Accredited SINTA 2 Ranking
Decree of the Director General of Higher Education, Research, and Technology, No. 158/E/KPT/2021
Validity period from Volume 5 Number 2 of 2021 to Volume 10 Number 1 of 2026

Published online at: **http://jurnal.iaii.or.id**

JURNAL RESTI
(Rekayasa Sistem dan Teknologi Informasi)
Vol. 9 No. 2 (2025) 425 - 434 e-ISSN: 2580-0760

Classification Model for Bot-IoT Attack Detection Using Correlation and Analysis of Variance

Firgiawan Faira[1*], Dandy Pramana Hostiadi[2], Roy Rudolf Huizen[3]
[1]Magister Program, Department of Magister Information System, Institut Teknologi dan Bisnis STIKOM Bali, Denpasar, Indonesia
[2,3]Department of Magister Information System, Institut Teknologi dan Bisnis STIKOM Bali, Denpasar, Indonesia
[1]232011017@stikom-bali.ac.id, [2]dandy@stikom-bali.ac.id, [3]roy@stikom-bali.ac.id

Abstract

Industry 4.0 requires secure networks as the advancements in IoT and AI exacerbate the challenges and vulnerabilities in data security. This research focuses on detecting Bot-IoT activity used the dataset Bot-IoT UNSW Canberra 2018. Bot-IoT dataset initially showed a significant imbalance, with 2,934,447 entries of attack activity and only 370 entries of normal activity. To address this imbalance, an innovative data aggregation technique was applied, effectively reducing similar patterns and trends. This approach resulted in a balanced dataset consisting of 8 attack activity points and 80 normal activity points. Feature selection using the ANOVA method identified 10 key features from a total of 17. The classification process utilized Random Forest (RF), k-Nearest Neighbors (kNN), Naïve Bayes (NB), and Decision Tree (DT) algorithms, with 100 iterations and an 80:20 training-testing split. Random Forest showed superior performance, achieving 97.5% accuracy, 97.4% precision, and 97.4% recall, with a total computation time of 11.54 seconds. N IN Conn P DstIP and seq had the highest positive correlation value (+0.937) according to Pearson correlation analysis, whereas N IN Conn P SrcIP and state number had the lowest negative correlation value (-0.224). This research focuses on the implementation of a data aggregation strategy to address class imbalance, greatly enhancing machine learning model performance and optimizing training time, is what makes this research distinctive. These results aid in the creation of strong cybersecurity systems that can identify dangers associated with the Internet of Things.

Keywords: Aggregate Data; ANOVA; Bot-IoT; Pearson Correlation; Classification

How to Cite: Firgiawan Faira and Dandy Pramana Hostiadi, "Classification Model for Bot-IoT Attack Detection Using Correlation and Analysis of Variance", *J. RESTI (Rekayasa Sist. Teknol. Inf.)* , vol. 9, no. 2, pp. 425 - 434, Apr. 2025.
DOI: https://doi.org/10.29207/resti.v9i2.6332

1. Introduction

In this 4.0 industry and 5.0 Society era, characterized by advancements in network technologies and the IoT, demands robust network security systems. This era connects and innovates industrial and societal life through disruptive technologies like IoT, cloud computing, and artificial intelligence[1]. However, as the demand for information and communication technology systems grows, so do critical challenges relate to data security. Vulnerabilities enable malicious users and programs to infiltrate systems, often resulting in data theft or system damage[2]. The rapid advancements in IoT technology have led to an increase in cyberattacks targeting IoT systems, aiming to harm targeted entities[3][4]. Examples of network security attacks include malicious threats commonly known as malware, especially botnet attacks[5][6]. Because botnets can inject DDoS, DoS, SPAM, phishing, and identity or personal data theft, they pose one of the most dangerous risks among these virus types[7][8]. Three main components enable a botnet to operate: the botmaster, the CnC (command & control), and the bots. The CnC (command & control) server serves as the central server used for recording or controlling infected computers, while the botmaster acts as the controller of the CnC (command & control) server[9]. These components enable hackers to control botnets and launch various attacks on networks. Despite the deployment of an Intrusion Detection System (IDS), a fundamental element of cybersecurity and information security, cyberattacks can still infiltrate networks and IoT devices[10].

The strategic security system becomes an attractive target for attackers or hackers who deliberately attempt

Received: 27-01-2025 | Accepted: 12-04-2025 | Published Online: 22-04-2025

to infiltrate with malicious intent, causing harm by injecting bots into IoT devices through the network[11].

Supported by several previous studies analyzing Bot-IoT activity using feature selection with classification models, one of which is Kerrakchou et al.[12] their study used the Bot-IoT UNSW Canberra 2018 dataset to find the most effective machine learning method for a classification. They compared six classification algorithms. The results revealed that, when applying a feature set of nine features, the Random Forest method performed best across four distinct measures. Hostiadi *et al*.[13] Eight features were identified with a 97.35% accuracy rate using the CTU-13 dataset and the cosine similarity approach based on feature selection for classification analysis. Halim *et al.* [14] This study introduced an enhanced feature selection method utilizing a genetic algorithm and evaluated its performance on three comparative network traffic datasets: the Bot-IoT dataset, UNSW-NB15, and CIRA-CIC-DOHBrw2020. Standard feature selection techniques were also compared. The results showed using GbFS increased accuracy, reaching a high of 99.80%. Liu *et al*.[15] Three classification analysis techniques were used with the UNSW Canberra Botnet Activity dataset, which has 43 features total 29 original features + 14 calculated features using k-NN, Random Forest, and Decision Tree. Feature selection identified six features, achieving an optimal accuracy of 99.98%.

The analysis of Bot-IoT detection using an ANOVA feature selection optimization technique is the main emphasis of this research, since it provides a statistically valid and legitimate way for comparing and grading characteristics in the Bot-IoT, Considering the context and literature review. This approach is selected because it facilitates efficient feature selection to identify the most pertinent characteristics by highlighting important interactions between dependent and independent variables within the existing features. As a reinforcement analysis in the feature selection process to attain the best classification accuracy in machine learning, feature correlation analysis is also used to validate and evaluate the correlations between features.

Additionally, the research uses Classification Modeling to determine how well machine learning models perform. The research uses the Bot-IoT UNSW Canberra 2018 dataset and applies data aggregation processing on daddr (destination IP address), which is novel in that it addresses data imbalance. This is required because, during the data preparation phase, particular activity records from the 2,934,818 training data records indicate similar patterns and trends. This is supported by the fact that the Bot-IoT attack type is characterized by DoS (Denial of Service), which floods the network with traffic, and Reconnaissance, which aims to gather as much information as possible from the target. Data aggregation is therefore seen to be crucial during the data preparation stage in order to minimize the size of the enormous dataset and speed up the computation process.

2. Research Methods

The first steps of this research is collected data at the UNSW Canberra Bot-IoT 2018 dataset, which is published via the UNSW Canberra website. The next step is data preparation, where characteristic analysis is conducted on the dataset. Given that many attack records have similar values, a new treatment is applied by performing data aggregation to reduce the dataset size. Following this, data preprocessing is carried out, including feature selection using the ANOVA (Analysis of Variance) method to determine the values and select the best features. Correlation analysis between features is then conducted using the Pearson correlation method. Machine learning classification models for attack and normal classes (binary data) are then applied using algorithms such as Random Forest, kNN, Decision Tree, and Naïve Bayes. Finally, after testing, the methods are assessed to ascertain the results of the identification of the machine learning model.

2.1 Bot-Iot Detection Model

Detecting Bot-IoT activity is the main goal of the research flow, it selects the best features for the classification model using ANOVA feature selection optimization, in addition to correlation analysis to validate and enhance the information obtained through the feature selection technique by analyzing the relationships between features.

Figure 1 displays the flowchart for the research process.

2.2 Data Preparation

Data composition and data aggregation comprise data preparation. Table 1 below shows the data composition of the UNSW Canberra 2018 Bot-IoT dataset[16], shown in Table 1.

Table 1. Total data for the attack class

No.	Attack Class	# Samples
1	Attack Activity	2,934,447
2	Normal Activity	370
Total		2,934,817

The 17 attributes shown in Table 2 that are connected to the UNSW Canberra 2018 Bot-IoT dataset are described in Table 2.

With a total of 2,934,817 activities, the attack feature in the Bot-IoT dataset pattern is divided into two classes: label 0 for non-attack activity (normal) and label 1 for attack activity (attack). Upon further investigation, it was found that most of the activity records in the Bot-IoT dataset have similar patterns and trends. As a result, a data aggregation process was performed with the goal of gaining better insights into the data characteristics, conducting pattern and trend identification analysis, and reducing data with similar patterns and trends. This would help optimize the computational process in machine learning. During the data preparation

observation phase, it was discovered that the Bot activity records in the dataset exhibited similar patterns and trends. Therefore, this research introduces novelty through the analysis of data aggregation.

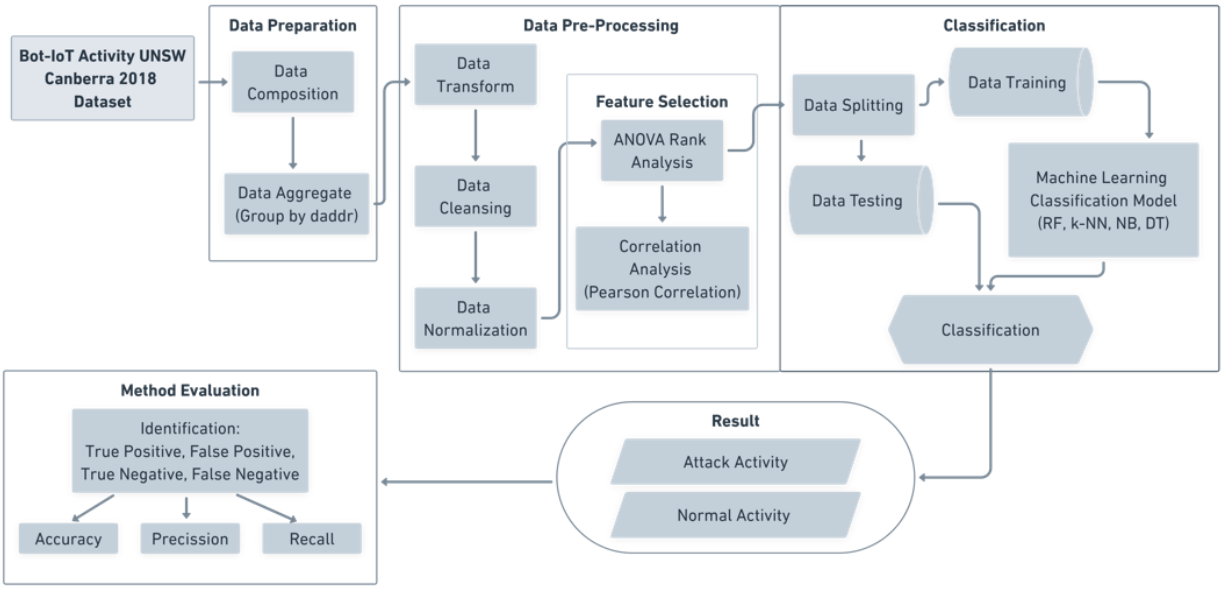

Figure 1. Bot-IoT Detection Model

Table 2. Explanation of the Features

No.	Feature	Description
1	Proto	Textual depiction of the network flow's transaction protocols
2	Saddr	The IP address of the source.
3	pkSeqID	Row Identifier
4	Sport	Port number of the source
5	Daddr	The destination's IP address
6	Dport	Destination port number
7	Seq	Argus sequence number
8	Stddev	The aggregate of the records' standard deviation
9	N IN Conn P SrcIP	number of connections coming in from each IP source.
10	N IN Conn P DstIP	number of connections coming in from each IP destination.
11	Min	Minimum time frame for aggregated records
12	Max	Maximum time frame for aggregated records
13	Mean	Average length of time for all records aggregated
14	state number	Numerical representation of feature state
15	Srate	Packets per second from the origin to the final location
16	Drate	From the destination to the source, packets per second
17	Attack	Normal traffic has a class label of 0, and attack traffic has a label of 1.

2.3 Data Pre-processing

Before raw data is entered into a machine learning model or algorithm, a number of procedures known as data preparation are conducted[17]. The objective is to prepare the data so that the model can handle it efficiently, improve its quality, and produce better analysis or prediction results[18].

The common steps involved in data preprocessing include transformation, cleaning, and normalization. The process of transformation involves transforming the data into a format better suited for modeling or analysis. This may include normalization, standardization, or other techniques such as logarithmic or square root transformations. Cleaning focuses on handling missing, incomplete, or invalid data, which may involve filling in missing values, removing invalid entries, or addressing outliers. Lastly, normalization helps to increase the accuracy and performance of analytical models by ensuring that all variables have a consistent range of values.

2.4 Selection Feature With ANOVA

A statistical technique called ANOVA (Analysis of Variance) is frequently used for feature selection in order to determine which features are most relevant to the prediction of the target variable[19]. When selecting features for machine learning models, the connection between each attribute and the target variable may be evaluated using ANOVA. This makes it possible to choose traits that will have the most influence[20].

The basic steps for using ANOVA in feature selection involve several key stages. First, the dataset is divided based on the categories of the target variable. Then, the variance is calculated both between and within each category. Finally, these variances are compared to determine whether the differences between categories are statistically significant, helping to identify which features have a meaningful impact on the target variable.

Features with significant p-values (typically less than 0.05) are chosen as relevant features when the feature selection procedure is completed. Every characteristic is assessed and given a ranking. The F-value is the result of the ANOVA computation. The labels are more disjointed when the F-value is larger. As in Formula 1, the distance between classes is used to calculate each feature's score.

$$\sigma_{cl}^2 = \frac{\sum(\bar{x}o - \bar{x})^2 a_i}{(k-l)} \quad (1)$$

The class o mean is denoted by $\bar{x}o$, the overall feature mean by $\bar{x}$, and the quantity of class instances o in the set by a_o. After that, the distance between classes is determined using Formula 2.

$$\sigma_{err}^2 = \frac{(\sum\sum(x_{op} - \bar{x})^2) - (\bar{x}o - \bar{x})^2 a_o}{(k-l)} \quad (2)$$

The total squared values for the feature for each class, less the feature mean, is then subtracted from the sum of the class's squared means less the feature mean, as shown in Formula 3.

$$F = \frac{\sigma_{cl}^2}{\sigma_{err}^2} \quad (3)$$

The chosen percentile process will be determined using the $f_{\_value}$, which is the outcome of the ANOVA computation. For every feature, the ANOVA computation yields a set of $f_{\_value}$. k_j is a representation of the features in K. Furthermore, the ANOVA computation results are compiled into a vector denoted by Fval, which has values for every feature number from n as shown in Formula 4.

$$Fval = [kval_1, kval_2, kval_3, \dots, kval_n]; j = \{1,2,3, \dots, n\} \quad (4)$$

By reducing the data dimensions, removing superfluous features, and concentrating on features that have the most influence on the target variable, feature selection using ANOVA can enhance a machine learning model's performance.

2.5 Correlation Analysis with Pearson Correlation

The correlation analysis in this study applies the Pearson correlation. A statistical method for determining the direction and degree of a linear relationship between two variables is Pearson's correlation[21]. The coefficient of Pearson correlation, additionally referred to as the Pearson Correlation Coefficient (r), has a value between -1 and 1. The formula for determining the Pearson correlation coefficient, which is shown in Formula 5

$$r = \frac{n(\Sigma xy) - (\Sigma x)(\Sigma y)}{\sqrt{[n\Sigma x^2 - (\Sigma x)^2]\,[n\Sigma y^2 - (\Sigma y)^2]}} \quad (5)$$

(x), (y) are the variables under comparison, and n is the number of data pairs.

2.6 Machine Learning Classification

In this study, Botnet behaviors are analyzed using machine learning techniques that utilize Random Forest, kNN, Decision Tree, and Naive Bayes classification models.

A supervised classification model called Random Forest is built using many decision trees. This technique is frequently applied to low-processing-power regression and classification tasks[22]. Based on the growth of trees put together by previously generated random vectors during the training process, popular feature classes are chosen. As an ensemble methodology, Random Forest is regarded as a group learning approach for regression and element classification. To understand irregular patterns, deep trees are utilized[23]. Next, as shown in Formula 6, the corresponding training set trees for these data are replaced.

for x = 1 …, X:

$$\frac{1}{x}\sum_{x=1}^{x} fx\,(R) \quad (6)$$

The fundamentals of Random Forest use include bagging (bootstrap aggregating) and the construction of numerous decision trees using random subsets of the training data. This method aids in lowering variance and enhancing the model's stability.

Lazy learning is used in the non-parametric machine learning technique known as kNN (k-Nearest Neighbor). A non-parametric approach is one that doesn't assume anything about the distribution of the underlying data[17][24]. In other words, there is no fixed number of parameters or parameter estimates in the model, regardless of whether the data is small or large. The process can be comprehensively described in several steps. First, determine the value of *k*. When choosing *k*, it is important to consider the training dataset with the smallest distance. The optimal value of *k* is best determined using cross-validation. Then, compute the distance of type *k* with the new object.

After that, choose the k nearest neighbors' labels. Lastly, give the new object the most common label based on the *k* labels that were chosen.

One feature of kNN is the use of a distance metric to quantify the "closeness" between new and existing data. [25].

An approach to data processing known as a decision tree builds a regression or classification model in the shape of a tree to forecast future events[26][27]. This is achieved by continuously splitting the data into smaller subsets, and gradually developing a decision tree in the process[28][29].

This technique results in a tree featuring decision nodes and leaf nodes. To measure and direct the dataset's splitting at each node, this structure is constructed using certain mathematical formulations

Entrophy is a key concept used to measures the level of uncertainty or impurity whitin a dataset as shown in Formula 7.

$$Entropy\ (S) = -\sum_{i=1}^{c} p_i\ log_2(p_i) \quad (7)$$

p_i represents the proportion of samples form class i in the dataset S. A higher entrophy value indicates grater disorder or impurity in the dataset.

The effectiveness of a particular characteristic in classifying the data is then assessed using Information Gain. After the dataset is divided according to that attribute, it calculates the decrease in entropy and is expressed as Formula 8.

$$Gain\ (S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_V|}{|S|} Entropy(S_v) \quad (8)$$

The subset of S that has the value v for feature A is denoted by S_v. For the split, the feature with the greatest information gain is selected.

Lastly, the Gini Index is another matric used measures impurity in a dataset, similar to entropy, but uses a different formula as shown in Formula 9.

$$Gini(S) = 1 - \sum_{i=1}^{c} {p_i}^2 \quad (9)$$

The percentage of class i samples in dataset S is denoted by p_i. When creating and refining decision trees for categorization tasks, a lower Gini Index is essential.

The theory of Bayes serves as the foundation for the Naïve Bayes family of classification methods. The functioning of this method is based on the conditional probability principle, which characterizes the likelihood of an event occurring given that a related event has previously occurred[24],[30]. The formula for conditional probability is shown in Formula 10.

$$P(B|A) = \frac{P(B|A)P(A)}{P(B)} \quad (10)$$

In data mining algorithms, Naïve Bayes is very helpful since it converts big datasets into insights that can be put to use[31], [27]. Among its many uses are facial recognition, which identifies facial features including the eyes, nose, mouth, and eyebrows; forecasting the weather; diagnosing illnesses; classifying news; classifying emails (such as spam); and many more. This versatility makes Naïve Bayes a powerful tool for extracting meaningful information from vast amounts of data.

2.7 Evaluation Matrix

The relationships in a confusion matrix are used to distinguish between real-world events and model predictions to estimate classification performance in machine learning. Formulas 11 through 14 list the formulas used for calculating F-measure, accuracy, precision, and recall[32], [33].

$$Accuracy\ = \frac{TP + TN}{TP + FP + TN + FN} \quad (11)$$

$$Precision\ = \frac{TP}{TP + FP} \quad (12)$$

$$Recall\ = \frac{TP}{TP + FN} \quad (13)$$

$$F1\ score\ = \frac{2 \times precision \times Recall}{precision + recall} \quad (14)$$

By evaluating a classification model's accuracy, prediction quality, and capacity to catch true positives, these metrics offer a thorough assessment of its performance.

3. Results and Discussions

A study of the features of the Bot-IoT dataset is the first step in the results of the study and discussion. The entire number of datasets appear in Table 3

Table 3. Total data for the attack class

No.	Attack Class	# Samples
1	Attack Activity	2,934,447
2	Normal Activity	370
Total		2,934,817

The analysis results show that the Bot-IoT dataset exhibits an imbalance between attack activities and normal activities, requiring specific treatment to balance the data. A more detailed analysis of the patterns and trends revealed many similarities in data characteristics, enabling the data aggregation process. This aggregation aims to reduce data volume and accelerate computational processes.

The data aggregation was performed using the Python programming language (library: pandas as pd) on the daddr (destination IP address) field, followed by class categorization into attack (malicious) and non-attack (normal) types.

The Bot-IoT dataset depicts a situation in which Kali Linux virtual machines (VMs) are used to carry out cyberattacks. This simulates an IoT botnet network within a virtual environment to study or test botnet behavior, cyberattacks, and defense mechanisms.

The system starts with a PF Sense Firewall (192.168.100.1), which regulates and filters data traffic between internal (LAN) and external (WAN) networks, ensuring that only authorized data can pass through. All virtual and physical devices are connected via a switch integrated into a VMware Cluster, a virtual environment that safely simulates IoT devices and attack targets. Additionally, a Packet Filter Firewall is used to filter data traffic before connections reach the internet, mimicking real-world conditions.

This dataset simulation involves key elements such as attacking machines (botnets), represented by the *saddr* (source IP address) feature, and target machines, represented by the *daddr* (destination IP address) feature. The botnet machines (192.168.100.150, 192.168.100.149, 192.168.100.148, and 192.168.100.147) are virtual machines simulating botnet-infected devices, using Kali Linux, for example, to launch attacks. The targets include an Ubuntu server (192.168.100.3) simulating an IoT server, Metasploitable (192.168.100.7) as a vulnerable system, Windows 7 (192.168.100.6) as a user device, Ubuntu Mobile (192.168.100.5) as an IoT device, and Ubuntu_Tap (192.168.100.4) for network traffic monitoring and logging.

Test the propagation of IoT botnets, carry out DDoS attacks on targets, and evaluate the performance of network defences. Additional pathways in the simulation are used to analyze network traffic and log activities occurring during the experiment.

A deeper analysis of Bot-IoT activity records reveals that most patterns and trends remain consistent without losing critical dataset information. This aligns with the characteristics of Denial of Service (DoS) attacks, which violate information availability by rendering systems unresponsive or crashing, such as through radio signal jamming or flooding the network with traffic. Similarly, reconnaissance attacks aim to gather as much information as possible from the target. The Bot-IoT dataset simulation captures a significant number of similar activities.

By grouping and aggregating data based on the *daddr* feature and classifying attack labels using the *attack* feature, two categories are defined: label 0 for non-attack (normal) activities and label 1 for attack activities. The dataset consists of 2,934,817 activity records. This research introduces a novelty by analysing data aggregation in the data preparation stage. Table 4 below shows the results of the data aggregation strategy.

Table 4. The results of data aggregation grouped by daddr

No.	Attack Class	# Samples
1	Attack Activity	8
2	Normal Activity	80
Total		88

After the data aggregation process to address the imbalance issue, the next step was data preprocessing, which included data transformation, data cleaning, and data normalization—an essential stage to prepare raw data for machine learning analysis. Data transformation was carried out to standardize the data, where categorical features such as *dport*, *proto*, *saddr*, *port*, and *daddr* were converted into numerical values using one-hot encoding, followed by feature merging to combine relevant features.

The 2.2% of missing values in the aggregated dataset were then handled via data cleaning, which involved replacing the missing values with the average or most frequent value of the corresponding characteristic in the Bot-IoT dataset.

Finally, data normalization was applied to standardize the feature scales and reduce bias, ensuring that all values fell within the interval [0,1].

In the next step, feature selection is done using ANOVA. Table 5 presents the findings of the ANOVA test.

The ANOVA results show that the highest value is for the feature N_IN_Conn_P_DstIP with a score of 220.548, followed by Max at 129.450, stddev at 124.615, Seq at 83.699, pkSeqID at 29.166, Sport at 13.963, N_IN_Conn_P_SrcIP at 10.759, Proto at 7.873, state_number at 6.064, Mean at 5.650, Min at 4.997, Saddr at 4.035, Dport at 3.549, Drate at 1.991, and the lowest values are for Daddr at 0.913 and Srate at 0.099. N_IN_Conn_P_DstIP, Max, and Stddev have a greater influence on the class separation in the dataset, while features with lower ANOVA values like daddr and srate contribute less to the classification process and may be considered for dimensionality reduction or feature elimination in the model.

Table 5. The ANOVA evaluation's findings on the characteristics.

Features	#ANOVA
N_IN_Conn_P_DstIP	220.548
Max	129.450
Stddev	124.615
Seq	83.699
pkSeqID	29.166
Sport	13.963
N_IN_Conn_P_SrcIP	10.759
Proto	7.873
state_number	6.064
Mean	5.650
Min	4.997
Saddr	4.035
Dport	3.549
Drate	1.991
Daddr	0.913
Srate	0.099

Using the Pearson correlation approach, correlation analysis is carried out following the acquisition of feature selection findings via ANOVA. A statistical method for determining the direction and degree of a linear connection between two variables is Pearson correlation. The Pearson correlation results as part of the analysis between the features are shown in Table 6.

Using Pearson correlation, 45 correlation analysis findings between characteristics were found. With a correlation value of +0.937, the link between the variables N IN Conn P Dest IP and seq has the greatest value, indicating a very strong positive relationship between them. The value of seq tends to rise in tandem with the number of N IN Conn P Dest IP. Conversely, the variables N IN Conn P Src IP and state number and N IN Conn P Dest IP and min had the lowest correlation values, both of which had a value of -0.224. This finding suggests a somewhat negative association, which means that mean or min values tend to decline as N IN Conn P Src IP or N IN Conn P Dest IP values rise.

Using the top ten features from earlier studies, this study will compare the feature selection outcomes[16].

Ten top features emerged from the best feature selection results. The next step involves building classification models using the following algorithms: Random Forest (number of trees = 50), k-Nearest Neighbors (k = 7), Decision Tree (minimum number of instances in leaves = 2), and Naïve Bayes. The results obtained from training the model with an 80:20 split and 100 iterations (in percentage), are displayed in Table 7.

Table 6. The result of the pearson correlation

Pearson Correlation Value	Variable 1	Variable 2
+0.937	N_IN_Conn_P_DstIP	seq
+0.913	drate	srate
+0.887	N_IN_Conn_P_DstIP	stddev
+0.831	seq	stddev
+0.767	max	mean
+0.728	max	stddev
+0.673	N_IN_Conn_P_DstIP	max
+0.526	max	seq
+0.479	mean	stddev
+0.349	N_IN_Conn_P_DstIP	N_IN_Conn_P_SrcIP
+0.348	N_IN_Conn_P_DstIP	mean
+0.322	N_IN_Conn_P_SrcIP	stddev
+0.301	N_IN_Conn_P_SrcIP	seq
+0.293	mean	seq
+0.243	max	state_number
-0.224	N_IN_Conn_P_SrcIP	state_number
-0.224	N_IN_Conn_P_DstIP	min
+0.212	N_IN_Conn_P_DstIP	state_number
+0.197	drate	max
+0.186	mean	min
-0.183	min	seq
-0.167	min	stddev
+0.167	seq	state_number
+0.134	N_IN_Conn_P_SrcIP	max
+0.122	N_IN_Conn_P_SrcIP	min
-0.100	N_IN_Conn_P_SrcIP	drate
+0.096	state_number	stddev
+0.095	N_IN_Conn_P_DstIP	drate
+0.094	min	state_number
-0.090	N_IN_Conn_P_SrcIP	srate
+0.090	drate	mean
-0.076	min	srate
+0.067	mean	state_number
+0.063	drate	state_number
-0.049	srate	state_number
-0.048	mean	srate
-0.044	srate	stddev
-0.042	max	min
+0.042	N_IN_Conn_P_DstIP	srate
+0.035	drate	stddev
-0.034	max	srate
+0.016	seq	srate
+0.014	drate	min
+0.004	drate	seq
-0.004	N_IN_Conn_P_SrcIP	mean

The results obtained from the classification models using 100 iterations are as follows:

For Random Forest, AUC = 0.994, indicating excellent performance in class differentiation, with CA = 0.975, showing 97.5% accuracy in predictions. With an F1 score of 0.974, precision and recall are well balanced, while precision and recall are both 0.974 and 0.975, respectively, showing how well the model identified the positive class. A significant correlation between forecasts and actual values is indicated by the MCC score of 0.784.

In the case of k-Nearest Neighbors (k = 7), AUC = 0.991 demonstrates strong class differentiation, with CA = 0.964, reflecting 96.4% correct predictions. The F1 score is 0.962, With both precision and recall at 0.964, demonstrating a solid balance between the two, this shows strong accuracy in detecting positive groups. MCC = 0.907 further confirms the excellent relationship between predictions and actual values.

For Decision Tree, AUC = 0.933 indicates good but slightly inferior performance compared to the other models, with CA = 0.960, reflecting 96% accuracy in predictions. The F1 score is 0.962, and precision is 0.965, while recall is 0.960, suggesting good effectiveness in identifying the positive class, with MCC = 0.800 showing a solid relationship between predictions and actual values.

Finally, Naive Bayes has an AUC of 0.998, which is excellent in distinguishing between classes. However, CA = 0.870 is lower than the other models, indicating that only 87% of predictions were correct. With precision of 0.944 and excellent accuracy in guessing positive classes, The F1 score of 0.891 indicates that recall and accuracy are well-balanced. While the recall of 0.870 suggests that the model is less effective in identifying all positive categories, the MCC of 0.608 demonstrates a worse correlation between predictions and actual values.

Table 7. The result of the model machine learning classification (%)

Model	AUC	CA	F1	Pre	Rec	MCC
Random Forest	99.4	97.5	97.4	97.4	97.5	85.4
kNN	99.1	96.4	96.2	96.4	96.4	78.4
Tree	93.3	96.0	96.2	96.5	96.0	80.0
Naive Bayes	99.8	87.0	89.1	94.4	87.0	60.8

The Random Forest approach generates the most accurate machine learning classification approach for detecting Bot-IoT activity, with a total accuracy of 97.5%.

The evaluation model for each classification model, where the positive notation represents the non-attack (normal) label, and the positive notation represents the attack label, displayed in the confusion matrix table for each algorithm used, shown in Figures 2 until 5:

		Predicted		
		Attack	**Non-Attack**	∑
Actual	**Attack**	146	32	178
	Non-Attack	13	1609	1622
	∑	159	1641	1800

Figure 2. Confusion Matrix for Random Forest

		Predicted		
		Attack	**Non-Attack**	∑
Actual	**Attack**	120	58	178
	Non-Attack	6	1616	1622
	∑	126	1674	1800

Figure 3. Confusion Matrix for kNN

		Predicted		
		Attack	**Non-Attack**	∑
Actual	**Attack**	161	17	178
	Non-Attack	55	1567	1622
	∑	159	1641	1800

Figure 4. Confusion Matrix for Decision Tree

		Predicted		
		Attack	**Non-Attack**	∑
Actual	**Attack**	178	0	178
	Non-Attack	234	1388	1622
	∑	159	1641	1800

Figure 5. Confusion Matrix for Naive Bayes

The researcher made a comparison between the model in this study and previous research that used the same dataset, specifically the Bot-IoT UNSW Canberra 2018 dataset. As described in Table 8, the proposed approach and the previous study model are contrasted.

The comparison with previous research models shows that, in real-time processing conditions, the analysis of attack values is significantly higher than non-attack (normal) values under an imbalanced data condition. The classification models took 1270.5 seconds to compute and had an accuracy of 98–99%. The computation time was somewhat lengthy, despite the accuracy being nearly flawless. In the following study, when the SMOTE treatment was applied to balance the data using the k-NN algorithm, an accuracy of 92.1% was obtained, while Naïve Bayes achieved 51.5%. However, the study did not report the computation time.The current research model uses the Random Forest approach, which yielded a 97.7% accuracy rate, to aggregate data to manage unbalanced data, compared to 96.6% for k-NN, 85.5% for Naïve Bayes, and 95.5% for Decision Tree. The total computation time recorded was 11.54 seconds. These results indicate that the proposed model, which applies a new treatment for imbalanced data in the Bot-IoT dataset using data aggregation, achieves an accuracy ranging from 85% to 97%, approaching near-perfect accuracy, while significantly reducing computation time compared to previous models.

Table 8. Comparison result

Model	Dataset	Feature	Accuracy (%) RF	kNN	NB	DT	SVM	Training Time (Seconds)	Correlation Analysis
Koroniotis et al[16]	Bot-IoT (Real Time Data)	10	-	-	-	-	88.37	1270.5	√
Kerrakchou et al[12]	Bot-Iot (Real Time Data)	9	99.99	-	98.13	99.88	-	-	-
Pokhrel et al[34]	Bot-Iot (Real Time Data)	8	-	99.6	99.4	-	-	-	-
	Bot-Iot (SMOTE Data)		-	92.1	51.5	-	-	-	-
Proposed Model	Bot-Iot (Aggregate Data)	10	97.7	96.6	85.5	95.5	-	11.54	√

4. Conclusions

An imbalance in the data is revealed by the research's findings on the Bot-IoT UNSW Canberra 2018 dataset, which initially displayed 2,934,447 attack activity and just 370 normal activities. To address this, a novel data aggregation technique was applied to reduce the dataset, focusing on patterns and trends with similar characteristics, resulting in a balanced dataset with 8 attack activities and 80 normal activities. This aggregated data underwent feature selection optimization using ANOVA, leading to the identification of 10 best features out of 17. Subsequently, Approach to classification: An 80:20 training model split and 100 iterations were employed for training Random Forest, kNN, Naive Bayes, and Decision Tree. The findings showed that the Random Forest was the best model, attaining 97.5% accuracy, 97.4% precision, and 97.4% recall. N IN Conn P DestIP and Seq had the highest correlation (+0.937), out of 45 feature correlation results from correlation analysis using Pearson correlation. On the other hand, the mean and N IN Conn P SrcIP had a kind of negative correlation (-0.004), which means that the mean value decreases when N IN Conn P SrcIP increases. From the combination of aggregating data to the machine learning classification procedure, the computing time came to 11.54 seconds. Future research could focus on simulation models with direct implementation on commonly used IoT devices and incorporate deep

learning models to learn and identify new bot attack patterns.

References

[1] L. L. Dhirani, E. Armstrong, and T. Newe, "Industrial iot, cyber threats, and standards landscape: Evaluation and roadmap," *Sensors*, vol. 21, no. 11. 2021. doi: 10.3390/s21113901.

[2] S. A. Rahman, H. Tout, C. Talhi, and A. Mourad, "Internet of Things intrusion Detection: Centralized, On-Device, or Federated Learning?," *IEEE Netw.*, vol. 34, no. 6, pp. 310–317, 2020, doi: 10.1109/MNET.011.2000286.

[3] S. Lee, A. Abdullah, N. Jhanjhi, and S. Kok, "Classification of botnet attacks in IoT smart factory using honeypot combined with machine learning," *PeerJ Comput. Sci.*, vol. 7, 2021, doi: 10.7717/PEERJ-CS.350.

[4] M. H. Nasir, J. Arshad, and M. M. Khan, "Collaborative device-level botnet detection for internet of things," *Comput. Secur.*, vol. 129, p. 103172, 2023, doi: 10.1016/j.cose.2023.103172.

[5] M. A. R. Putra, T. Ahmad, and D. P. Hostiadi, "Analysis of Botnet Attack Communication Pattern Behavior on Computer Networks," *Int. J. Intell. Eng. Syst.*, vol. 15, no. 4, pp. 533–544, 2022, doi: 10.22266/ijies2022.0831.48.

[6] M. Safaei Pour, C. Nader, K. Friday, and E. Bou-Harb, "A Comprehensive Survey of Recent Internet Measurement Techniques for Cyber Security," *Comput. Secur.*, vol. 128, p. 103123, 2023, doi: 10.1016/j.cose.2023.103123.

[7] M. A. R. Putra, D. P. Hostiadi, and T. Ahmad, "Simultaneous Botnet Dataset Generator: A simulation tool for generating a botnet dataset with simultaneous attack characteristic[Formula presented]," *Softw. Impacts*, vol. 14, 2022, doi: 10.1016/j.simpa.2022.100441.

[8] W. A. Safitri, T. Ahmad, and D. P. Hostiadi, "Analyzing Machine Learning-based Feature Selection for Botnet Detection," *2022 1st Int. Conf. Inf. Syst. Inf. Technol. ICISIT 2022*, pp. 386–391, 2022, doi: 10.1109/ICISIT54091.2022.9872812.

[9] P. Jithu, J. Shareena, A. Ramdas, and A. P. Haripriya, "Intrusion Detection System for IOT Botnet Attacks Using Deep Learning," *SN Comput. Sci.*, vol. 2, no. 3, 2021, doi: 10.1007/s42979-021-00516-9.

[10] D. P. Hostiadi, Y. P. Atmojo, R. R. Huizen, I. M. D. Susila, G. A. Pradipta, and I. M. Liandana, "A New Approach Feature Selection for Intrusion Detection System Using Correlation Analysis," *2022 4th Int. Conf. Cybern. Intell. Syst. ICORIS 2022*, 2022, doi: 10.1109/ICORIS56080.2022.10031468.

[11] C. E. Beckerman, "Is there a cyber security dilemma?," *J. Cybersecurity*, vol. 8, no. 1, pp. 1–14, 2022, doi: 10.1093/cybsec/tyac012.

[12] I. Kerrakchou, A. A. El Hassan, S. Chadli, M. Emharraf, and M. Saber, "Selection of efficient machine learning algorithm on Bot-IoT dataset for intrusion detection in internet of things networks," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 31, no. 3, pp. 1784–1793, 2023, doi: 10.11591/ijeecs.v31.i3.pp1784-1793.

[13] D. P. Hostiadi, T. Ahmad, and W. Wibisono, "A New Approach of Botnet Activity Detection Model based on Time Periodic Analysis," *CENIM 2020 - Proceeding Int. Conf. Comput. Eng. Network, Intell. Multimed. 2020*, no. Cenim, pp. 315–320, 2020, doi: 10.1109/CENIM51130.2020.9297846.

[14] Z. Halim *et al.*, "An effective genetic algorithm-based feature selection method for intrusion detection systems," *Comput. Secur.*, vol. 110, p. 102448, 2021, doi: 10.1016/j.cose.2021.102448.

[15] X. Liu and Y. Du, "Towards Effective Feature Selection for IoT Botnet Attack Detection Using a Genetic Algorithm," *Electron.*, vol. 12, no. 5, 2023, doi: 10.3390/electronics12051260.

[16] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset," *Futur. Gener. Comput. Syst.*, vol. 100, pp. 779–796, 2019, doi: 10.1016/j.future.2019.05.041.

[17] F. Taher, M. Abdel-Salam, M. Elhoseny, and I. M. El-Hasnony, "Reliable Machine Learning Model for IIoT Botnet Detection," *IEEE Access*, vol. 11, 2023, doi: 10.1109/ACCESS.2023.3253432.

[18] M. A. R. Putra, U. L. Yuhana, T. Ahmad, and D. P. Hostiadi, "Analyzing the Effect of Network Traffic Segmentation on the Accuracy of Botnet Activity Detection," *Proceeding Int. Conf. Comput. Eng. Netw. Intell. Multimedia, CENIM 2022*, pp. 321–326, 2022, doi: 10.1109/CENIM56801.2022.10037365.

[19] D. P. Hostiadi, T. Ahmad, M. A. R. Putra, G. A. Pradipta, P. D. W. Ayu, and M. Liandana, "A New Approach of Botnet Activity Detection Models Using Combination of Univariate and ANOVA Feature Selection Techniques," *Int. J. Intell. Eng. Syst.*, vol. 17, no. 3, pp. 485–502, 2024, doi: 10.22266/ijies2024.0630.38.

[20] M. Matsumoto, A. S. M. Miah, N. Asai, and J. Shin, "Machine Learning-Based Differential Diagnosis of Parkinson's Disease Using Kinematic Feature Extraction and Selection," pp. 1–15, 2025, [Online]. Available: http://arxiv.org/abs/2501.02014

[21] H. Pan, X. You, S. Liu, and D. Zhang, "Pearson correlation coefficient-based pheromone refactoring mechanism for multi-colony ant colony optimization," *Appl. Intell.*, vol. 51, no. 2, pp. 752–774, 2021, doi: 10.1007/s10489-020-01841-x.

[22] F. H. Moh'd, K. A. Notodiputro, and Y. Angraini, "Enhancing interpretability in random forest: Leveraging inTrees for association rule extraction insights," *IAES Int. J. Artif. Intell.*, vol. 13, no. 4, pp. 4054–4061, 2024, doi: 10.11591/ijai.v13.i4.pp4054-4061.

[23] P. R. Maidamwar, P. P. Lokulwar, and K. Kumar, "Ensemble Learning Approach for Classification of Network Intrusion Detection in IoT Environment," *Int. J. Comput. Netw. Inf. Secur.*, vol. 15, no. 3, 2023, doi: 10.5815/ijcnis.2023.03.03.

[24] A. Agarwal, P. Sharma, M. Alshehri, A. A. Mohamed, and O. Alfarraj, "Classification model for accuracy and intrusion detection using machine learning approach," *PeerJ Comput. Sci.*, vol. 7, pp. 1–22, 2021, doi: 10.7717/PEERJ-CS.437.

[25] T. H. H. Aldhyani and H. Alkahtani, "Artificial Intelligence Algorithm-Based Economic Denial of Sustainability Attack Detection Systems: Cloud Computing Environments," *Sensors*, vol. 22, no. 13, 2022, doi: 10.3390/s22134685.

[26] B. Charbuty and A. Abdulazeez, "Classification Based on Decision Tree Algorithm for Machine Learning," *J. Appl. Sci. Technol. Trends*, vol. 2, no. 01, pp. 20–28, 2021, doi: 10.38094/jastt20165.

[27] M. Panda, A. A. A. Mousa, and A. E. Hassanien, "Developing an Efficient Feature Engineering and Machine Learning Model for Detecting IoT-Botnet Cyber Attacks," *IEEE Access*, vol. 9, 2021, doi: 10.1109/ACCESS.2021.3092054.

[28] K. Alissa, T. Alyas, K. Zafar, Q. Abbas, N. Tabassum, and S. Sakib, "Botnet Attack Detection in IoT Using Machine Learning," *Comput. Intell. Neurosci.*, vol. 2022, 2022, doi: 10.1155/2022/4515642.

[29] S. Vanitha and P. Balasubramanie, "Improved Ant Colony Optimization and Machine Learning Based Ensemble Intrusion Detection Model," *Intell. Autom. Soft Comput.*, vol. 36, no. 1, 2023, doi: 10.32604/iasc.2023.032324.

[30] M. A. R. Putra, T. Ahmad, D. P. Hostiadi, R. M. Ijtihadie, and P. Maniriho, "Botnet Attack Analysis through Graph Visualization," *Int. J. Intell. Eng. Syst.*, vol. 17, no. 1, 2024, doi: 10.22266/ijies2024.0229.75.

[31] Q. Long, "A Bayesian explanation of machine learning models based on modes and functional ANOVA," 2024, [Online]. Available: http://arxiv.org/abs/2411.02746

[32] S. El Hajla, E. M. Ennaji, Y. Maleh, and S. Mounir, "Enhancing IoT network defense: advanced intrusion detection via ensemble learning techniques," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 35, no. 3, pp. 2010–2020, 2024, doi: 10.11591/ijeecs.v35.i3.pp2010-2020.

[33] R. Sistem, "JURNAL RESTI Comparison of Machine Learning Algorithms in Detecting Tea Leaf," vol. 5, no. 158, pp. 6–12, 2024.

[34] S. Pokhrel, R. Abbas, and B. Aryal, "IoT Security: Botnet detection in IoT using Machine learning," no. April, 2021, doi: 10.48550/arXiv.2104.02231.

