

Revolutionizing internet of things intrusion detection using machine learning with unidirectional, bidirectional, and packet features

Zulhipni Reno Saputra Elsi¹, Deris Stiawan², Bhakti Yudho Suprpto³, M. Agus Syamsul Arifin⁴, Mohd. Yazid Idris⁵, Rahmat Budiarto⁶

¹Faculty of Engineering, Universitas Muhammadiyah Palembang, Palembang, Indonesia

²Department of Computer Engineering, Faculty of Computer Science, Sriwijaya University, Palembang, Indonesia

³Faculty of Engineering, Sriwijaya University, Palembang, Indonesia

⁴Departement of Informatic, Faculty of Engineering, Universitas Jenderal Soedirman, Purwokerto, Indonesia

⁵Faculty of Computing, Universiti Teknologi Malaysia, Johor Bahru, Malaysia

⁶College of Computing and Information, Al Baha University, Al Bahah, Saudi Arabia

Article Info

Article history:

Received Jul 6, 2024

Revised Jun 17, 2025

Accepted Jul 10, 2025

Keywords:

Bidirectional

Correlation-based feature selection

Data type-based feature selection

Package features

Unidirectional

ABSTRACT

Detection of attacks on internet of things (IoT) networks is an important challenge that requires effective and efficient solutions. This study proposes the use of various machine learning (ML) techniques in classifying attacks using unidirectional, bidirectional, and packet features. The proposed methods that implement decision tree (DT), random forest (RF), extreme gradient boosting classifier (XGBC), AdaBoost (AB) and linear discriminant analysis (LDA) work perfectly with all kinds of datasets and includes. It also works very well with data type-based feature selection (DTBFS) and correlation-based feature selection (CBFS). The experiment results show a significant improvement compared to previous studies and reveals that unidirectional and bidirectional features provide higher accuracy compared to packet features. Furthermore, ML models, particularly DT, and RF, have faster computing times compared to more complex deep learning models. This analysis also shows potential overfitting in some models, which requires further validation with different datasets. Based on these findings, we recommend the use of RF and DT for scenarios with unidirectional and bidirectional features, while AB and LDA for packet features. The study concludes that using the right ML techniques along with features that work in both directions can make an intrusion detection system for IoT networks becomes very accurate.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Deris Stiawan

Department of Computer Engineering, Faculty of Computer Science, Sriwijaya University

Palembang, Indonesia

Email: deris@unsri.ac.id

1. INTRODUCTION

The growing use of internet of things (IoT) devices in many industries has created an urgent need for efficient security processes [1]. The IoT devices are internet-connected devices commonly employed in diverse settings, ranging from connected households to industrial systems [2], [3]. The limited computing resources and insecure communication protocols of these devices make them susceptible to cyber-attacks [4]. message queue telemetry transport (MQTT) is a commonly employed protocol in IoT networks, specifically

developed for low complexity communications [5]. The primary issue in this domain is the identification and categorization of attacks targeting IoT devices that utilizing the MQTT protocol [6].

Challenges in detecting cyber-attacks on IoT devices include constrained device processing capabilities, intricate and diverse attack types, and large amounts of data [7]. Given the typically constrained processing and storage capacity of IoT devices, it is often unfeasible to deploy intricate security measures [8]. Moreover, the attacks on IoT devices are diverse, ranging from denial of service (DoS) attacks to malware entry, necessitating adaptable, and efficient detection methods [9], [10].

Employing statistical techniques for extracting features from packet header flow, including unidirectional and bidirectional characteristics, as well as general packet features from protocols such as MQTT, transmission control protocol (TCP) (including IP packets and IP flows), and user datagram protocol (UDP) [11]–[15], is a viable approach to address this issue. This functionality allows for thorough examination of network traffic patterns linked to attacks without the need for excessive data processing [13]. By extracting packet features from the MQTT-IoT-IDS2020 dataset, a comprehensive database is obtained for training attack detection models. In order to develop more precise and comprehensive detection models, this dataset encompasses a broad spectrum of typical attack types against the MQTT protocol.

Using unidirectional and bidirectional capabilities, the system can assess network traffic from either a one-way or two-way standpoint, so offering a more comprehensive understanding of network activity [14], [15]. A unidirectional feature is designed to analyze data that moves in a single direction, such from a device to a server, whereas a bidirectional feature is designed to analyze data that moves in both directions between a device and a server. Examining the traffic of the TCP and UDP protocols further expands upon this methodology. The TCP protocol's examination of IP packets and IP flows enables the detection of likely communication patterns and irregularities in ongoing connections, while the characteristics of the UDP protocol facilitate the detection of attack patterns that arise in connectionless communications [16], [17].

By reducing the computational burden on IoT devices, this method allows for early detection and improved categorization of attacks. The development of an effective and efficient security system to protect IoT devices from ever-evolving cyber threats relies on the use of statistical feature extraction techniques and the MQTT-IoT-IDS2020 dataset. This study enhances the advancement of an attack detection and classification system for IoT devices by employing efficient and effective methods for extracting meaningful features. The following are few significant contributions that this research has made: i) statistical methods utilization for extracting features that depend on the characteristics of packet header flow, particularly unidirectional and bidirectional features, in order to detect possible attacks; ii) packet feature extraction approach derived from the MQTT, TCP, and UDP protocols; iii) evaluation and comparison using the MQTT-IoT-IDS2020 dataset; and iv) accuracy enhancement and comprehensiveness of detection model, encompassing a range of typical attacks targeting the MQTT protocol.

2. RELATED WORK

Related researches about intrusion detection in IoT networks have adopted various techniques, including preprocessing, feature extraction, and classification. Alasmari and Alhogail [18] used a generalized linear model (GLM) with random over-sampling and automatic feature engineering to make an optimization model that was 100% accurate and had a 100% F1-score. Automatic feature engineering also improved performance by 38.9% and reduced detection time by 67.7%. However, this research is exclusive to the MQTT protocol for smart home environments, lacking testing on other protocols or broader IoT scenarios. Aliabdi [19] suggested a mixed algorithm that uses both a convolutional neural network (CNN) and long short-term memory (LSTM). On the network security lab-knowledge discovery and data mining (NSL-KDD) dataset, the proposed algorithm achieved 99% accuracy, and on the MQTT protocol, achieved over 97% accuracy. However, the complexity of this algorithm may not be suitable for IoT devices with limited resources. Liu *et al.* [20] created a multi-node, multi-class classification ensemble approach to find attacks in distributed cyber-physical systems. In situations where multiple nodes were censoring data, this approach worked better than the full-data approach. However, the complexity of this approach is high and limited to specific data-censoring scenarios.

Chen *et al.* [21] used a hybrid feature selection and layered classification model, which outperformed six machine learning (ML)/decision tree (DT) algorithms in accuracy and resource consumption on four public datasets. However, the complexity of this method may not be suitable for low-resource IoT devices. Gorzalczy and Rudzinski [22] improved a fuzzy algorithm-based classification system using a multi-objective evolutionary algorithm. The system worked better in terms of accuracy and simplicity, with ease of understanding being the main focus. In the meantime, Chaganti *et al.* [23] developed a bidirectional gated recurrent unit (Bi-GRU)-CNN model for IoT malware detection and classification,

which achieved 100% accuracy for IoT malware detection and 98% for IoT malware family classification. However, they restricted the evaluation to features like byte sequences.

Attota *et al* [24] proposed a federated learning-based intrusion detection (MV-FLID) method using multi-view ensemble learning. This method was more accurate than centralized non-federated learning (FL) methods, however, it is still challenging to implement and needs a lot of resources. Also, Liu *et al.* [25] created a bidirectional gated recurrent unit attention (BGRUA) deep learning model for classifying hypertext transfer protocol secure (HTTPS) traffic. This model does a better job of classifying encrypted traffic than other methods in terms of accuracy, precision, recall, and F1-score, but it also only classifies HTTPS traffic. Samy *et al.* [26] also developed an attack detection framework using deep learning and implemented it on fog nodes. Researchers only tested it on fog nodes, achieving a detection rate of 99.97%, an accuracy of 99.96% in binary class classification, and an accuracy of 99.65% in multiclass classification. Huang *et al.* [27] also worked on a k-nearest neighbor (KNN)-based classification model that uses statistical features from header-derived flow and achieves about 90% accuracy while trying to use as little computing power as possible. Table 1 summarizes the other important related works.

Table 1. Summary of ML techniques for detecting IoT anomalies and attacks

Ref	Dataset	Attack types	Techniques	Performance metrics	Drawbacks/Gap
[28]	MQTT dataset	MQTT-enabled IoT security	Hybrid feature selection (XGBoost, MaxPoolingID)	Accuracy, precision, recall, F1-score	Limited to MQTT datasets; generalization to other untested protocols
[29]	MQTT-IoT-IDS-2020, NSL-KDD	Various network intrusions	ML-based (normalization, oversampling, undersampling)	Accuracy, time efficiency	Complex pre-processing pipeline; performance on non-IoT data sets has not been fully explored
[30]	CIC DoS 2017	Low-rate denial of service (LR DoS)	AI-based anomaly detection (FCNN)	Accuracy, precision, recall, F1-score, detection time, ROC	Focuses only on LR DoS attacks; efficiency on other types of attacks was not demonstrated
[31]	TON-IoT	IoT network intrusions	Feature extraction vs. feature selection	Accuracy, F1-score, runtime	Feature selection provides faster results but potentially reduces accuracy; more room for increased accuracy
[32]	CICIDS2017	Various network intrusions	General intrusion detection framework (autoencoder, classification)	Accuracy (high for both binary and multiclass classification)	Complex frameworks may be overkill for environments with fewer resources
[32]	NSL-KDD	DDoS, PROBE, R2L, U2R	Tree-based ML techniques (DT, RF, XGBoost)	Accuracy	Only uses five features; may not capture the full spectrum of network behavior
[33]	UNSW-NB15	Various IoT intrusions	Feature clusters (flow, MQTT, TCP)	Accuracy (binary: dan multi-class)	Especially for UNSW-NB15; other data sets may not provide similar results
[34]	CSE-CIC-IDS2018	DDoS attacks	Feature-engineering and ML-based detection (RF, SVM, KNN, DT, XGBoost)	Accuracy, precision, recall, F1-score	Focus on DDoS; its applicability to other types of attacks has not been tested
[35]	NSL-KDD, UNSW-NB15, CCIDS2017	Various IoT intrusions	Extreme gradient ensemble boosting, feature selection	Accuracy	High computational complexity; may not be suitable for low-resource IoT devices
[36]	BoT-IoT	DDoS, DoS, Reconnaissance, Information Theft	Supervised ML (KNN, LR, SVM, MLP, DT, RF)	Accuracy, precision, recall, F1-score, ROC	Limited to BoT-IoT dataset; effectiveness on other non-validated data sets

3. METHOD

This section outlines the steps and decisions made during the process of proposing a new IDS to detect attacks in IoT networks. It presents the ML architecture designed for attack detection and explains the feature extraction techniques used. Furthermore, it describes the feature selection process, the classification algorithm applied, and the use of the confusion matrix for evaluation.

3.1. Proposed method

This study introduces a novel integration of unidirectional, bidirectional, and packet-level features for detecting IoT attacks. Each feature type offers a unique view of the data such as unidirectional and bidirectional features provide statistical flow characteristics, while packet features reflect protocol-level

attributes. Their combination ensures the detection system captures traffic behaviors and protocol abuse patterns, thereby improving accuracy and robustness.

Figure 1 illustrates the proposed method, which is divided into several processes. The first process is feature extraction with three feature extractions, namely unidirectional-based features, bidirectional-based features and packet-based features. This feature extraction process produces 3 new datasets for the 3 feature extraction processes. The second process is featuring selection by eliminating features using data type-based feature selection (DTBFS), eliminating features that have data object, data types and correlation-based feature selection (CBFS) with threshold=0.8. The third step is to perform classification task using the 5 selected algorithms, i.e.: DT, random forest (RF), extreme gradient boosting classifier (XGBC), AdaBoost (AB), linear discriminant analysis (LDA), and finally compare the performance of the matrix for each classification, the performance compared is accuracy, precision, recall, F1-score, and performance time.

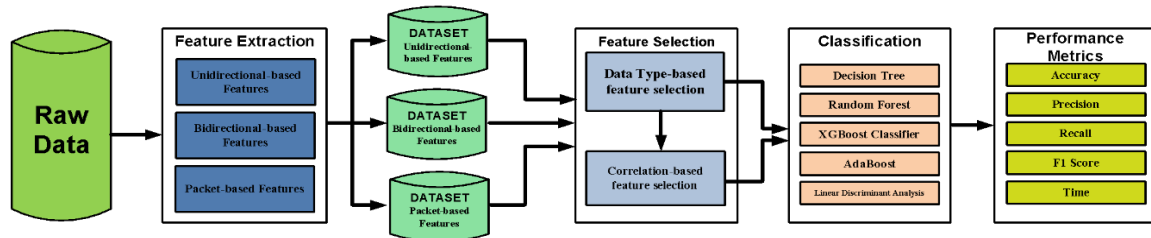


Figure 1. ML architecture of the proposed method

Five files (in .pcap format) consist of raw data: normal, scan_a, scan_su, sparta, and mqtt_bruteforce. We pre-process each file using unidirectional extraction, bidirectional extraction, and packet features, where each raw data will be 3 new files (*.csv). Figure 2 illustrates the process of converting 1 file into 3 files (in .csv format), such as normal raw data will become 3 files, namely uniflow_Normal.csv, biflow_Normal.csv, and packet_Normal.csv then data sets, such as uniflow: uniflow_Normal.csv, uniflow_scan_A.csv, uniflow_scan_sU.csv, uniflow_sparta.csv, and uniflow_mqtt_bruteforce.csv will be combined into 1 new .csv file with 5 classes. Finally, from 5 raw pcap data, 3 csv files will be obtained, namely Combined unidirectional_multi_class.csv, Combined bidirectional_multi_class.csv, and Combined Packet_feature_multi_class.csv.

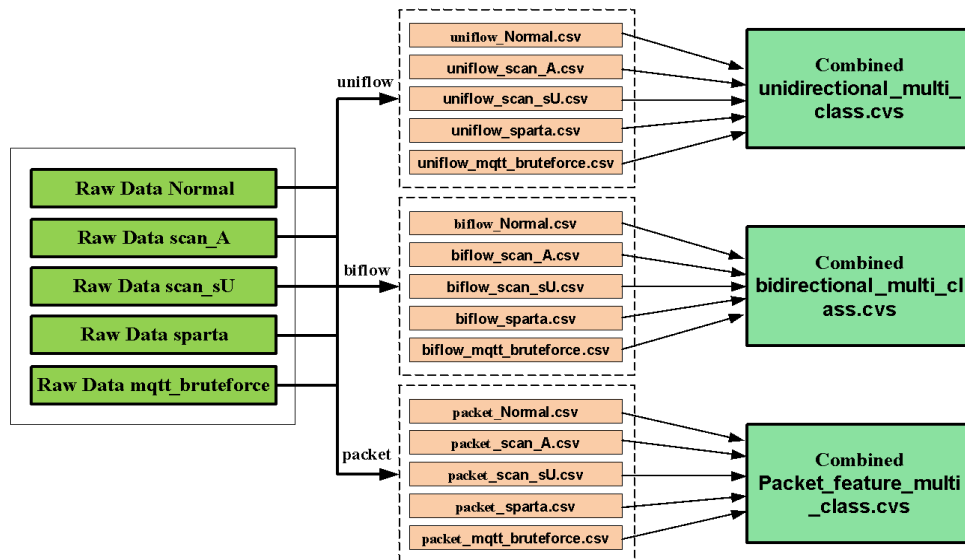


Figure 2. MQTT-IoT-IDS2020 pre-processing

3.2. MQTT-IoT-IDS2020 dataset

This study uses the MQTT-IoT-IDS2020 [37] dataset due to its focus on MQTT-based traffic, which is highly relevant in real-world smart home and lightweight IoT network deployments. This dataset includes modern intrusion attempts such as scanning, brute-force, and session hijacking, making it a suitable

benchmark for validating intrusion detection models. The next paragraph is a detailed explanation of each data component in the dataset.

- Normal data: the normal data in this dataset reflects the daily activities of an IoT network without any attacks. This data includes regular communications between IoT devices and MQTT servers. This normal activity is important for training anomaly and attack detection models, as it provides a baseline of expected network behavior.
- Scan_A data: scan_A data describes a network scanning attack carried out by an attacker to identify vulnerable devices. These attacks typically include port scanning and IP scanning to find weak points in the network that can be exploited further. Scan_sU data: scan_sU data covers more specific types of scanning attacks, often involving more in-depth and targeted scans to identify services running on a particular device. These attacks may include UDP scanning and scanning of specific services that use the MQTT protocol.
- Spartan data: sparta's data refers to a specific type of attack that uses a tool called Sparta to perform security scans against IoT networks. Sparta is a powerful scanning tool and used to identify vulnerabilities in various network services. This data includes the results of an attack that used Spartan scanning techniques against IoT devices which communicate via MQTT.
- MQTT-Bruteforce data: MQTT-Bruteforce data reflects brute force attacks against MQTT servers. In this attack, the attacker tries various username and password combinations with hope can illegally accessing the MQTT server. This data includes logs of failed as well as successful login attempts, providing insight into brute force attack patterns against MQTT servers.

3.3. Feature extraction

The data extraction process was carried out using the Scapy and dpkt libraries to read PCAP files containing network traffic. After the data was successfully extracted, the Pandas library was used to manage and manipulate the data in the form of a dataframe, facilitating further analysis. All extraction results were then saved in CSV format for efficient use in the subsequent model processing and training stages.

3.3.1. Unidirectional features

These features represent one-way traffic statistics, such as from client to server. Extracted metrics include packet count, byte count, inter-arrival time statistics, and average packet size. These are critical for detecting one-way anomalies like flooding or scanning.

3.3.2. Bidirectional features

Bidirectional features capture the full session context between communicating hosts. They include forward and backward packet counts, data volume, response delays, and flag usage. These features allow the model to analyze request-response consistency and session symmetry.

3.3.3. Packet-level features

These features are derived directly from the MQTT, TCP, and UDP packet headers. They include flags (e.g., SYN, ACK, and MQTT QoS), status codes, and metadata such as IP/MAC addresses. These are essential for identifying protocol-level misuse and malformed packet behavior.

3.4. Feature selection

Feature selection is a crucial process in data modeling that aims to select the most relevant attributes from raw data to improve model performance and reduce computational complexity. In this research, the feature selection process is conducted in two stages. The two stages are DTBFS and CBFS.

In the first stage, DTBFS, we consider the data types present in the MQTT-IoT-IDS2020 dataset, which includes integer, float, and object types. We focus exclusively on integer and float features, as these numeric types can be directly utilized by ML algorithms for modeling and attack detection. Features with the object data type are removed except for those indicating the class or type of attack because they require additional processing such as encoding, which can introduce complexity and increase computational time. While this step may risk excluding certain categorical metadata, redundant protocol identifiers and categorical information are often represented numerically in other retained features, ensuring minimal information loss. By filtering the dataset in this way, we streamline the data to contain only numeric values, making it ready for efficient analysis and model training.

In the second stage, CBFS is applied using the Pearson correlation method. This approach is used to measure the linear relationship between features and identify those with a significant influence on the target variable. A commonly used correlation threshold of 0.8, as cited in the feature selection literature [38], is employed to identify and eliminate multicollinearity among features. Features with high correlation to the target variable but low correlation with each other are retained to ensure uniqueness and relevance is shown in Algorithm 1. This step reduces data redundancy and simplifies the model, ultimately improving

interpretability while minimizing the risk of overfitting. Through this two-stage feature selection process, we enhance the overall performance, efficiency, and accuracy of the IoT network intrusion detection model.

Algorithm 1. Feature selection algorithm (DTBFS+CBFS)

```

1: Input: Dataset D with multiple features including numeric and object types.
2: Initialize:
3:   label_column=column that contains class/attack label
4:   numeric_features=empty list
5: Step 1: Drop object-type features (DTBFS)
6:   For each feature f in dataset D:
7:     If f is of object type and f≠label_column:
8:       Drop f from dataset D
9:     Else if f is numeric (integer or float):
10:      Add f to numeric_features
11: Step 2: Calculate pearson correlation (CBFS)
12:   Compute correlation matrix C for all features in numeric_features
13: Step 3: Remove highly correlated features
14:   For each pair of features (f1, f2) in C:
15:     If |C[f1][f2]| > 0.8:
16:       Drop one of the features (e.g., f2) based on lower correlation with target or
       domain relevance
17: Output: Reduced dataset D_reduced with selected features.
```

3.5. Classification

In the process of detecting attacks on IoT networks, selecting the right classification algorithm is crucial for achieving optimal accuracy and efficiency. In this study, we utilize several popular and proven classification algorithms widely used in various ML applications. These include DT, RF, XGBC, AB, and LDA

DT is an algorithm that builds a prediction model using a DT structure. Each node in the tree represents a feature; each branch represents a decision; and each leaf represents an outcome. The main advantage of DT is its high interpretability, which makes it easy to understand and visualize. RF is a development of DT that combines a number of DT to increase accuracy and reduce overfitting. Using bagging techniques, RF builds many DT from different subsets of data and combines the results.

XGBC is a boosting algorithm that combines many weak decisions tree models to form a strong model. XGBC is renowned for its high speed and performance, as well as its ability to handle large and imbalanced datasets. This algorithm iteratively corrects previous model errors, focusing each new tree on the mistakes made by the previous tree. Meanwhile, AB is another boosting algorithm that combines a number of weak DT models to form a strong model. However, unlike XGBC, AB adjusts the weight of each data instance based on the error of the previous model, so that data that is difficult to classify gets more attention in the next iteration. This algorithm is effective in increasing model accuracy on data that is not too large and complex.

One of the objectives of the statistical technique known as LDA is to identify linear feature combinations that can be used to differentiate between two or more classes in the data. This technique is frequently utilized in the processes of pattern recognition, classification, and dimensionality reduction. LDA is a technique that endeavors to project data into a space with fewer dimensions while successfully preserving the various classes.

3.6. Confusion matrix

Confusion matrix is a very useful tool in evaluating the performance of classification models. This matrix provides a clear picture of how the classification model makes predictions on test data by comparing the model predictions with the actual labels. The confusion matrix consists of four main components: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). TP: number of cases where the model correctly predicted the positive class. TN: number of cases where the model correctly predicted the negative class, FP: number of cases where the model incorrectly predicted the positive class when it should have been negative. FN: number of cases where the model incorrectly predicted a negative class when it should have been positive.

Using the confusion matrix, we can calculate several other important evaluation metrics such as accuracy, precision, recall, and F1-score, all of which provide deeper insight about model performance as shown in accuracy (1): the proportion of correct predictions out of all predictions, is a general idea of how often the model makes correct predictions. Precision (2): the proportion of correct positive predictions. Recall (3): the proportion of total positives that were correctly detected. F1-score (4): F1-score provides a balance between precision and recall and is very useful when the class distribution is unbalanced.

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

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

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

$$F1 - Score = \frac{2(precision*recall)}{precision+recall} \quad (4)$$

4. RESULTS AND DISCUSSION

In this section, we discuss the results of applying unidirectional, bidirectional, and packet feature extraction methods combined with DTBFS and CBFS feature selection for detecting attacks on IoT networks. The classification results based on these extraction and selection methods will be analyzed, including the confusion matrix, accuracy, and processing time. Furthermore, we compare these results with other studies that have used the same dataset. The list of the features derived from different raw data extraction methods are as follows. For unidirectional_multi_class. CSV produces 19 features, while bidirectional_multi_class. csv produces 36 features, and packet_feature_multi_class.CSV produces 53 features. Each dataset (*.csv) consists of object, float64, and int64 data types.

Table 2 presents the names of the features of each dataset. In unidirectional extraction describes one-way traffic between two points (e.g. from source to destination), such as ip_src feature, ip_dst feature as source and destination IP addresses, prt_src feature, prt_dst feature as source and destination ports used in communication, proto feature as protocol used in communication (such as TCP, UDP). Features like num_pkts feature, num_bytes feature is the number of packets and bytes sent in a one-way data stream; mean_iat feature, std_iat feature, min_iat feature, max_iat feature to measure the time between packet arrivals (inter arrival time), this can be used to detect abnormal traffic patterns; and std_pkt_len feature, min_pkt_len feature, max_pkt_len feature are statistics of packet length sent in one direction.

Table 2. Unidirectional, bidirectional, and package feature extraction features in MQTT-IoT-IDS2020

Unidirectional	Bidirectional	Package feature
ip_src	ip_src	fwd_std_pkt_len
ip_dst	ip_dst	bwd_std_pkt_len
prt_src	prt_src	fwd_min_pkt_len
prt_dst	prt_dst	bwd_min_pkt_len
proto	proto	fwd_max_pkt_len
num_pkts	fwd_num_pkts	bwd_max_pkt_len
mean_iat	bwd_num_pkts	fwd_num_bytes
std_iat	fwd_mean_iat	bwd_num_bytes
min_iat	bwd_mean_iat	fwd_num_psh_flags
max_iat	fwd_std_iat	bwd_num_psh_flags
mean_offset	bwd_std_iat	fwd_num_rst_flags
mean_pkt_len	fwd_min_iat	bwd_num_rst_flags
num_bytes	bwd_min_iat	fwd_num_urg_flags
num_psh_flags	fwd_max_iat	bwd_num_urg_flags
num_rst_flags	bwd_max_iat	sec_ip_src
num_urg_flags	fwd_mean_offset	num_src_flows
std_pkt_len	bwd_mean_offset	src_ip_dst_prt_delta
min_pkt_len	fwd_mean_pkt_len	
max_pkt_len	bwd_mean_pkt_len	
		Sta
		dport
		dst_ip
		dst_mac
		dst_port
		f4b_a
		f4b_b
		flags
		id
		ip_a
		ip_b
		ip_flag_df
		ip_flag_mf
		ip_flag_rb
		ip_len
		length
		mac_a
		mac_b
		mqtt_flag_clean
		mqtt_flag_passwd
		mqtt_flag_qos
		mqtt_flag_reserved
		mqtt_flag_retain
		mqtt_flag_uname
		mqtt_flag_willflag
		mqtt_messagelength
		mqtt_message_type
		options
		port_a
		port_b
		prot
		sfp_a
		sfp_b
		sport
		src_ip
		src_mac
		src_port
		tcp_flag_cwr
		tcp_flag_ecn
		tcp_flag_fin
		tcp_flag_ns
		tcp_flag_push
		tcp_flag_res
		tcp_flag_reset
		tcp_flag_syn
		tcp_flag_urg
		timestamp
		tran_prot
		transport
		ts_end
		ts_start
		ttl

Bidirectional extraction describes including two-way traffic data between source and destination, such as fwd_std_pkt_len feature, bwd_std_pkt_len feature is the average length of the packet in the forward and backward directions; fwd_min_pkt_len feature, bwd_min_pkt_len feature is the minimum length of the packet in the forward and backward directions; fwd_max_pkt_len feature, bwd_max_pkt_len feature is the maximum length of the packet sent in the forward and backward directions; fwd_num_pkts feature, bwd_num_pkts feature is the number of packets sent in forward and backward directions; fwd_num_psh_flags feature, bwd_num_psh_flags feature is the number of push flags in packets in each direction; and sec_ip_src feature is the second IP address source used in bidirectional communication. Feature packet extraction describes features with specific protocols and packet characteristics, such as Sta feature, flags feature, options feature about metadata about status and flags in packets; mqtt_flag_passwd feature, mqtt_flag_qos feature, mqtt_flag_uname feature refers to MQTT flag, which is important in IoT communication, because MQTT is a commonly used protocol in IoT networks; tcp_flag_cwr feature, tcp_flag_ecn feature, tcp_flag_syn feature, etc. are related to flags in TCP protocol. These flags indicate the

TCP session status (such as SYN for connection initiation or FIN for connection termination); feature ip_a, feature ip_b, feature mac_a, feature mac_b are the IP and MAC addresses used in the packet.

Figures 3(a) to 3(t) illustrates the confusion matrix with the unidirectional extraction dataset showing the training and testing results of various ML algorithms (DT, RF, XGBoost, AB, and LDA) on two feature selection techniques, namely DTBFS and CBFS. Thus, providing a comprehensive picture of how feature selection affects the predictive ability of each algorithm, while Figures 4(a) to 4(t) illustrates the confusion matrix with the bidirectional extraction dataset, Figures 5(a) to 5(t) illustrates the confusion matrix with the packet feature extraction dataset. Figures 3 to 5 contain the values of TP, TN, FP and FN that can be used to measure the values of precision, recall, and F1-score.

Table 3 shows the precision value of the applied classification models; Table 4 shows the recall value of the applied classification models while Table 5 shows the F1-score values, which are the harmonic mean between precision and recall. All classification models are applied to data with a dividing ratio of 75% for training and 25% for testing. Each table shows how the model responds to data with different characteristics (unidirectional, bidirectional, and packet features) and how the model performance can be improved with an appropriate feature selection method (DTBFS or CBFS).

DT, RF, and XGBC have a value of 100 in precision, recall, and F1 score on both types of DTBS and CBFS feature selection for all types of unidirectional, bidirectional, and parcel features in feature extraction. Meanwhile, AB has a more balanced performance, although AB's precision, recall, and F1 score are lower than other models, its performance is quite consistent and more realistic, especially in the package feature scenario. While LDA performs well on package features, the use of the CBFS method generally improves the performance of LDA compared to DTBFS, making it more suitable for complex datasets, with data precision values ranging from 75.229 to 92.330, recall ranging from 64.207 to 90.584 while F1 score ranging from 63.166 to 85.401.

Table 3. Precision value

Classification	Split data (75%:25%)	Unidirectional		Bidirectional		Packet feature	
		DTBFS	CBFS	DTBFS	CBFS	DTBFS	CBFS
DT	Training data	100.000	100.000	100.000	100.000	100.000	100.000
	Testing data	100.000	100.000	100.000	100.000	100.000	100.000
RF	Training data	100.000	100.000	100.000	100.000	100.000	100.000
	Testing data	100.000	100.000	100.000	100.000	100.000	100.000
XGBC	Training data	100.000	100.000	100.000	100.000	100.000	100.000
	Testing data	100.000	100.000	100.000	100.000	100.000	100.000
AB	Training data	52.270	52.270	51.361	51.361	73.841	73.841
	Testing data	52.278	52.278	51.375	51.375	73.818	73.818
LDA	Training data	75.229	82.072	78.010	78.434	92.330	90.584
	Testing data	75.248	81.986	77.837	78.319	92.242	90.581

Table 4. Recall value

Classification	Split data (75%:25%)	Unidirectional		Bidirectional		Packet feature	
		DTBFS	CBFS	DTBFS	CBFS	DTBFS	CBFS
DT	Training data	100.000	100.000	100.000	100.000	100.000	100.000
	Testing data	100.000	100.000	100.000	100.000	100.000	100.000
RF	Training data	100.000	100.000	100.000	100.000	100.000	100.000
	Testing data	100.000	100.000	100.000	100.000	100.000	100.000
XGBC	Training data	100.000	100.000	100.000	100.000	100.000	100.000
	Testing data	100.000	100.000	100.000	100.000	100.000	100.000
AB	Training data	60.000	60.000	60.000	60.000	80.000	80.000
	Testing data	60.000	60.000	60.000	60.000	80.000	80.000
LDA	Training data	64.207	63.158	69.235	69.565	84.701	83.030
	Testing data	64.258	63.175	69.060	69.458	84.725	83.064

Table 5. F1-score value

Classification	Split data (75%:25%)	Unidirectional		Bidirectional		Packet feature	
		DTBFS	CBFS	DTBFS	CBFS	DTBFS	CBFS
DT	Training data	100.000	100.000	100.000	100.000	100.000	100.000
	Testing data	100.000	100.000	100.000	100.000	100.000	100.000
RF	Training data	100.000	100.000	100.000	100.000	100.000	100.000
	Testing data	100.000	100.000	100.000	100.000	100.000	100.000
XGBC	Training data	100.000	100.000	100.000	100.000	100.000	100.000
	Testing data	100.000	100.000	100.000	100.000	100.000	100.000
AB	Training data	55.210	55.210	54.490	54.490	76.360	76.360
	Testing data	55.215	55.215	54.502	54.502	76.344	76.344
LDA	Training data	67.196	63.166	72.606	72.988	85.401	83.032
	Testing data	67.231	63.143	72.396	72.847	85.427	83.088



Figure 3. Unidirectional matrix confusion: (a) DT-DTBFS training, DT-DTBFS testing, (c) DT-CBFS training, (d) DT-CBFS testing, (e) RF-DTBFS training, (f) RF-DTBFS testing, (g) training RF-CBFS, (h) testing RF-CBFS, (i) training XGBC-DTBFS, (j) testing XGBC-DTBFS, (k) training XGBC-CBFS, (l) testing XGBC-CBFS, (m) training AB-DTBFS, (n) testing AB-DTBFS, (o) training AB-CBFS, (p) testing AB-CBFS, (q) training LDA-DTBFS, (r) testing LDA-DTBFS, (s) training LDA-CBFS, and (t) testing LDA-CBFS

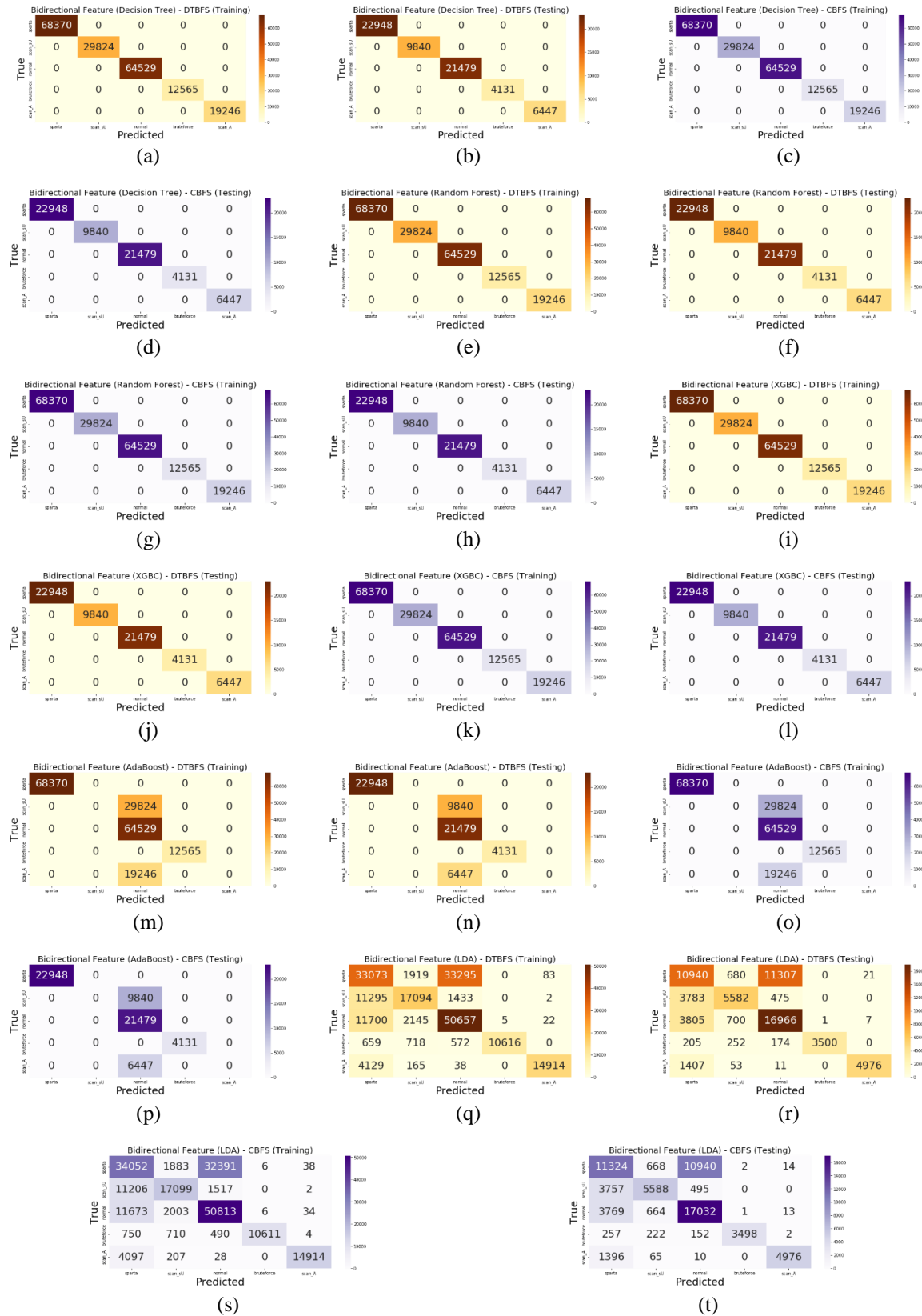


Figure 4. Bidirectional matrix confusion: (a) training DT-DTBFS, (b) testing DT-DTBFS, (c) training DT-CBFS, (d) testing DT-CBFS, (e) training RF-DTBFS, (f) testing RF-DTBFS, (g) training RF-CBFS, (h) testing RF-CBFS, (i) training XGBC-DTBFS, (j) testing XGBC-DTBFS, (k) training XGBC-CBFS, (l) testing XGBC-CBFS, (m) training AB-DTBFS, (n) testing AB-DTBFS, (o) training AB-CBFS, (p) testing AB-CBFS, (q) training LDA-DTBFS, (r) testing LDA-DTBFS, (s) training LDA-CBFS, and (t) testing LDA-CBFS

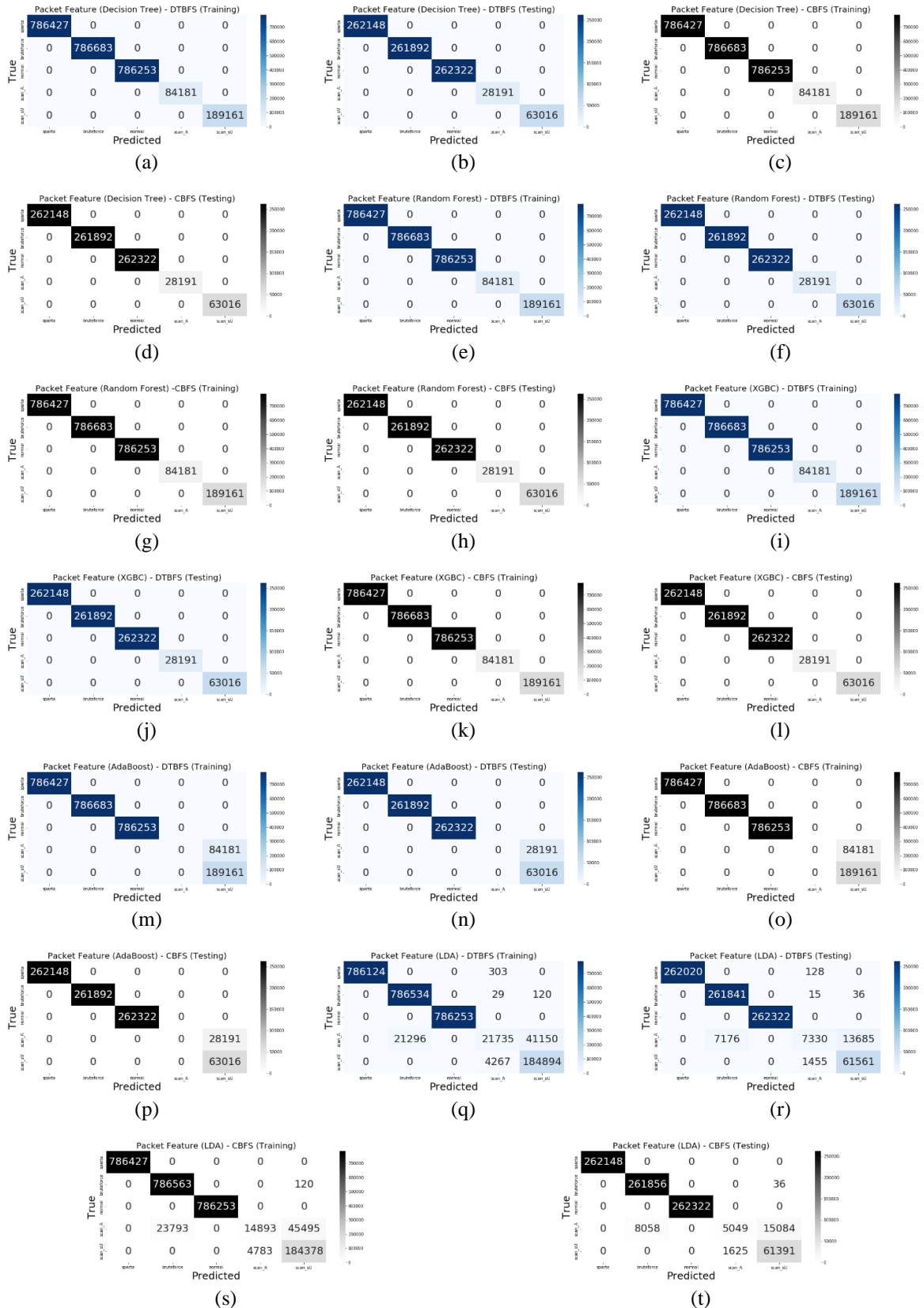


Figure 5. Packet Feature matrix confusion: (a) training DT-DTBFS, (b) testing DT-DTBFS, (c) training DT-CBFS, (d) testing DT-CBFS, (e) training RF-DTBFS, (f) testing RF-DTBFS, (g) training RF-CBFS, (h) testing RF-CBFS, (i) training XGBC-DTBFS, (j) testing XGBC-DTBFS, (k) training XGBC-CBFS, (l) testing XGBC-CBFS, (m) training AB-DTBFS, (n) testing AB-DTBFS, (o) training AB-CBFS, (p) testing AB-CBFS, (q) training LDA-DTBFS, (r) testing LDA-DTBFS, (s) training LDA-CBFS, and (t) testing LDA-CBFS

Figure 6 shows a comparison of the accuracy of various classification algorithms with three different types of features: unidirectional, bidirectional, and package features. Each algorithm was tested using two feature selection methods, namely DTBFS and CBFS. Figure 6 exhibits DT, RF and XGBC classifications have an accuracy value of 100% for each feature extraction and feature selection on the training and testing data, while AB and LDA have an accuracy of 57.599 to 65.536% for unidirectional and bidirectional feature extraction while the packet feature extraction feature is 97.174 to 97.449%.

Table 6 shows the comparison of classification time (in seconds) for various classification algorithms, including DT, RF, XGBC, AB, and LDA. The classification time was measured on three data scenarios (unidirectional, bidirectional, and packet feature) with two feature selection methods, namely DTBFS and CBFS. The execution time of each algorithm in training and testing the model is very important in determining the efficiency and scalability of each approach, especially in the context of IoT data which often has high volume and complexity. The time difference between training and testing also provides insight into the computational requirements and efficiency of each algorithm in various data scenarios.

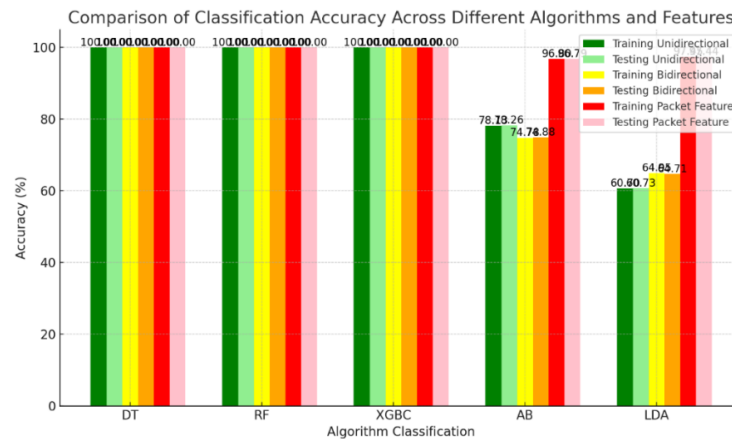


Figure 6. Comparison graph of accuracy values

Table 6. Comparison of classification times (seconds)

Classification	Split Data (75%:25%)	Unidirectional		Bidirectional		Packet Feature	
		DTBFS	CBFS	DTBFS	CBFS	DTBFS	CBFS
DT	Training data	0.0791	0.605	0.499	0.322	6.440	5.977
	Testing data	0.057	0.048	0.052	0.022	0.660	0.449
RF	Training data	22.054	16.204	12.044	6.303	118.455	115.464
	Testing data	3.205	3.196	1.577	1.126	18.642	17.332
XGBC	Training data	10.508	8.749	8.002	5.183	157.975	120.496
	Testing data	0.132	0.118	0.085	0.066	1.131	0.948
AB	Training data	28.080	25.619	17.829	13.543	254.920	229.451
	Testing data	3.075	2.659	1.776	1.516	27.850	24.490
LDA	Training data	1.395	1.068	1.539	0.650	43.043	12.182
	Testing data	0.141	0.103	0.084	0.045	1.848	0.492

Based on Table 6, the processing time of the LDA shows a relatively fast processing time, while AB has a slower processing time, especially for packet features. RF and XGBC require longer training time, especially for packet features, however, CBFS helps reduce their training time. Overall, CBFS is more efficient than DTBFS in terms of training and testing time.

Table 7 presents a comparison of the accuracy values produced by various classification methods on the MQTT-IoT-IDS2020 dataset. This table includes ML and DL-based methods, which are used to detect attacks in IoT networks. Classification is carried out based on three feature approaches, namely packet, unidirectional, and bidirectional. This table also lists the results of several previous referenced studies, and the methods proposed in this study. The highest accuracy for the ML approach was found in the RF and DT classifications, which reached 99.98% on the unidirectional and bidirectional features. On the other hand, DL approaches show varying but still in high accuracy, with DNN-Uniflow and CNN-RNN-LSTM reaching up to 99.54%. The proposed methods in this study using DT, RF, and XGBC algorithms with two feature selection methods, namely DTBFS and CBFS, shows excellent performance, with accuracy reaching 100% almost on all scenarios.

Table 7. Comparison of accuracy values with the MQTT-IoT-IDS2020 dataset

Ref	Method	Classification	Unidirectional (accuracy) (%)	Bidirectional (accuracy) (%)	Packet feature (accuracy) (%)
[39]	ML	LR	98.23	99.44	78.87
[39]	ML	k-NN	99.68	99.90	69.13
[39]	ML	DT	99.96	99.95	88.55
[39]	ML	RF	99.98	99.97	65.39
[39]	ML	SVM (RBF kernel)	97.96	96.61	77
[39]	ML	NB	78.00	97.55	81.15
[39]	ML	SVM (Linear kernel)	82.60	98.50	66.69
[24]	DL	FL	-	-	94.18
[24]	DL	Non-FL	-	-	98.00
[40]	DL	DNN	99.51	99.46	97.03
[40]	DL	CNN-RNN-LSTM	99.51	99.40	98.01
[40]	DL	LSTM	99.54	99.40	97.98
[41]	DL	DNN (Binaryclass)	99.14	99.75	94.94
[41]	DL	DNN (Multiclass)	97.08	98.13	90.80
[42]	ML	k-NN	-	-	97.76
[42]	ML	SVM	-	-	97.80
[42]	ML	NB	-	-	97.58
[42]	ML	RF	-	-	99.98
[42]	ML	DT	-	-	99.98
[42]	ML	SGD	-	-	97.58
[43]	ML	k-NN	-	-	80.82
[43]	ML	LDA	-	-	76.72
[43]	DL	CNN	-	-	80.28
[43]	DL	CNN-LSTM	-	-	98.94
[43]	DL	CNN- k-Fold	-	-	77.68
[43]	DL	CNN-LSTM- k-Fold	-	-	93.22
Proposed	ML	DT-DTBFS	100.00	100.00	100.00
Proposed	ML	DT-CBFS	100.00	100.00	100.00
Proposed	ML	RF-DTBFS	100.00	100.00	100.00
Proposed	ML	RF-CBFS	100.00	100.00	100.00
Proposed	ML	XGBC-DTBFS	100.00	100.00	100.00
Proposed	ML	XGBC-CBFS	100.00	100.00	100.00
Proposed	ML	AB-DTBFS	78.20	74.83	96.80
Proposed	ML	AB-CBFS	78.20	74.83	96.80
Proposed	ML	LDA-DTBFS	60.71	64.83	97.44
Proposed	ML	LDA-CBFS	57.67	65.48	97.18

5. CONCLUSION

In this research, we analyze the effectiveness of various ML techniques to classify attacks using unidirectional, bidirectional, and packet feature extraction. The DT, RF, and XGBC classifications with selection features DTBFS and CBFS, all of which achieved 100% accuracy in unidirectional, bidirectional, and packet feature extraction. In contrast, AB and LDA demonstrated high accuracy with packet features, each approaching 96.80 and 97.44%. Compared with deep learning models, ML models, especially DT and RF, show faster training and testing times while maintaining high accuracy. On the other hand, DTBFS and CBFS feature selection are proved to be effective, showing no significant difference in model's accuracy. Perfect accuracy by some models indicates a potential risk of overfitting. We need to validate this model further on different datasets to confirm its generalizability. ML models exhibit lower computational complexity and faster processing time compared to DL models, so that is suitable for real-time applications. We recommend RF and DT for unidirectional and bidirectional feature extraction due to their high accuracy and efficiency, and AB and LDA for packet feature extraction. For future research, the authors plan to further validate the model with various datasets to ensure the generalizability and endurance of the proposed method. It is important to validate the model with various IoT datasets. Furthermore, future work should consider optimizing model parameters and feature selection to enhance performance and minimize computational requirements in real-world implementations.

ACKNOWLEDGMENTS

The authors would like to express their sincere gratitude to the Directorate General of Higher Education, Research, and Technology, Ministry of Education, Culture, Research, and Technology of the Republic of Indonesia, for the financial support provided through the Doctoral Dissertation Research Grant in 2023. Special thanks are also extended to the Institute of Research and Community Service at Sriwijaya University for their valuable administrative and institutional support throughout this research activity.

FUNDING INFORMATION

This research was funded by the Directorate General of Higher Education, Research, and Technology, Ministry of Education, Culture, Research, and Technology of the Republic of Indonesia through the Doctoral Dissertation Research Grant (Contract Number: 164/E5/PG.02.00.PL/2023, dated June 19, 2023). Additional support was provided by the Institute of Research and Community Service, Sriwijaya University, through Research Contract Number: 0143.11/UN9/SB3.LP2M.PT/2023, dated July 5, 2023.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Zulhipni Reno Saputra	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			✓	
Elsi														
Deris Stiawan	✓	✓		✓		✓		✓	✓	✓	✓	✓		
Bhakti Yudho Suprpto	✓			✓			✓		✓	✓	✓	✓		
M. Agus Syamsul Arifin	✓		✓							✓			✓	
Mohd. Yazid Idris	✓									✓		✓		
Rahmat Budiarto	✓									✓		✓		

C : Conceptualization	I : Investigation	Vi : Visualization
M : Methodology	R : Resources	Su : Supervision
So : Software	D : Data Curation	P : Project administration
Va : Validation	O : Writing - Original Draft	Fu : Funding acquisition
Fo : Formal analysis	E : Writing - Review & Editing	

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

The data that support the findings of this study are openly available in [Lecture Notes in Networks and Systems] at http://doi.org/10.1007/978-3-030-64758-2_6, reference [37].

REFERENCES

[1] A. J. Hintaw, S. Manickam, M. F. Aboalmaaly, and S. Karuppayah, "MQTT Vulnerabilities, Attack Vectors and Solutions in the Internet of Things (IoT)," *IETE Journal of Research*, vol. 69, no. 6, pp. 3368–3397, Aug. 2023, doi: 10.1080/03772063.2021.1912651.

[2] A. Arabiat and M. Altayeb, "Enhancing internet of things security: evaluating machine learning classifiers for attack prediction," *International Journal of Electrical and Computer Engineering*, vol. 14, no. 5, pp. 6036–6046, Oct. 2024, doi: 10.11591/ijece.v14i5.pp6036-6046.

[3] C. Patel and N. Doshi, "'a Novel MQTT Security framework in Generic IoT Model,'" *Procedia Computer Science*, vol. 171, pp. 1399–1408, 2020, doi: 10.1016/j.procs.2020.04.150.

[4] I. Makhdoom, M. Abolhasan, J. Lipman, R. P. Liu, and W. Ni, "Anatomy of Threats to the Internet of Things," *IEEE Communications Surveys and Tutorials*, vol. 21, no. 2, pp. 1636–1675, 2019, doi: 10.1109/COMST.2018.2874978.

[5] F. Chen, Y. Huo, J. Zhu, and D. Fan, "A Review on the Study on MQTT Security Challenge," in *2020 IEEE International Conference on Smart Cloud, SmartCloud 2020*, Nov. 2020, pp. 128–133. doi: 10.1109/SmartCloud49737.2020.00032.

[6] D. Dikii, S. Arustamov, and A. Grishentsev, "DoS attacks detection in MQTT networks," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 21, no. 1, pp. 601–608, 2021, doi: 10.11591/ijeecs.v21.i1.pp601-608.

[7] A. P. Haripriya and K. Kulothungan, "Secure-MQTT: an efficient fuzzy logic-based approach to detect DoS attack in MQTT protocol for internet of things," *Eurasip Journal on Wireless Communications and Networking*, vol. 2019, no. 1, p. 90, Dec. 2019, doi: 10.1186/s13638-019-1402-8.

[8] J. Arshad, M. A. Azad, R. Amad, K. Salah, M. Alazab, and R. Iqbal, "A Review of Performance, Energy and Privacy of Intrusion Detection Systems for IoT," *Electronics*, vol. 9, no. 4, p. 629, Apr. 2020, doi: 10.3390/electronics9040629.

[9] S. Mishra and A. Paul, "A critical analysis of attack detection schemes in IoT and open challenges," in *2020 IEEE International Conference on Computing, Power and Communication Technologies, GUCON 2020*, Oct. 2020, pp. 57–62. doi: 10.1109/GUCON48875.2020.9231077.

[10] D. Javeed, T. Gao, and M. T. Khan, "SDN-enabled hybrid DL-driven framework for the detection of emerging cyber threats in IoT," *Electronics*, vol. 10, no. 8, 2021, doi: 10.3390/electronics10080918.




[11] A. Dasgupta et al., "Exploring Unsupervised Learning Methods for Automated Protocol Analysis," *2021 IEEE Symposium Series on Computational Intelligence, SSCI 2021 - Proceedings*, 2021, doi: 10.1109/SSCI50451.2021.9660173.

[12] A. F. Gentile, D. Macri, D. L. Carni, E. Greco, and F. Lamonaca, "A network performance analysis of MQTT security protocols with constrained hardware in the dark Net for DMS," *Applied Sciences*, vol. 14, no. 18, 2024, doi: 10.3390/app14188501.




- [13] R. Doriguzzi-Corin, L. A. D. Knob, L. Mendozzi, D. Siracusa, and M. Savi, "Introducing packet-level analysis in programmable data planes to advance Network Intrusion Detection," *Computer Networks*, vol. 239, 2024, doi: 10.1016/j.comnet.2023.110162.
- [14] S. Choi and J. Cho, "Novel Feature Extraction Method for Detecting Malicious MQTT Traffic Using Seq2Seq," *Applied Sciences*, vol. 12, no. 23, Dec. 2022, doi: 10.3390/app122312306.
- [15] X. Li, M. Xu, P. Vijayakumar, N. Kumar, and X. Liu, "Detection of Low-Frequency and Multi-Stage Attacks in Industrial Internet of Things," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 8, pp. 8820–8831, Aug. 2020, doi: 10.1109/TVT.2020.2995133.
- [16] M. Bykova, S. Ostermann, and B. Tjaden, "Detecting network intrusions via a statistical analysis of network packet characteristics," in *Proceedings of the Annual Southeastern Symposium on System Theory*, 2001, vol. 2001-Janua, pp. 309–314, doi: 10.1109/SSST.2001.918537.
- [17] A. Bhandari, S. Gautam, T. K. Koirala, and M. Ruhul Islam, "Packet sniffing and network traffic analysis using TCP—A new approach," in *Lecture Notes in Electrical Engineering*, vol. 443, 2018, pp. 273–280, doi: 10.1007/978-981-10-4765-7_28.
- [18] R. Alasmari and A. A. Alhogail, "Protecting Smart-Home IoT Devices From MQTT Attacks: An Empirical Study of ML-Based IDS," *IEEE Access*, vol. 12, pp. 25993–26004, 2024, doi: 10.1109/ACCESS.2024.3367113.
- [19] H. F. Aliabdi, "A Hybrid Method for Intrusion Detection in the IoT," *International journal of Web Research*, vol. 5, no. 2, pp. 55–60, 2022.
- [20] J. Liu, Y. Tang, H. Zhao, X. Wang, F. Li, and J. Zhang, "CPS Attack Detection under Limited Local Information in Cyber Security: An Ensemble Multi-Node Multi-Class Classification Approach," *ACM Transactions on Sensor Networks*, vol. 20, no. 2, pp. 1–27, Mar. 2024, doi: 10.1145/3585520.
- [21] X. Chen, P. Wang, Y. Yang, and M. Liu, "Resource-Constraint Deep Forest-Based Intrusion Detection Method in Internet of Things for Consumer Electronic," *IEEE Transactions on Consumer Electronics*, vol. 70, no. 2, pp. 4976–4987, May 2024, doi: 10.1109/TCE.2024.3373126.
- [22] M. B. Gorzalczany and F. Rudzinski, "Intrusion Detection in Internet of Things With MQTT Protocol - An Accurate and Interpretable Genetic-Fuzzy Rule-Based Solution," *IEEE Internet of Things Journal*, vol. 9, no. 24, pp. 24843–24855, Dec. 2022, doi: 10.1109/JIOT.2022.3194837.
- [23] R. Chaganti, V. Ravi, and T. D. Pham, "Deep learning based cross architecture internet of things malware detection and classification," *Computers and Security*, vol. 120, Sep. 2022, doi: 10.1016/j.cose.2022.102779.
- [24] D. C. Attota, V. Mothukuri, R. M. Parizi, and S. Pouriyeh, "An Ensemble Multi-View Federated Learning Intrusion Detection for IoT," *IEEE Access*, vol. 9, pp. 117734–117745, 2021, doi: 10.1109/ACCESS.2021.3107337.
- [25] X. Liu *et al.*, "Attention-based bidirectional GRU networks for efficient HTTPS traffic classification," *Information Sciences*, vol. 541, pp. 297–315, Dec. 2020, doi: 10.1016/j.ins.2020.05.035.
- [26] A. Samy, H. Yu, and H. Zhang, "Fog-Based Attack Detection Framework for Internet of Things Using Deep Learning," *IEEE Access*, vol. 8, pp. 74571–74585, 2020, doi: 10.1109/ACCESS.2020.2988854.
- [27] S. Huang, K. Chen, C. Liu, A. Liang, and H. Guan, "A statistical-feature-based approach to internet traffic classification using machine learning," in *2009 International Conference on Ultra Modern Telecommunications and Workshops*, Oct. 2009, pp. 1–6, doi: 10.1109/ICUMT.2009.5345539.
- [28] S. H. S. Ariffin, N. H. Mustaffa, F. Dewanta, I. W. Hamzah, M. A. Baharudin, and N. H. Abdul Wahab, "Hybrid Feature Selection Based Lightweight Network Intrusion Detection System for MQTT Protocol," in *International Conference on Software, Knowledge Information, Industrial Management and Applications, SKIMA*, Dec. 2023, pp. 226–230, doi: 10.1109/SKIMA59232.2023.10387337.
- [29] T. T. Jui, M. N. Hoq, S. Majumdar, and M. S. Hossain, "Feature Reduction through Data Preprocessing for Intrusion Detection in IoT Networks," in *Proceedings - 2021 3rd IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications, TPS-ISA 2021*, Dec. 2021, pp. 41–50, doi: 10.1109/TPSISA52974.2021.00005.
- [30] H. S. Ilango, M. Ma, and R. Su, "A FeedForward-Convolutional Neural Network to Detect Low-Rate DoS in IoT," *Engineering Applications of Artificial Intelligence*, vol. 114, Sep. 2022, doi: 10.1016/j.engappai.2022.105059.
- [31] J. Li, M. S. Othman, H. Chen, and L. M. Yusuf, "Optimizing IoT intrusion detection system: feature selection versus feature extraction in machine learning," *Journal of Big Data*, vol. 11, no. 1, 2024, doi: 10.1186/s40537-024-00892-y.
- [32] C. Zhang *et al.*, "A Novel Framework Design of Network Intrusion Detection Based on Machine Learning Techniques," *Security and Communication Networks*, vol. 2021, pp. 1–15, Jan. 2021, doi: 10.1155/2021/6610675.
- [33] M. Ahmad, Q. Riaz, M. Zeeshan, H. Tahir, S. A. Haider, and M. S. Khan, "Intrusion detection in internet of things using supervised machine learning based on application and transport layer features using UNSW-NB15 data-set," *Eurasip Journal on Wireless Communications and Networking*, vol. 2021, no. 1, p. 10, Dec. 2021, doi: 10.1186/s13638-021-01893-8.
- [34] Z. Liu, Y. Wang, F. Feng, Y. Liu, Z. Li, and Y. Shan, "A DDoS Detection Method Based on Feature Engineering and Machine Learning in Software-Defined Networks," *Sensors*, vol. 23, no. 13, p. 6176, Jul. 2023, doi: 10.3390/s23136176.
- [35] T. D. Diwan *et al.*, "Feature Entropy Estimation (FEE) for Malicious IoT Traffic and Detection Using Machine Learning," *Mobile Information Systems*, vol. 2021, pp. 1–13, Dec. 2021, doi: 10.1155/2021/8091363.
- [36] H. Tyagi and R. Kumar, "Attack and anomaly detection in IoT networks using supervised machine learning approaches," *Revue d'Intelligence Artificielle*, vol. 35, no. 1, pp. 11–21, 2021, doi: 10.18280/ria.350102.
- [37] H. Hindy, C. Tachtatzis, R. Atkinson, E. Bayne, and X. Bellekens, "MQTT-IoT-IDS2020: MQTT Internet of Things Intrusion Detection Dataset," *IEEE Dataport*, 2020, doi: 10.0.82.235/-ep04.
- [38] Y. Zhang and Z. Wang, "Feature Engineering and Model Optimization Based Classification Method for Network Intrusion Detection," *Applied Sciences*, vol. 13, no. 16, p. 9363, Aug. 2023, doi: 10.3390/app13169363.
- [39] H. Hindy, E. Bayne, M. Bures, R. Atkinson, C. Tachtatzis, and X. Bellekens, "Machine Learning Based IoT Intrusion Detection System: An MQTT Case Study (MQTT-IoT-IDS2020 Dataset)," *Lecture Notes in Networks and Systems*, vol. 180, pp. 73–84, 2021, doi: 10.1007/978-3-030-64758-2_6.
- [40] F. Mosaiyebzadeh, L. G. Araujo Rodriguez, D. Macedo Batista, and R. Hirata, "A Network Intrusion Detection System using Deep Learning against MQTT Attacks in IoT," in *Proceedings - 2021 IEEE Latin-American Conference on Communications, LATINCOM 2021*, Nov. 2021, pp. 1–6, doi: 10.1109/LATINCOM53176.2021.9647850.
- [41] M. A. Khan *et al.*, "A deep learning-based intrusion detection system for mqtt enabled iot," *Sensors*, vol. 21, no. 21, p. 7016, Oct. 2021, doi: 10.3390/s21217016.
- [42] N. Saran and N. Kesswani, "A comparative study of supervised Machine Learning classifiers for Intrusion Detection in Internet of Things," *Procedia Computer Science*, vol. 218, pp. 2049–2057, 2022, doi: 10.1016/j.procs.2023.01.181.
- [43] A. Alzahrani and T. H. H. Aldhyani, "Artificial Intelligence Algorithms for Detecting and Classifying MQTT Protocol Internet of Things Attacks," *Electronics*, vol. 11, no. 22, p. 3837, Nov. 2022, doi: 10.3390/electronics11223837.

BIOGRAPHIES OF AUTHORS






Zulhipni Reno Saputra Elsi    received a doctorate degree from the Faculty of Engineering, Sriwijaya University. Currently serves as a lecturer at the Faculty of Engineering, Muhammadiyah University, Palembang, Indonesia. His research interests include internet of things, computer networks, information security, and intrusion detection systems. He can be contacted at email: Zulhipni_renosaputra@um-palembang.ac.id.






Deris Stiawan    received a Ph.D. degree in Computer Engineering from Universiti Teknologi Malaysia, Malaysia. He is currently a full Professor at Department of Computer Engineering, Faculty of Computer Science, Sriwijaya University. His research interests include computer network, intrusion detection/prevention system, and heterogeneous network. He can be contacted at email: deris@unsri.ac.id.






Bhakti Yudho Suprpto    is an Associate Professor in the Department of Electrical at the Faculty of Engineering, Sriwijaya University, Indonesia. He obtained a Doctorate Degree in Electrical Engineering from Universitas Indonesia. His professional profile has derived to robotic and control, which focused on, fuzzy logic, and neural network. His research interests include control system. He can be contacted at email: bhakti@ft.unsri.ac.id.






M. Agus Syamsul Arifin    received a doctorate degree from the Faculty of Engineering, Sriwijaya University. Currently he serves as a lecture at the Department of Informatic, Faculty of Engineer Universitas Jenderal Soedirman, Indonesia. His research interests include computer networks, IoT, information security, and intrusion detection systems. He can be contacted at email: mas.agus1988@gmail.com.



Mohd Yazid Idris    is an Associate Professor at Faculty of Computing, Universiti Teknologi Malaysia. He obtained his M.Sc. and Ph.D. in the area of Software Engineering, and Information Technology (IT) Security in 1998 and 2008, respectively. In software engineering, he focuses on the research of designing and development of mobile and telecommunication software. His main research activity in IT security is in the area of intrusion prevention and detection (IPD). He can be contacted at email: yazid@utm.my.



Rahmat Budiarto    received B.Sc. degree from Bandung Institute of Technology in 1986, M.Eng. and Dr. Eng. in Computer Science from Nagoya Institute of Technology in 1995 and 1998, respectively. Currently, he is a full Professor at College of Computing and Information, Albaha University, Saudi Arabia. His research interests include intelligent systems, brain modeling, IPv6, network security, wireless sensor networks, and MANETs. He can be contacted at email: rahmat@bu.edu.sa.