and random forest. The constructed model was then evaluated using the latest attack dataset, CICIoT2023, which consists of seven types of attacks: DDoS, DoS, Mirai, spoofing, reconnaissance, web-based attacks, and brute force. The evaluation results showed that the hybrid feature selection technique produced a more efficient model compared to several single feature selection methods by selecting 5 out of 46 features. Furthermore, Bayesian optimization successfully identified the optimal parameter values, improving model performance in terms of accuracy, precision, recall, and F1 score up to 99.74%, while reducing computational time by as much as 97.41%. Based on these findings, the proposed attack detection model using hybrid feature selection and Bayesian optimization can serve as a reference for implementing cybersecurity solutions in IoT networks.

KEYWORDS — Internet of Things, Attack Detection, Machine Learning, Feature Selection, Hyperparameter Optimization, Bayesian Optimization.

I. INTRODUCTION

The Internet of things (IoT) is a key enabling technology in the current era of the Industrial Revolution and plays an important role across various sectors, including smart homes, autonomous vehicles, manufacturing industries, and healthcare facilities [1]. The term IoT, also known as smart devices, refers to a collection of electronic devices such as sensors, actuators, and other physical objects that are interconnected via the Internet, enabling them to capture, store, process, and transmit data without human intervention [2]. This technology provides convenience through work automation and efficiency in resource utilization [3].

The rapid growth of IoT devices increases the threat of cyberattacks. According to the 2024 Indonesian Cybersecurity Landscape, the most frequent cyberattack is the Mirai botnet, a type of attack targeting IoT devices [4]. Other vulnerabilities, including limited computational resources, lack of software updates, and weak security mechanisms, also increase exposure to cyber threats [5]. Cyberattacks on IoT devices disrupt device performance, preventing them from operating optimally, and may even cause them to operate out of control. Such incidents can result in damage, including device malfunction, data theft, and disruption of organizational business processes [6].

One of the most promising alternatives to mitigate such threats is machine learning (ML)-based cyberattack detection models [7]. In IoT networks, these models can be deployed in intrusion detection systems (IDS), which monitor and inspect data packets transmitted over the Internet and raise alerts for suspicious activity [8]. IDS plays a critical role in IoT networks because it can adapt to new and evolving cyberattacks. However, the continuous growth of data volume degrades the performance of ML-based models [9]. The growing dimensionality of cyberattack data leads to increased model complexity and higher computational resource demands. Moreover, the reliance on default parameter values in ML algorithms further limits optimal performance [10]. Therefore, feature selection techniques and hyperparameter optimization (HPO) are required to improve the performance of the cyber-attack detection model in IoT networks.

Several studies have proposed ML-based methods to address these challenges, including the development of the CICIoT2023 dataset [11]. CICIoT2023 is a recent dataset generated from seven different types of attacks on IoT networks consisting of 105 devices. The collected data were evaluated using several ML algorithms, and the results showed that random forest (RF) achieved the highest accuracy of 99.68%. However, this study did not evaluate computational efficiency.

Another study utilizing CICIoT2023 built cyberattack detection models using three deep learning (DL) algorithms: deep neural network (DNN), convolutional neural network (CNN), and long short-term memory (LSTM) [12]. The findings indicated that CNN achieved the highest performance, with 99.40% accuracy and 618 s computation time for binary classification, and 99.10% accuracy with 767 s computation time for multiclass classification. In another work, a hybrid classification model combining decision tree (DT), RF, and gradient boosting was proposed [13]. The model achieved 99.51% accuracy with a computation time of 448 s. Similarly, another hybrid classification approach combined three DL algorithms, namely autoencoder (AE), LSTM, and CNN [14]. The results showed an accuracy of 99.15% with an average training time of 150 s per cycle. Although these studies

achieved accuracy levels above 99%, the required computation times remained high, exceeding two minutes.

In the area of feature selection, methods such as filter, wrapper, and embedded approaches have been applied [15]–[17]. Each method has strengths and limitations. Filter methods are efficient in computation time but do not account for model performance or feature interactions. Wrapper methods achieve higher accuracy but require significantly longer computation time. Embedded methods balance computation time and accuracy but are limited to specific algorithms. To address these limitations, hybrid feature selection has been proposed to combine the advantages of these methods.

Hybrid feature selection techniques have been studied by combining two filter methods, namely variance and correlation, for detecting botnet attacks in IoT networks [18]. This method selected 14 optimal features, where DT achieved 100% accuracy with 15.85 s computation time, and naïve Bayes achieved 99.29% accuracy with 2.10 s computation time. Another study combined minimum redundancy maximum relevance and principal component analysis (MRMR-PCA) for detecting distributed denial of service (DDoS) attacks in the IoT network [19]. This method selected 10 optimal features, and the resulting detection model achieved 99.90% accuracy with a computation time of 60.817 s. Although both studies achieved high accuracy with low computation time, they focused only on specific attack types.

Beyond feature selection, model performance also depends on parameter tuning in ML algorithms, where several HPO techniques have been explored. Genetic algorithms have been applied for both feature selection and HPO in RF and eXtreme Gradient Boosting (XGBoost) to detect port scan and distributed denial of service (DDoS) attacks in IoT networks [20]. The results showed that RF achieved 96.36% accuracy with 31.24 s computation time, while XGBoost achieved 96.36% accuracy with 1.82 s computation time. Although computational efficiency was high, the accuracy levels remained relatively low. Another study employed Bayesian optimization for AE and DNN models [21]. The findings indicated that detection accuracy reached 99.99%, but required a computation time of 232.393 s.

Reviews of prior studies shows that ML-based cyberattack detection models have achieved high accuracy levels of over 90%. Nevertheless, there remains room for improvement in simultaneously enhancing accuracy and reducing computational cost. Hence, this study proposes a hybrid feature selection method that integrates correlation-based filtering and feature importance, combined with Bayesian optimization, to improve the performance of ML-based cyberattack detection models. Correlation filtering is intended to quickly remove highly correlated features, while feature importance is used to select features with a significant impact on the model. Bayesian optimization is applied to efficiently identify optimal parameters for lightweight and robust ML algorithms suitable for IoT networks, namely DT and RF. The proposed method is evaluated using the latest cyberattack dataset, CICIoT2023. The findings are expected to serve as a promising alternative for strengthening cybersecurity in IoT networks.

II. ATTACK DETECTION MODELS

A. CYBER ATTACKS ON IoT NETWORKS

The term IoT was first introduced in 1999 by Kevin Ashton, a technology innovator working in a company specializing in radio frequency identification (RFID) [22]. RFID technology can be used to connect various physical objects, or "things," to the internet, enabling data collection and exchange among them. However, as more physical objects are connected to the internet, the risk of cyberattacks increases.

Cyberattacks are defined as criminal acts carried out by individuals or groups that compromise the confidentiality, integrity, or availability of information [23]. These actions may include unauthorized system access, data theft, manipulation, and even destruction of data in computer systems and networks. The objectives of such attacks vary, ranging from stealing sensitive information to disrupting organizational business operations, potentially causing financial, reputational, or even legal damage.

Several types of cyberattacks exist. One of the most common in IoT networks is the botnet [24]. A botnet is malicious software (malware) that infects multiple IoT devices, allowing remote control by attackers to launch larger-scale attacks, such as DDoS [25]. A DDoS attack overwhelms target servers with traffic beyond their capacity, leading to service overload and unavailability, which can disrupt organizational operations. In addition to botnets and DDoS, other types of IoT network attacks include brute force, spoofing, man-in-the-middle, and web-based attacks such as backdoors and command injection [26].

B. MACHINE-LEARNING BASED ATTACK DETECTION

ML is a branch of artificial intelligence in which a computer, or "machine," learns patterns from data through specific algorithms without being explicitly programmed [27]. Unlike conventional computer programs that rely on human instructions, ML systems identify data patterns to provide predictions or decisions. The primary goal of ML is to enhance computer performance in specific tasks to support human activities.

Classification models are one ML approach designed to categorize data into specific classes based on their features. Cyberattack detection in IoT networks is built using classification models, typically implemented in intrusion detection systems (IDS) [28]. IDS devices function to monitor and detect data packet activity traversing the internet and to raise alerts when suspicious activity is observed. IDS detection operates using two approaches: signature-based and anomaly-based. In the signature method, data packets are compared to a database of previously identified attacks, often maintained by security communities or service providers. In the anomaly method, packets are compared to statistically normal traffic patterns. Anomaly detection is advantageous because it can identify novel attack types absent from signature databases by leveraging classification models. These models identify data packets based on feature characteristics such as packet size, count, and rate.

Attack detection models for IoT networks have several distinctive characteristics compared to those for general computer networks [29]. First, IoT devices are directly connected to physical environments, making them vulnerable to physical attacks. Second, IoT networks contain more device types than computer networks, introducing greater complexity due to heterogeneous protocols and communication media. Third, IoT devices have limited computational resources, restricting the implementation of advanced security mechanisms. Therefore, detection models designed for IoT networks need to be both robust and lightweight. Two

classification algorithms considered robust and lightweight, and used in this study, are the DT and the RF.

1) DECISION TREE (DT)

The DT algorithm constructs a learning model in the form of a tree structure consisting of roots, branches, and leaves [30]. Each root, representing dataset features, is chosen based on the most informative attribute using criteria such as gini or entropy. Each root forms branches that represent decision rules, which eventually lead to leaves representing final decision outcomes based on class labels. This recursive process continues until stopping criteria, such as maximum tree depth or minimum sample size, are met.

The DT algorithm has the advantage of being highly interpretable and lightweight, making it suitable for devices with limited computational resources, such as IoT devices. Furthermore, DT demonstrates strong performance in building cyberattack detection models [31]. However, its limitation is susceptibility to overfitting, where models fit training data too closely, reducing accuracy on unseen data.

2) RANDOM FOREST (RF)

RF is an algorithm that combines the power of multiple DTs to improve performance [32]. Each tree is built using different random subsets of data and features, reducing the risk of overfitting associated with a single DT. Then, each tree provides a decision outcome, and the final decision is determined by majority voting, namely selecting the decision value that appears most frequently.

The RF algorithm has the advantage of handling high-dimensional data with strong performance, such as cyberattack data [33]. In addition, it is capable of managing outliers and noise in the dataset. Outliers are data points with significantly different values within the dataset, while noise refers to irrelevant data, such as inconsistent or incomplete records. The limitation of this algorithm is that it requires greater computational resources to construct multiple DT trees.

C. CICIoT2023 DATASET

The CICIoT2023 dataset is a cyberattack dataset for IoT networks published in 2023 by the Canadian Institute for Cybersecurity (CIC), University of New Brunswick (UNB), Canada [34]. The dataset was generated by executing 33 types of attacks on multiple IoT devices, producing 46,686,579 records, which are evaluated and widely used by researchers worldwide. The dataset categorizes attacks into seven classes: DDoS, denial of service (DoS), Mirai, reconnaissance, brute force, spoofing, and web-based attacks. The distribution of attack data is presented in Table I.

The dataset contains three types of features, namely time-based features, traffic-based features, and statistical features, totaling 46 features as shown in Table II. Time-based features capture information related to the timing of activities, while traffic-based features describe the characteristics of data packet traffic. Meanwhile, statistical features provide statistical values derived from the characteristics of multiple packets within the same flow.

TABLE I
NUMBER OF LABEL DATA ON CICIOT2023

Label	Number
DDoS	33,984,560
DoS	8,090,738
Mirai	2,634,124
Benign	1,098,195
Spoofing	486,504
Reconnaissance	354,565
Web-based	24,829
Brute Force	13,064
Total	46,686,579

TABLE II
FEATURES ON CICIOT2023

Dataset CICIoT2023	Features
6 time-based features	Flow_Duration, Duration, Rate, Srate, Drate, IAT
28 traffic-based features	Header_Length, Protocol_Type, FIN_Flag, SYN_Flag, RST_Flag, PSH_Flag, ACK_Flag, ECE_Flag, CWR_Flag, ACK_Count, SYN_Count, FIN_Count, URG_Count, RST_Count, HTTP, HTTPS, DNS, Telnet, SMTP, SSH, IRC, TCP, UDP, DHCP, ARP, ICMP, IPv, LLC
12 statistic-based features	Tot_Sum, Min, Max, Avg, Std, Tot_Size, Number, Magnitude, Radius, Covariance, Variance, Weight

D. FEATURE SELECTION

Feature selection is the process of reducing data dimensionality by selecting and/or removing certain features using a specific technique [35]. The main objective of this process is to reduce complexity and computational resource usage, thereby improving the performance of ML models. In addition, it can reduce the risk of overfitting and simplify model interpretation. In general, there are three categories of feature selection methods, namely filter, wrapper, and embedded. The filter method selects features based on statistical scoring, while the wrapper method selects features based on the scores obtained from model learning results, such as RF. Finally, the embedded method selects features during the model-building process, such as feature importance. The feature selection techniques used in this study are as follows.

1) CORRELATION FILTER

In statistics, correlation measures the relationship between two different variables. The correlation value ranges from –1, which indicates a perfectly inverse relationship, to +1, which indicates a perfectly direct relationship [36]. Within this range, correlation can be categorized into five levels: very low correlation (0–0.2), low correlation (0.2–0.4), moderate correlation (0.4–0.6), high correlation (0.6–0.8), and very high correlation (0.8–1). One commonly used correlation type in ML models is the Pearson correlation, which calculates linear correlation using (1):

$$r = \frac{\sum[(X-\bar{X})(Y-\bar{Y})]}{\sqrt{[\sum(X-\bar{X})^2 \sum(Y-\bar{Y})^2]}} \quad (1)$$

where r is the correlation value; X, Y are the values of variables X and Y at the ith data point; an $\bar{X}, \bar{Y}$ are the mean values of variables X and Y, respectively.

In ML models, highly correlated features do not provide additional information to the model. Feature selection using the correlation filter removes features with very high correlation values, typically at a threshold of 0.8. This technique eliminates multicollinearity, i.e., model instability caused by redundant features. Furthermore, this technique works efficiently since it

only relies on statistical computation without involving ML algorithms.

2) FEATURE IMPORTANCE

When building models using algorithms such as DT or RF, the most informative features are selected during the training process. Implicitly, these algorithms already perform feature selection [37]. The metric used to determine the most informative features is feature importance, which is calculated using the concept of entropy. Entropy measures the homogeneity or disorder of data, ranging from 0 to 1. Values close to 0 indicate relatively homogeneous data, whereas values close to 1 indicate diverse data. The entropy value can be computed using (2):

$$S = -\sum_{i=1}^{c} p_i \, log_2 \, p_i \tag{2}$$

where p_i is the proportion of the element in the ith class within the sample space S, and c is the total number of classes.

After feature importance values are obtained, features are ranked from the highest to the lowest. Feature selection is then performed by choosing the top-ranked features. The main advantage of this technique is that the selected features are guaranteed to be relevant to the ML model. Moreover, it can handle nonlinear relationships between features, which the correlation filter technique cannot capture.

3) HYBRID FEATURE SELECTION

Each feature selection technique has its own strengths and weaknesses. Hybrid feature selection aims to combine its advantages while addressing its limitations. This study adopted a two-stage hybrid feature selection approach. First, the correlation filter was applied to rapidly select a subset of features by removing redundant ones. Next, feature importance was applied to select the most informative subset of features for the model. This combination is expected to reduce data dimensionality while improving the performance of cyberattack detection models in IoT networks.

E. HYPERPARAMETER OPTIMIZATION (HPO)

Every ML algorithm has parameters whose values are set before training begins, commonly referred to as hyperparameters. HPO is the process of finding the optimal values of these hyperparameters [38]. Its objective is to identify hyperparameter configurations that enhance both model performance and generalization. In addition, it improves training efficiency by avoiding manual hyperparameter tuning. Popular HPO techniques include grid search, random search, and Bayesian optimization.

Bayesian optimization is an HPO technique that employs a probabilistic model to search for the best values based on previous search results. The probabilistic model is constructed from the objective function using a Gaussian process (GP) to predict the function values at points that have not yet been evaluated. Based on the GP prediction, an acquisition function is then used to determine the next evaluation point. This technique is more efficient because it focuses only on the more promising search space, unlike the other two techniques that explore many combinations.

III. METHODOLOGY

In general, this study consisted of three stages to build an ML-based detection model, namely data preparation, feature selection, and model evaluation.

A. DATA PREPARATION

Data preparation was the first stage in building a classification model, which transformed raw data into a suitable format ready for ML model training. This stage aimed to improve data quality before further processing and analysis. The processes in this stage included data cleaning, coding, class balancing, and normalization.

Data cleaning was carried out by handling duplicate, empty, undefined, and inconsistent records. Based on the inspection, no duplicate, empty, or undefined data had been found, but three features had contained only the value 0, namely 'Telnet,' 'SMTP,' and 'IRC.' Since features containing only the value 0 did not affect the ML model, these three features were removed. The final result of this process was a dataset consisting of 43 features.

Next, data coding was performed by converting the label 'Benign' as normal data into the value 0, and all other labels as attack data into the value 1. The purpose of this conversion into numerical codes was to make it easier for the ML model to interpret the data. The final result of this process was a dataset with 1,098,195 records labeled as 0 and 45,588,384 records labeled as 1, which had originated from the sum of seven attack types.

Based on the coding results, it was found that the data labels were imbalanced. This condition could cause the model to become biased, i.e., predicting the majority class more frequently than the minority class. Therefore, class balancing was carried out using the random under sampling technique, which randomly removed some records in the majority class until their number equaled the minority class. This technique had the advantage of computational efficiency, particularly for large-dimensional cyberattack datasets [39]. In addition, it also eliminated the risk of data duplication that might occur in oversampling techniques. The final result of this process was a balanced dataset, with both class 0 and class 1 containing 1,098,195 records, resulting in a total of 2,196,390 records.

Finally, normalization was performed to rescale the values of each feature so that they fell within the same range. This step aimed to prevent feature dominance, ensuring that each feature had an equal influence on the ML model [40]. In this study, normalization was carried out using the Min-Max technique, which transformed feature values into a range between 0 and 1, as computed by (3):

$$X_{normalisasi} = \frac{X_{awal} - min(X)}{max(X) - min(X)}. \tag{3}$$

B. ATTACK DETECTION MODEL CONSTRUCTION

The next stage was the construction of the cyberattack detection model, which consisted of feature selection, data splitting, and model training. Feature selection was the initial process in this stage, aiming to select the most informative features using a hybrid selection technique that combined correlation filtering and feature importance. The resulting dataset was then randomly split into 80% training data and 20% testing data. Finally, the training data were used to train ML models using Bayesian optimization on the DT and RF algorithms, while the testing data were used to evaluate the ML models. The parameter search space for DT and RF in Bayesian optimization is presented in Table III.

C. MODEL EVALUATION

Model evaluation aims to measure the performance of the trained ML models in predicting unseen data. For classification

TABLE III
DT AND RF ALGORITHM PARAMETER VALUE RANGE

Parameter	Description	Value Range
criterion	Criteria for selecting features to be used at each node	*gini, entropy*
max_depth	Maximum depth of the tree	2 – 10
min_samples_split	Minimum number of samples needed to separate nodes	2 – 5
min_samples_leaf	Minimum number of samples that must be present in each leaf	2 – 5
n_estimators	Number of Decision Trees used in the Random Forest algorithm	10 – 100

models, evaluation was performed using a confusion matrix, which is an n × n table representing the number of classes, containing the predicted and actual class values. The elements of the confusion matrix included true positive (TP), representing the number of positive records correctly predicted as positive; true negative (TN), representing the number of negative records correctly predicted as negative; false positive (FP), representing the number of negative records incorrectly predicted as positive; and false negative (FN), representing the number of positive records incorrectly predicted as negative. Based on these elements, the evaluation interpretation was carried out using the metrics in (4) to (7).

Equation (4) represents the accuracy metric, i.e., the ratio of correctly predicted samples to the total number of samples.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}. \quad (4)$$

Equation (5) represents the precision metric, i.e., the ratio of correctly predicted attacks to the total predicted attacks. This metric indicates the model's effectiveness in terms of accuracy in predicting cyberattacks.

$$Precision = \frac{TP}{TP+FP}. \quad (5)$$

Equation (6) represents the recall metric, i.e., the ratio of correctly predicted attacks to the total actual attacks. This metric reflects the model's sensitivity in minimizing undetected actual cyberattacks.

$$Recall = \frac{TP}{TP+FN}. \quad (6)$$

Equation (7) represents the F1 metric, which is the harmonic mean of precision and recall. This metric demonstrates the overall performance and balance of the model in predicting data packets.

$$F1 = \frac{2\ x\ Presisi\ x\ Recall}{Presisi+Recall}. \quad (7)$$

Finally, computational time was measured as the total time required to train and test the model, reflecting the efficiency of the model.

IV. RESULTS AND DISCUSSION

The cyber-attack detection model in this study was built on hardware with an Intel Xeon 3.5 GHz processor and 64 GB of RAM, while the software environment used was Jupyter Notebook v.7.2.2 with Python v.3.12.7. Several libraries were employed: Pandas for data preparation, Matplotlib for data visualization, scikit-learn for building and evaluating ML models, and scikit-optimize for Bayesian optimization.

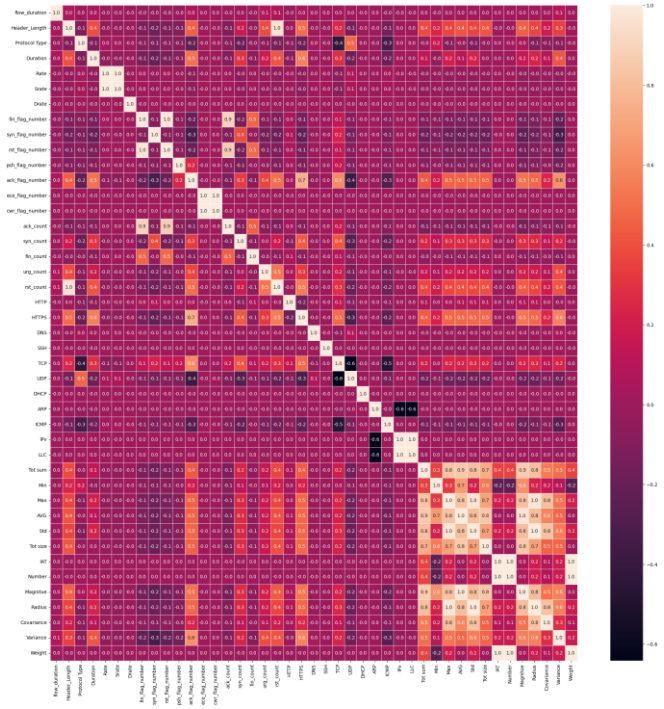

Figure 1. Correlation matrix.

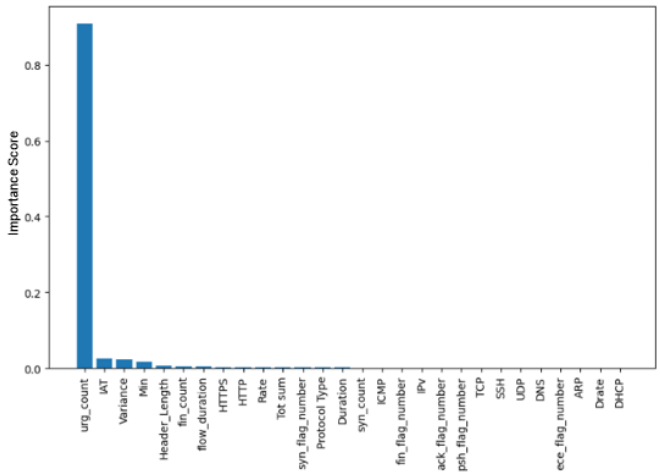

Figure 2. Feature importance measurement.

A. RESULTS OF HYBRID FEATURE SELECTION

The correlation filter technique was the first step in the feature selection stage, aiming to quickly eliminate highly correlated features. This technique was carried out by calculating the correlation values of the 43 features obtained from the data preparation stage. Figure 1 presents the constructed correlation matrix, where lighter colors indicate higher correlation values. This study applied a correlation threshold of 0.8, and 15 features with correlation values above this threshold were removed. The outcome of this stage was a dataset with 28 selected features.

The next step was the feature importance technique, which aimed to select the most influential features for the model. This was conducted by first building a DT model from the 28

features obtained after the correlation filter. Then, the attribute feature_importance was extracted and ranked, as shown in Figure 2.

Figure 2 shows that some features had much higher feature importance values than others, indicating that they had a significant impact on the model. Conversely, several features had low or nearly zero values, indicating negligible influence on the model. This study then selected the top five features as the final result of the hybrid feature selection technique, as follows:

1. URG_count is the number of data packets with an urg flag in the same data flow;
2. IAT or inter-arrival time is the time difference between the arrival of two consecutive data packets;
3. Variance is the variance of the packet lengths entering and leaving within the same data flow;
4. Min is the minimum packet length in the same data flow; and
5. Header_Length is the header length, which is the initial part of a packet containing control information for data transmission and processing.

The selected features from the hybrid feature selection were then compared with several single feature selection techniques using the DT algorithm, as presented in Table IV. The table shows that the hybrid technique achieved the lowest computational time among all single techniques, i.e., 7.20 s, corresponding to the highest reduction rate of 87.19%. This indicated that the hybrid technique excelled in computational efficiency. However, its accuracy was not higher than some single techniques. This occurred because the number of selected features was fewer than in other techniques, which implied that some important features might not have been included. Nevertheless, these results demonstrated a trade-off between model effectiveness and efficiency: the more features used, the higher the accuracy, but also the longer the computational time. Conversely, feature reduction lowered computational time but could reduce accuracy. Therefore, the hybrid feature selection technique needed to be combined with HPO to restore accuracy levels.

Table IV
Performance Comparison Between Feature Selection Methods

Method	Number of Features	Accuracy (%)	Computation Time (s)	Time Reduction (%)
All Features	43	99.56	56.20	-
Correlation Filter	28	99.55	31.16	44.56
Mutual Information	10	99.36	13.27	76.39
Chi-Square	10	98.68	14.89	73.51
Recursive Feature Elimination	10	99.52	18.50	67.08
Feature Importance	5	99.46	11.19	80.09
Hybrid Method	5	99.37	7.20	87.19

Table V
Best Parameter Values of DT and RF Algorithms

Parameter	Value Range	DT Best Value	RF Best Value
criterion	gini, entropy	gini	entropy
max_depth	2 – 10	10	10
min_samples_split	2 – 5	2	2
min_samples_leaf	2 – 5	2	2
n_estimators	10 – 100	-	28

B. ATTACK DETECTION WITH BAYESIAN OPTIMIZATION

The ML-based cyber-attack detection model in this study was built using algorithms that are both robust and lightweight for IoT networks, namely DT and RF. Model performance had first been improved by reducing data dimensionality through the hybrid feature selection technique, which chose the five most informative features (URG_count, IAT, Variance, Min, and Header_Length). The next enhancement was achieved using Bayesian optimization to identify the optimal parameter values for the employed algorithms. This was implemented with the scikit-optimize (skopt) library on the parameter search space shown in Table III. The objective function for the optimization was accuracy, with five-fold cross-validation. The optimal values were then extracted using the best_params attribute, as displayed in Table V.

Table V presents the optimal parameter values obtained from Bayesian optimization for the DT and RF algorithms. For DT, the best parameter value for the criterion was gini, while for RF, the best value was entropy. Meanwhile, the parameters max_depth, min_samples_split, and min_samples_leaf had identical optimal values in both algorithms. Finally, the optimal number of decision trees in RF was 28. The models were then evaluated by comparing performance between those without HPO and those optimized with grid search, random search, and Bayesian optimization.

Table VI shows that all models achieved high performance, with accuracy, precision, recall, and F1 exceeding 99%. For the DT algorithm, hybrid feature selection and all HPO techniques improved performance from 99.56% to 99.64% while reducing computation time. Grid search and Bayesian optimization significantly reduced computation time to 3.50 s and 3.59 s, or reductions of 93.77% and 93.61%, respectively. For RF, hybrid feature selection reduced computation time from 569.60 s to 330.40 s (41.99%) while maintaining high accuracy, precision, recall, and F1 of 99.74%. However, applying HPO reduced accuracy slightly to 99.63% across all methods. Despite this decrease, HPO substantially reduced computation time—from 569.60 s to 15.49 s (97.28%) with GS, 156.33 s (72.55%) with random search, and 14.73 s (97.41%) with Bayesian optimization.

When compared with previous studies, Table VII shows that the proposed model achieved the highest accuracy, precision, recall, and F1, along with the lowest computation time. This indicates that the model offered superior effectiveness and efficiency, making it applicable to IoT networks. The selected features and optimized parameters were then evaluated in the detection of seven types of attacks, as shown in Figure 3 and Figure 4.

Figure 3 presents the evaluation of the DT algorithm using the selected features and parameters to detect attack types in IoT networks. The graph indicates that the highest precision, 100%, was achieved for brute force, DDoS, DoS, Mirai, and Web attacks, showing that the model effectively identified these attacks without false positives. This occurred because these attack types exhibited unique data patterns with minimal similarity to other attacks. The highest recall, 100%, was

TABLE VI
PERFORMANCE COMPARISON BETWEEN MODELS WITH AND WITHOUT HYPERPARAMETER METHOD

Model	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)	Computation Time (s)	Time Reduction (%)
DT with all features	99.56	99.56	99.56	99.56	56.20	-
DT with only FS	99.37	99.37	99.37	99.37	7.20	87.19
DT with FS and GS	99.64	99.64	99.64	99.64	3.50	93.77
DT with FS and RS	99.64	99.64	99.64	99.64	4.91	91.26
DT with FS and OB	99.64	99.64	99.64	99.64	3.59	93.61
RF with all features	99.74	99.74	99.74	99.74	569.60	-
RF with only FS	99.74	99.74	99.74	99.74	330.40	41.99
RF with FS and GS	99.63	99.63	99.63	99.63	15.49	97.28
RF with FS and RS	99.64	99.64	99.64	99.64	156.33	72.55
RF with FS and OB	99.63	99.64	99.63	99.63	14.73	97.41

FS: feature selection; GS: grid search; RS: random search; OB: Bayesian optimization

TABLE VII
PERFORMANCE COMPARISON BETWEEN PREVIOUS STUDIES

Reference	Model	Best Value
[11]	Model ML (Logistic Regression, Perceptron, Adaboost, RF, and DNN)	Accuracy: 99,68% Precision: 96,52% Recall: 96,54% F1: 96,53%
[12]	Model DL (DNN, CNN, and LSTM)	Accuracy: 99,40% Precision: 99,43% Recall: 99,40% F1: 99,41% Computation Time: 625 s
[13]	Model Hybrid ML (DT-RF-GB) with feature importance as the feature selection method	Accuracy: 99,51% Precision: 98,51% Recall: 99,63% F1: 99,07% Computation Time: 452 s
[21]	Model Hybrid DL (DNN-BiLSTM) with feature importance and Optuna	Accuracy: 93,13% Precision: 91,80% Recall: 93,13% F1: 91,94% Computation Time: 714,8 s
This Study	Model DT with hybrid feature selection and Bayesian optimization	Accuracy: 99,64% Precision: 99,64% Recall: 99,64% F1: 99,64% Computation Time: 3,59 s

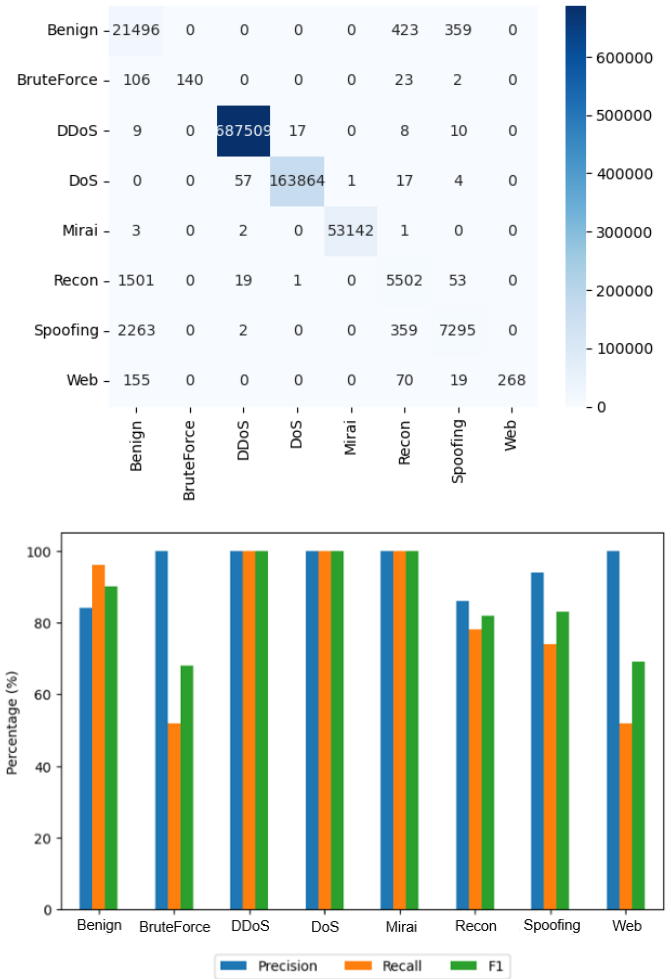

Figure 3. Evaluation of model performance using the decision tree algorithm.

obtained for DDoS, DoS, and Mirai attacks, indicating that the model was highly sensitive in identifying these attacks without false negatives. Finally, the highest F1 score, 100%, was also achieved for DDoS, DoS, and Mirai, indicating perfect balance in detection. These results confirmed that the DT-based model could effectively detect various cyber-attacks in IoT networks.

Figure 4 presents the evaluation of the RF algorithm using the selected features and parameters to detect attack types in IoT networks. The graph indicates that the highest precision, 100%, was achieved for brute force, DDoS, DoS, Mirai, and Web attacks, showing that the model effectively identified these attacks without false positives. This also occurred because these attack types exhibited unique data patterns with minimal similarity to other attacks. Similarly, the highest recall, 100%, was achieved for DDoS, DoS, and Mirai, indicating that the model was highly sensitive in identifying these attacks without false negatives. Finally, the highest F1 score, 100%, was also achieved for DDoS, DoS, and Mirai, demonstrating perfect balance in detection. These findings confirmed that the RF-based model could also effectively detect various cyber-attacks in IoT networks.

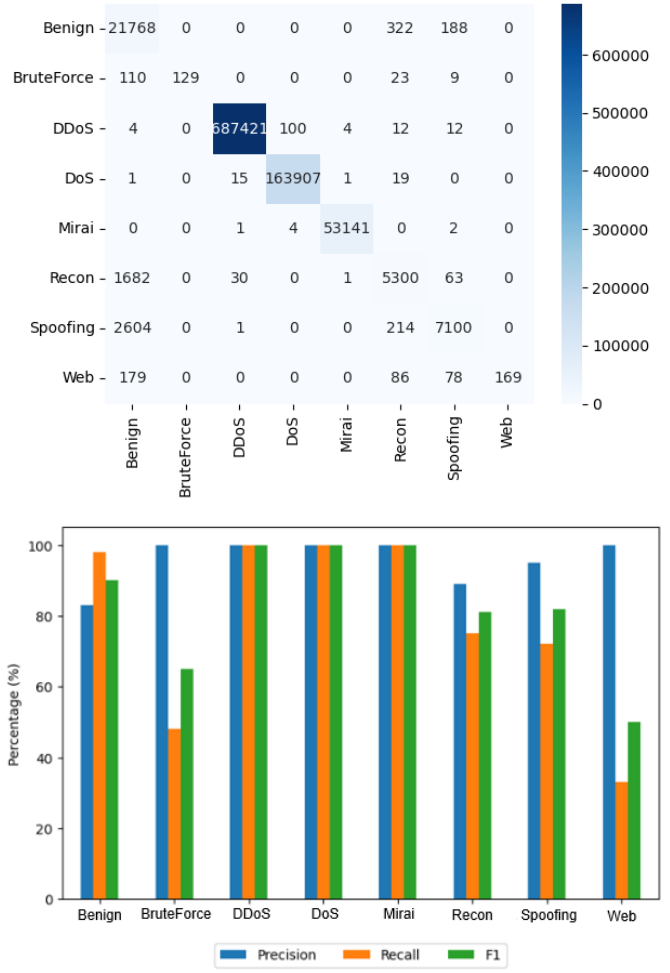

Figure 4. Evaluation of Model Performance Using the Random Forest Algorithm.

V. CONCLUSION

ML-based cyberattack detection models constitute the best alternative to address the risks of cyberattacks in the rapidly growing IoT networks. The use of feature selection to reduce data dimensionality and HPO to identify the optimal values of ML algorithm parameters is required to improve model performance. This study proposes a hybrid feature selection technique that combines correlation filtering and feature importance. In addition, Bayesian optimization was employed to determine the optimal values of the parameters of the ML algorithms used, namely DT and RF. The most recent and validated IoT cyberattack dataset, CICIoT2023, was utilized to evaluate the model. The results show that the hybrid feature selection technique offers superiority in terms of computation time, achieving the lowest among all single techniques at 7.20 s, along with the highest reduction rate of 87.19%, by selecting the five most relevant features to the model: URG_count, IAT, Variance, Min, and Header_Length. Furthermore, combining the model with HPO using Bayesian optimization significantly improved model performance, reaching an accuracy of 99.64% with a computation time of 3.59 s for DT, and an accuracy of 99.63% with a computation time of 14.73 s for RF. Therefore, the proposed ML-based cyberattack detection model may serve as a reference for the implementation of cybersecurity in IoT networks.

AUTHORS' CONTRIBUTION

Conceptualization, Samsudiat; methodology, Samsudiat; software, Samsudiat; validation, Kalamullah Ramli; formal analysis, Samsudiat; resources, Samsudiat; data curation, Samsudiat; writing—drafting, Samsudiat; writing—review and editing, Samsudiat and Kalamullah Ramli; visualization, Samsudiat; supervision, Kalamullah Ramli; project administration, Samsudiat; funding acquisition, Samsudiat.

ACKNOWLEDGMENT

This research received funding support from the Scholarship Program of the Ministry of Communication and Digital Affairs and infrastructure support from the National Research and Innovation Agency.

REFERENCES

[1] I.M.A. Alonso, "IoT cybersecurity: Protecting the merging of the physical and digital world," Telefónica. Access date: 26-Des-2024. [Online]. Available: https://www.telefonica.com/en/communication-room/blog/iot-cybersecurity-protecting-the-merging-of-the-physical-and-digital-world/

[2] S. Haque, F. El-Moussa, M. Komninos, and R. Muttukrishnan, "A systematic review of data-driven attack detection trends in IoT," *Sensors,* vol. 23, no. 16, pp. 1–29, Aug. 2023, doi: 10.3390/s23167191.

[3] K. Shafique *et al.*, "Internet of things (IoT) for next-generation smart systems: A review of current challenges, future trends, and prospects for emerging 5G-IoT scenarios," *IEEE Access*, vol. 8, pp. 23022–23040, Jan. 2020, doi: 10.1109/ACCESS.2020.2970118.

[4] "Lanskap Keamanan Siber Indonesia 2024," National Cyber and Crypto Agency, 2025.

[5] R. Mahmoud, T. Yousuf, F. Aloul, and I. Zualkernan, "Internet of things (IoT) security: Current status, challenges, and prospective measures," dalam *2015 10th Int. Conf. Internet Technol. Secur. Trans. (ICITST)*, 2015, pp. 336–341, doi: 10.1109/ICITST.2015.7412116.

[6] S. Yaras and M. Dener, "IoT-based intrusion detection system using new hybrid deep learning algorithm," *Electronics,* vol. 13, no. 6, pp. 1–28, Mar. 2024, doi: 10.3390/electronics13061053.

[7] M.A. Al-Garadi *et al.*, "A survey of machine and deep learning methods for Internet of things (IoT) security," *IEEE Commun. Surv. Tutor.*, vol. 22, no. 3, pp. 1646–1685, Apr. 2020, doi: 10.1109/COMST.2020.2988293.

[8] N. Mishra and S. Pandya, "Internet of things applications, security challenges, attacks, intrusion detection, and future visions: A Systematic review," *IEEE Access*, vol. 9, pp. 59353–59377, Apr. 2021, doi: 10.1109/ACCESS.2021.3073408.

[9] G.T. Reddy *et al.*, "Analysis of dimensionality reduction techniques on big data," *IEEE Access*, vol. 8, pp. 54776–54788, Mar. 2020, doi: 10.1109/ACCESS.2020.2980942.

[10] P. Sahu *et al.*, "Enhancing industrial IoT intrusion detection with hyperparameter optimization," dalam *2024 15th Int. Conf. Comput. Commun. Netw. Technol. (ICCCNT)*, 2024, pp. 1–6, doi: 10.1109/ICCCNT61001.2024.10723326.

[11] E.C.P. Neto *et al.*, "CICIoT2023: A real-time dataset and benchmark for large-scale attacks in IoT environment," *Sensors,* vol. 23, no. 13, pp. 1–26, Jul. 2023, doi: 10.3390/s23135941.

[12] F.L. Becerra-Suarez, V.A. Tuesta-Monteza, H.I. Mejia-Cabrera, and J. Arcila-Diaz, "Performance evaluation of deep learning models for classifying cybersecurity attacks in IoT networks," *Informatics*, vol. 11, no. 2, pp. 1–13, Jun. 2024, doi: 10.3390/informatics11020032.

[13] T.-T.-H. Le *et al.*, "Toward enhanced attack detection and explanation in intrusion detection system-based IoT environment data," *IEEE Access*, vol. 11, pp. 131661–131676, Nov. 2023, doi: 10.1109/ACCESS.2023.3336678.

[14] B. Susilo, A. Muis, and R.F. Sari, "Intelligent intrusion detection system against various attacks based on a hybrid deep learning algorithm," *Sensors*, vol. 25, no. 2, pp. 1–26, Jan. 2025, doi: 10.3390/s25020580.

[15] Q.R.S. Fitni and K. Ramli, "Implementation of ensemble learning and feature selection for performance improvements in anomaly-based intrusion detection systems," dalam *2020 IEEE Int. Conf. Ind. 4.0 Artif. Intell. Commun. Technol. (IAICT)*, 2020, pp. 118–124, doi: 10.1109/IAICT50021.2020.9172014.

[16] W. Lian *et al.*, "An intrusion detection method based on decision tree-recursive feature elimination in ensemble learning," *Math. Probl. Eng.*, vol. 2020, no. 1, pp. 1–15, Nov. 2020, doi: 10.1155/2020/2835023.

[17] A.A. Megantara and T. Ahmad, "Feature importance ranking for increasing performance of intrusion detection system," dalam *2020 3rd Int. Conf. Comput. Inform. Eng. (IC2IE)*, 2020, pp. 37–42, doi: 10.1109/IC2IE50715.2020.9274570.

[18] H. Kurniawan *et al.*, "Enhancing the detection of botnet attacks in the Internet of things networks through the utilization of hybrid feature selection," dalam *2024 FORTEI-Int. Conf. Electr. Eng. (FORTEI-ICEE)*, 2024, pp. 89–94, doi: 10.1109/FORTEI-ICEE64706.2024.10824638.

[19] J.J. Shirley and M. Priya, "Hybrid MRMR-PCA BagDT – An effective feature selection based ensemble model for real-time intrusion detection in IoT environment," *IEEE Access*, vol. 12, pp. 144230–144248, Sep. 2024, doi: 10.1109/ACCESS.2024.3468897.

[20] J.-B. Altidor and C. Talhi, "Enhancing port scan and DDoS attack detection using genetic and machine learning algorithms," in *2024 7th Conf. Cloud Internet Things (CIoT)*, 2024, pp. 1–7, doi: 10.1109/CioT63799.2024.10757005.

[21] Y.N. Kunang, S. Nurmaini, D. Stiawan, and B.Y. Suprapto, "Improving classification attacks in IoT intrusion detection system using Bayesian hyperparameter optimization," in *2020 3rd Int. Semin. Res. Inf. Technol. Intell. Syst. (ISRITI)*, 2020, pp. 146–151, doi: 10.1109/ISRITI51436.2020.9315360.

[22] K. Ashton, "That 'Internet of things' thing." RFID JOURNAL. Access date: 26-Des-2024. [Online]. Available: https://www.rfidjournal.com/expert-views/that-internet-of-things-thing/73881

[23] L. Chettri and R. Bera, "A comprehensive survey on Internet of things (IoT) toward 5G wireless systems," *IEEE Internet Things J.*, vol. 7, no. 1, pp. 16–32, Jan. 2020, doi: 10.1109/JIOT.2019.2948888.

[24] S. Dange and M. Chatterjee, "IoT botnet: The largest threat to the IoT network," in *Data Commun. Netw., Proc. GUCON 2019*, 2019, pp. 137–157, doi: 10.1007/978-981-15-0132-6_10.

[25] S. Yamaguchi, "Botnet defense system: Concept, design, and basic strategy," *Information*, vol. 11, no. 11, pp. 1–15, Nov. 2020, doi: 10.3390/info11110516.

[26] Y. Lu and L.D. Xu, "Internet of things (IoT) cybersecurity research: A review of current research topics," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2103–2115, Apr. 2019, doi: 10.1109/JIOT.2018.2869847.

[27] Y. Xin *et al.*, "Machine learning and deep learning methods for cybersecurity," *IEEE Access*, vol. 6, pp. 35365–35381, May 2018, doi: 10.1109/ACCESS.2018.2836950.

[28] H.-J. Liao, C.-H.R. Lin, Y.-C. Lin, and K.-Y. Tung, "Intrusion detection system: A comprehensive review," *J. Netw. Comput. Appl.*, vol. 36, no. 1, pp. 16-24, Jan. 2013, doi: 10.1016/j.jnca.2012.09.004.

[29] F. Hussain, R. Hussain, S.A. Hassan, and E. Hossain, "Machine learning in IoT security: Current solutions and future challenges", *IEEE Commun. Surv. Tutor.,* vol. 22, no. 3, pp. 1686–1721, Apr. 2020, doi: 10.1109/COMST.2020.2986444.

[30] L. Breiman, J. Friedman, R.A. Olshen, and C.J. Stone, *Classification and Regression Trees.* New York, NY, AS: Chapman & Hall/CRC, 2017.

[31] B. Mahbooba, M. Timilsina, R. Sahal, and M. Serrano, "Explainable artificial intelligence (XAI) to enhance trust management in intrusion detection systems using decision tree model," *Complexity,* vol. 2021, no. 1, pp. 1–11, Jan. 2021, doi: 10.1155/2021/6634811.

[32] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, pp. 5-32, Oct. 2001, doi: 10.1023/A:1010933404324.

[33] A.K. Balyan *et al.*, "A hybrid intrusion detection model using EGA-PSO and improved random forest method," *Sensors,* vol. 22, no. 16, pp. 1–20, Aug. 2022, doi: 10.3390/s22165986.

[34] Canadian Institute for Cybersecurity (CIC), 2023, "CIC IoT Dataset 2023", Canadian Institute for Cybersecurity (CIC), University of New Brunswick (UNB), Canada. [Online]. Available: https://www.unb.ca/cic/datasets/iotdataset-2023.html

[35] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157-1182, Mar. 2003, doi: 10.1162/153244303322753616.

[36] D. Edelmann, T.F. Móri, and G.J. Székely, "On relationships between the Pearson and the distance correlation coefficients," *Stat. Probab. Lett.*, vol. 169, pp. 1–6, Feb. 2021, doi: 10.1016/j.spl.2020.108960.

[37] I.H. Sarker, Y.B. Abushark, F. Alsolami, and A.I. Khan, "IntruDTree: A machine learning based cyber security intrusion detection model," *Symmetry,* vol. 12, no. 5, pp. 1–15, May 2020, doi: 10.3390/sym12050754.

[38] B. Bischl *et al*., "Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges," *WIREs Data Min. Knowl. Discov.*, vol. 13, no. 2, pp. 1–43, Mar./Apr. 2023, doi: 10.1002/widm.1484.

[39] R. Zuech, J. Hancock, and T.M. Khoshgoftaar, "Detecting web attacks using random undersampling and ensemble learners," *J. Big Data*, vol. 8, no. 1, pp. 1–20, May 2021, doi: 10.1186/s4053-021-00460-8.

[40] M.A. Umar, Z. Chen, K. Shuaib, and Y. Liu, "Effects of feature selection and normalization on network intrusion detection," *Data Sci. Manag.*, vol. 8, no. 1, pp. 23-39, Mar. 2025, doi: 10.1016/j.dsm.2024.08.001.

This page is intentionally left blank