IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 28, 2021, accepted August 9, 2021, date of publication August 17, 2021, date of current version August 27, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3105517

Ping Flood Attack Pattern Recognition Using a K-Means Algorithm in an Internet of Things (IoT) Network

DERIS STIAWAN[1], MEILINDA EKA SURYANI[1], SUSANTO[2,3], MOHD YAZID IDRIS[4], MUAWYA N. ALDALAIEN[5], NIZAR ALSHARIF[6], AND RAHMAT BUDIARTO[6]

[1]Department of Computer Engineering, Faculty of Computer Science, Universitas Sriwijaya, Palembang 30119, Indonesia
[2]Faculty of Engineering, Universitas Sriwijaya, Palembang 30119, Indonesia
[3]Faculty of Computer Science, Universitas Bina Insan, Lubuk Linggau 31626, Indonesia
[4]School of Computing, Faculty of Engineering, Universiti Teknologi Malaysia, Johor 81310, Malaysia
[5]King Hussein School of Computing Sciences, Princess Sumaya University for Technology, Amman 11941, Jordan
[6]College of Computer Science and Information Technology, Al-Baha University, Al Bahah 65731, Saudi Arabia

Corresponding authors: Deris Stiawan (deris@unsri.ac.id) and Rahmat Budiarto (rahmat@bu.edu.sa)

ABSTRACT Security is the main challenge in Internet of Things (IoT) systems. The devices on the IoT networks are very heterogeneous, many of them have limited resources, and they are connected globally, which makes the IoT much more challenging to secure than other types of networks. Denial of service (DoS) is the most popular method used to attack IoT networks, either by flooding services or crashing services. Intrusion detection system (IDS) is one of countermeasures for DoS attack. Unfortunately, the existing IDSs are still suffering from detection accuracy problem due to difficulty of recognizing features of the DoS attacks. Thus, we need to determine specific features that representing well the traffic attacks, so the IDS will be able to distinguish normal traffic from the attacks. In this work, we investigate ping flood attack pattern recognition on IoT networks. Experiments were conducted using wireless communication with three different scenarios: normal traffic, attack traffic, and combined normal-attack traffic. Each scenario created an associated dataset. The datasets were then grouped into two clusters: normal and attack. The K-Means algorithm was used to produce the clustering results. The average number of packets in the attack cluster was 95 931 packets, and the average in the normal cluster was 4,068 packets. The accuracy level of the clustering results was calculated using a confusion matrix. The accuracy of the clustering using the implemented K-Means algorithm was 99.94%. The rates from the confusion matrix were true negative (98.62%), true positive (100.00%), false negative (0.00%), and false positive (1.38%).

INDEX TERMS Internet of Things (IoT), pattern recognition, ping flood, K-means, clustering.

I. INTRODUCTION

The Internet of Things (IoT) is a computing concept in which physical objects connected to the Internet are able to identify themselves and communicate with other devices in the network. In other words, the IoT provides a giant interconnected network of devices (''things'') that can serve any purpose imagined by their creators [1]. The IoT has been a major research topic for almost a decade, and its use has been growing rapidly. The IoT is a hybrid network of small—usually wireless sensor network (WSN)—devices and conventional Internet network components. Unlike the conventional Internet, in which the devices are more homogeneous and powerful, the nodes (''things'') in the IoT are more heterogeneous devices and have limited resources. An IoT device could be a light bulb, microwave, car part, smartphone, PC/laptop, powerful server machine, or cloud component [2], [3].

Many successful applications of IoT ecosystem have been developed, such as in surveillance [4] and smart cities [5]. IoT network stability also becomes a vital need due to recent developments and the explosion of IoT applications, as well as its prevalence in multiple security-sensitive scenarios. Traditional Internet routing protocols are ineffective for IoT devices with limited resources [6]. Hence, various solutions

The associate editor coordinating the review of this manuscript and approving it for publication was Burak Kantarci.

are introduced for routing in Pv6 over Low -Power Wireless Personal Area Networks (6LoWPAN). Triantafyllou *et al.* [7] addressed existing network protocols, systems, and technology used in IoT networks and applications. In the majority of IoT networks, routing is performed using the routing protocol for Low-Power and Lossy Networks (RPL). RPL is an IPv6 routing protocol that is standardized for the IoT by Internet Engineering Task Force (IETF). It has the benefits of secure modes availability, energy efficiency, and flexibility to operate in various environments.

Existing RPL attacks are categorized into WSN-inherited and RPL-specific attacks. WSN-inherited attacks include: Blackhole attack, Greyhole or Selective Forward attack, Sinkhole attack, Wormhole attack, Sybil attack and Hello Flood attack. Whereas RPL-specific attacks include: Rank attack, Version Attack, Local Repair attack, Neighbor attack, Replay attack, DIS attack and RPL attacks in Storing mode. Some works have been carried out on securing RPL protocol against DoS/DDoS attacks [8], [9] and mitigating forwarding misbehaviors in RPL-based low power and lossy networks [10], [11].

Nevertheless, as devices on IoT networks are very heterogeneous, they may run different routing protocols instead of RPL. Thus, a generic IDS as counter measure for multiple routing protocols in IoT networks is required and may be developed through exploration of network traffic behavior.

There is uncertainty in the security triad (confidentiality, integrity, and availability) of an IoT network, which may affect future development [12]. One of the main goals of IoT security is to provide data to users whenever needed. Data availability requires direct access for users to the source of information; therefore, there is also a need to provide security to overcome any potential attacks toward an IoT network, such as in denial of service (DoS) attacks [13]. A DoS attack is a malicious way for an individual to consume resources such as a user's bandwidth [14], [15]. Generally, a DoS attack involves greatly increasing network traffic with a large number of false or unnecessary messages. One of the most common DoS attacks is an Internet control message protocol (ICMP) flood or ping flood [14]. Besides, recent analysis of 10 IoT devices found 250 vulnerabilities, including an open telnet port, outdated Linux firmware, and un-encrypted sensitive data transmission [16]. Furthermore, due to the heterogeneous of IoT devices where many of them have limited resources, and they are connected globally, makes the IoT networks much more challenging to secure than other types of networks. Therefore, security becomes big challenge in IoT networks [17], [18].

Existing intrusion detection systems (IDSs) for wireless sensor networks (WSNs) may not be suitable to protect IoT devices, because these IDSs were primarily developed for more restricted devices and have not considered the specific needs of the IoT [3]. Hence, this paper attempts to support appropriate IDS for the IoT networks by differentiating ping flood traffic from normal traffic. The proposed IDS uses the K-Means algorithm to identify DoS attacks on IoT networks, specifically the ping flood attack. The performance of the clustering tool is investigated by considering the resulting confusion matrix.

There are two general methods of DoS attacks: flooding services or crashing services. Flood attacks occur when the system receives too much traffic for the server to buffer, causing them to slow down and eventually stop [13], [14]. Generally, a DoS attack involves greatly increasing network traffic with a large number of false or unnecessary messages. One of the most common DoS attacks is an Internet control message protocol (ICMP) flood or ping flood [13]. This research work focuses on ICMP (ping) flooding attack because it overwhelms the targeted device's network connections with bogus traffic and causes legitimate requests are prevented from getting through.

This research work possesses various contributions in the domain of intrusion detection for IoT networks.

1) First, a dataset generated from real traffic testbed and combined with malware traffic was created as an alternative for benchmarking anomaly detection systems for IoT networks.
2) Second, significant and relevant features of ping flood attack in IoT networks.
3) Lastly, the selected features are used for anomaly detection system. The model will help IoT networks to defend against ping flood attacks.

The rest of the paper is arranged as follows. Section II provides the theoretical background and related works. Section III discusses the research methodology. Section IV discusses the experimental results and analysis, and, finally, Section V concludes the work.

II. THEORETICAL BACKGROUND AND RELATED WORK

A. INTERNET CONTROL MESSAGE PROTOCOL (ICMP) FLOOD

The ICMP is usually used by network technology to diagnose or to report errors; however, now this protocol is often used by an individual to attack a victim by sending large amounts of messages [19], [20]. The first two fields of an ICMP packet header determine whether the packet is a request message or an error message. An ICMP error message is not automatically sent as a response to an ICMP error. When an ICMP error message is sent, it will include the Internet protocol (IP) header and datagram that caused the error so that the receiver can associate the error with the process. When type 0 (echo reply) is sent, the reply is no longer type 8 (echo request). The last field in the ICMP format is a checksum. This field is used for error checking. Before an ICMP message is transmitted, a checksum is calculated and inserted into the field. On the recipient side, the checksum is calculated again and verified with the checksum field. If there is a problem with the checksum or other information, this indicates some mistake or a false message [19]. The evolution of this attack and defenses against it involve finding different ways to create large amounts of seemingly valid messages countered by

methods to identify and eliminate invalid messages and their sources. Some examples of ICMP flood attacks are smurf attacks, a ping flood, and the ping of death.

B. K-MEANS CLUSTERING ALGORITHM

One of the most studied clustering algorithms is K-Means [19]–[22]. The algorithm reduces the total of the different intra-clusters. Its simplicity and agility have made this a well-known algorithm for clustering in different fields of knowledge. The K-Means algorithm reduces the total of different intra-cluster. Simplicity and agility has made this algorithm as a famous way to do clustering in different fields of knowledge.

K-Means is a method with a centroid model. A centroid is the midpoint of a cluster that contains a value. The centroid is also used to distance of object data. Object data can be placed within a cluster if it is closest to the centroid of that cluster [23]–[25].

A general rule for finding the optimal K partitions locally is by moving a point of Cluster 1 to another cluster [21], [26]. K-Means distributes all the objects to K clusters randomly by the following procedure [27];

1) Compute an average value for each cluster and use this average value to represent the cluster;
2) Redistribute objects to the nearest cluster according to their distance to the center of the cluster;
3) Update the cluster average value by calculating the average value of the objects in each cluster;
4) Calculate a criterion function until the criteria function meets.

C. DoS/DDoS ATTACK AND ITS COUNTER MEASURES IN IoT NETWORKS

This section discusses DoS/DDoS attack and its countermeasures in IoT environment, and then at the end of this section, we classify them.

Research work by Pu *et al.* [28] recognizes malicious node's pattern behavior of DoS attack that causes energy harvesting. The researchers determine series of scenarios, analyze and identify vulnerable cases. A system called EYES was introduced to detect malicious nodes and immediately isolated from the network. The detection rate of the proposed system reaches 70 to 92% accuracy.

Further research by Shukla [29] discusses wormhole attack detection on IoT networks. This research applies IDSs with three different approaches: IDS with K-Means, IDS with a decision tree, and a hybrid IDS (K-Means and decision tree). Among the three, the IDS that used the K-Means approach had a range with the highest detection rate, 70–93%. The IDS with the decision tree had a detection rate range of 71–80%, and the IDS with the hybrid approach had a range of 71–75%.

Machine learning-based solutions of IDS systems for IoT environment have been introduced [30], [38]. Qiu *et al.* [30] have proposed network intrusion detection system (NIDS) for IoT environment using deep learning AutoEncoder technique. The proposed NIDS was able to achieve 94% accuracy in detecting DoS/DDoS attacks. While Bovenzi *et al.* [31] introduced two-stage hierarchical Network Intrusion Detection approach, i.e.: (i) anomaly detection using a novel lightweight solution based on a Multi-Modal Deep AutoEncoder (M2-DAE), and (ii) attack classification, using soft-output classifiers. The proposed approach has successfully detected known attacks' patterns, such as: DDoS, DoS, Scan, Theft, and unknown patterns in IoT networks.

In other research, Fadlil *et al.* [35] discussed an analysis of threats on an IoT network. The work utilizes an offline IDS to collect and analyze information from a variety of IoT networks, as well as to identify DoS attacks on them. Hussain *et al.* [36] have carried out experiments on network attack detection through recognition of DDoS attack patterns using Naïve Bayes method. De Lima Filho *et al.* [37] have performed analysis on DoS/DDoS attack pattern recognition in IoT environment using ResNet. Non-images network traffic data were converted into image data. The DoS/DDoS attacks are detected using Convolutional Neural Network (CNN) model. Experimental results show high accuracy level.

Moreover, other researchers use different approaches in recognizing network attacks patterns. Li *et al.* [38] developed monitoring system called sFlow, consists of collectors and agents (embedded in router, switch, or independent probes). The architecture of the system is designed for recognizing normal network traffic pattern and several types of DoS attacks, and then classified using machine learning algorithms. Kumar *et al.* [39] introduced acknowledgment-based approach for DoS attack patterns recognition and investigated its effect. While Wedel *et al.* [40] used a cryptography technique and blockchain in detecting DDoS/DoS anomaly patterns on IoT networks with a combination of machine learning algorithms XGBOST and Random Forest. Table 1 briefly summarizes researches on attacks' patterns on IoT networks.

III. METHODOLOGY

This work followed the standard steps of a research methodology. A design topology and scenario were described, followed by experiments conducted to create a dataset. Ping flood attack pattern extraction was next performed, using feature extraction method. The ping flood attack pattern recognition was conducted using the K-Means clustering method. Lastly, the findings were analyzed.

A. IoT NETWORK TOPOLOGY

The IoT network topology shown in Figure 1 consists of a monitoring server, WiFi router, two units of middleware for two nodes with the Zigbee protocol, and four nodes with the WiFi protocol, along with various sensors (DHT11, DHT22, MQ2, soil moisture, and water level).

Node 1 is a WeMos D1 WiFi interface (hereafter, WeMos) with a DHT22 temperature sensor. Node 2 is a WeMos with a soil moisture sensor. Node 3 is a WeMos with an MQ2 gas sensor. Node 4 is a WeMos with a water level sensor. Node 5 is a Xbee wireless interface (hereafter, Xbee)

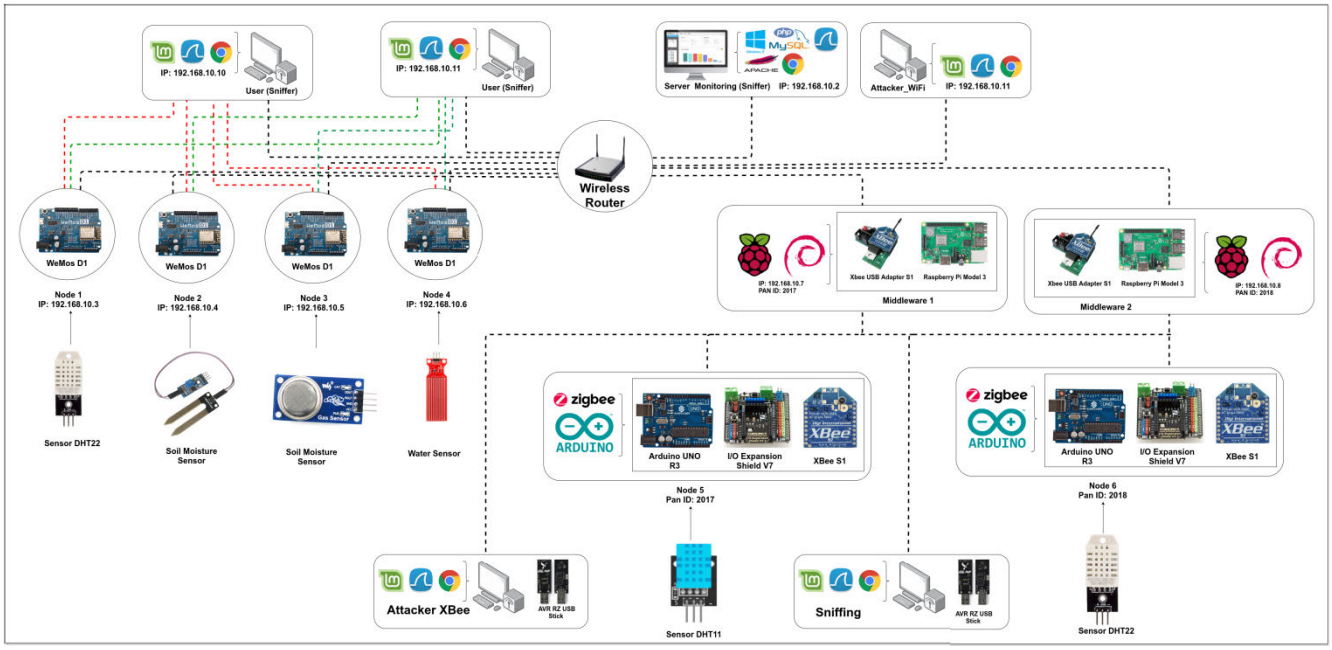

FIGURE 1. IoT network topology for creating the dataset.

TABLE 1. Summary of related works on DoS/DDoS attack pattern.

Ref. & (Year)	Experiment	Method	Pros/Cons
[34] (2016)	Detecting normal and non-normal (DoS/DDoS attacks) traffic patterns.	ANN	Detection accuracy level more than 99%; the highest accuracy level was 99.99%. Having good TPR and FPR.
[36] (2020)	Detecting DoS/DDoS attacks traffic patterns	CNN	Accuracy level reaches 99.99%
[31] (2020)	Identifying normal and non-normal (DoS/DDoS attacks) traffic patterns.	M2-DAE	FPR level was 1%; TPR 99%; Retraining proses was limited, based only on anomaly traffics.
[35] (2017)	Recognizing DDoS attack patterns	Naïve Bayes	Ability to predict fast and with high accuracy
[38] (2017)	Identifying DoS attack patterns & investigating its effects	Acknowledgment-based	Can be used for preparing strategy on counter attack
[37] (2019)	Recognizing normal network traffic patterns and some types of DoS attacks	Monitoring based + Machine Learning	Still need improvement to optimize detection accuracy level
[39] (2021)	Detecting anomaly and DDoS attack patterns	Crytography (Blockchain) + XGBOST and RF	Very effective in detecting various recent attacks including DDoS/DoS

and DHT11 temperature and humidity sensor. Node 6 is an Xbee and DHT22 temperature sensor. The server and nodes have middleware connected via a wireless router running the dynamic host configuration protocol (DHCP).

B. DATASET CREATION SCENARIO

The creation of the dataset used three scenarios: normal data retrieval, attack data retrieval, and a combination of normal and attack data retrieval. The data retrieval in the scenarios aimed to compare the normal, attack, and combined normal-attack data. The scenarios were as follows:

1) Normal data retrieval was performed on a system that runs without any attacks in which the sensor data from all the nodes can be accepted by the server. The user sends a ping to Node 1 through Node 4 using four terminals; each terminal sends a ping to one node when the data is retrieved. The experiments were repeated several times, with each experiment taking duration of five minutes.
2) Attack data retrieval was performed in experiments with durations of five minutes. Data was taken from a network that experienced a DoS attack in the form of a ping flood from an individual with an IP address of 192.168.1.11. The attacker performs flooding against Node 1 in the first minute. At the second minute, attacks on Node 1 were stopped; however, the attack continued on Node 2. At the third minute, the attack switched to Node 3. The attack targets Node 4 at the fourth minute. In the final minute, the attack is sent to Nodes 1 through 4 simultaneously using four terminals.
3) As for the combined normal-attack data retrieval, as expected, it combines elements of the previous two data retrieval scenarios. In essence, the previous two scenarios are run at the same time. That is, the user sends a ping to Nodes 1 through 4 using four terminals (as in the first scenario), and at the same time, the attack

on Node 1 begins (as in the second scenario). The duration for this data retrieval scenario is five minutes with the previously described actions for both scenarios occurring at the appropriate times.

C. FEATURE EXTRACTION

Wireshark packet analyzer software was used to extract attributes of the obtained datasets for the three scenarios. This information was stored in a file with a comma-separated values (CSV) format. The feature extraction was done through observation on the occurrences/statistics of the traffic attributes. We then rank them and determine the significant and relevant features that representing attack patterns. Thus, no feature extraction algorithms are used in this research work. In addition, this work also used the Snort software as a tool for validation analysis. Figure 2 shows the flowchart of the feature extraction process that was used with the datasets.

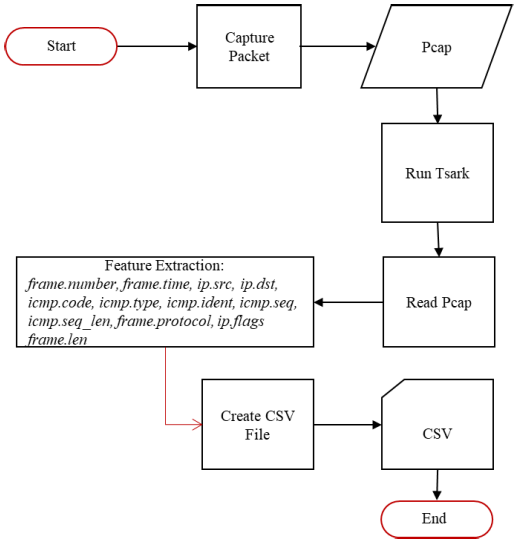

FIGURE 2. Feature extraction flowchart.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

A. THE CREATED DATASET

The created datasets had the following characteristics. The biggest dataset was for the combined scenario, which had a size of 93 MB and contained 1 662 966 packets (95% were ICMP packets); the smallest dataset was for the normal scenario. It had a size of 2.68 MB with a total of 19 639 packets. In the normal scenario dataset, the most common traffic was transmission control protocol (TCP) packets (38%). The dataset was made available as public dataset in the following URL: https://zenodo.org/record/4436208. Table 2 shows the data packet statistics for all three data retrieval scenarios.

B. PING FLOOD ATTACK PATTERN ANALYSIS

The ping flood attack pattern can be found by performing the following analysis:

TABLE 2. Data packet statistics.

	Normal	Attack	Normal-Attack
Size	2.68 MB	89.3 MB	93 MB
ICMP	2,440	1,534,886	1,577,730
TCP	8,506	2,603	1,222
LLC	6,997	54,782	82,118
UDP	1,007	1,334	935
ARP	743	1,580	961
Total	19,693	1,595,315	1,662,966

1) Attack patterns are derived by identifying the extracted feature attributes that have the same value in every data packet in the normal dataset but have different values in the attack dataset.
2) Based on the correlation between the results of feature extraction and the raw data, as well as the validation using the Snort software, the obtained unique attributes can be identified as an attack pattern.

Based on the analysis and the attributes that were extracted, the gained unique attributes of the ping flood attack pattern are shown in Table 3. Attributes that can be used in a ping flood attack pattern are a frame length of the frame header with a value of 42 and IP flags from the IP header with values of 0×00.

TABLE 3. Ping flood attack pattern on IoT.

Type	Attribute	
	Frame length	Flags
ICMP Normal	98	0x02
Ping Flood	42	0x00

C. VALIDATION ANALYSIS USING SNORT

As a basis to demonstrate the existence of a ping flood attack in the attack and combined datasets, experiments needed to be conducted using the Snort software. The experiments used the default rules of snort.

To perform the validation analysis with the Snort alerts and the feature extraction results, a manual correlation procedure was used. Figure 3 illustrates the identification of a correlation between a Snort alert and a feature extraction result from the attack dataset.

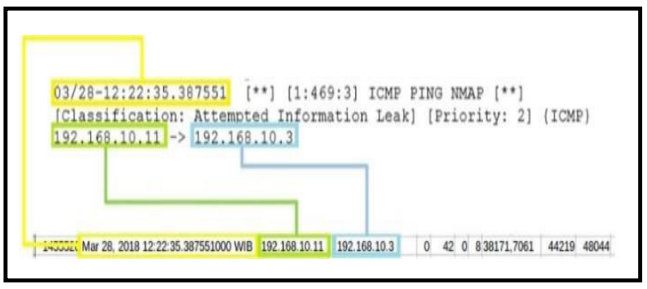

FIGURE 3. Correlation of snort alert and feature extraction.

The upper part is the Snort alert, and the last line is the feature extraction result obtained, as indicated by the

TABLE 4. Data cluster.

Data	Cluster	
	Normal	Attack
Normal	4,427	95,573
Attack	3,710	96,290

flowchart in Figure 2. Figure 3 shows a match between the alert from Snort and the feature extraction result. According to the both, there is an attack launched by a node with an IP address of 192.168.10.11 against a node with an IP address of 192.168.10.3 at time 12:22:35.387551; according to the Snort alert, the priority value of this attack was 2, which means that the attack was of medium severity.

D. IMPLEMENTATION OF THE CLASSIFICATION METHOD USING K-MEANS ALGORITHM

Before performing the classification, data normalization was performed by eliminating attributes that would not be used in the classification process. In this case, the attributes that would be ignored in the classification process were the following: *frame.number, frame.time, ip.src, ip.dst, icmp.code, icmp.type, icmp.ident, icmp.seq, icmp.seq_len,* and *frame.protocol.* As for features/attributes that would be used, these were the features of the attack pattern that had been previously obtained, i.e.: *frame.len* and *ip.flags*.

The stages of the K-Means algorithm in working with the dataset were as follows:

1) DETERMINE THE NUMBER OF CLUSTERS At this stage, two clusters were created to classify the normal and attack data packets. The clustering results of the K-Means algorithm and of the Random Tree both use Cluster 0 and Cluster 1.
2) DETERMINE THE INITIAL CENTROID The next step was to determine the midpoint (centroid) of each cluster randomly.
3) EUCLIDEAN DISTANCE CALCULATION The distance between the packets (cluster members) and the centroid of each cluster were calculated using a distance formula (Euclidean distance). This stage allows an object to move to a different cluster, depending on these distances. These calculations determine the cluster identity of each data packet. If the distance of a data packet is closer to the centroid of Cluster 0 compared with the centroid distance of Cluster 1, then the data packet is a member of Cluster 0, and vice versa.
4) ITERATE The next step was to iterate the clustering of the data packets based on the closest distance. In other words, the closest centroid is chosen. This iteration process was performed repeatedly until a condition was obtained in which there were neither changes nor cluster moves.

The extracted attributes and data analysis provide cluster data of ping flood attacks patterns as shown in Table 4.

According to Wedel and Kamakura [40], the minimum of data size for K-Means classification is 2^M, where M is the

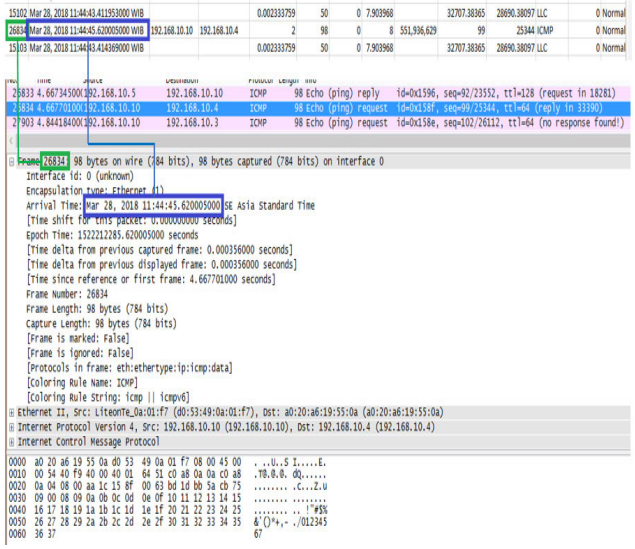

FIGURE 4. Normal data validation.

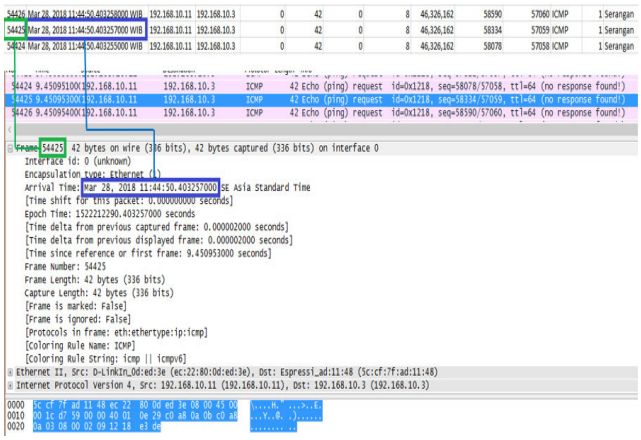

FIGURE 5. Attack data validation.

number of features in the data. Feature extraction process provides 12 features; hence, the minimum data size is 4096. Thus, 1 662 966 packets data of the created dataset is more than enough to support K-Means works properly.

E. VALIDATION OF RESULT OF RECOGNITION METHOD AGAINST RAW DATA

This validation process analyzed the correlation between the CSV file, which was output from the K-Means clustering process, and the captured packets file or raw data in the form of pcap files from the Wireshark software. The correlations were between both types of data in the combined normal-attack (third scenario) dataset.

Figure 4 and Figure 5 show the normal and attack data validation, respectively, in the form of correlations between the clustering results data and the raw data captured by the Wireshark software. The parameters in this validation were the number of the packet and its timestamp. Figure 4 shows

the information obtained from a normal packet in the dataset, which had a frame number of 26834 and was captured on March 28, 2018 at 11:44:45.620005000. Figure 5 shows a packet from the attack data. The information in Figure 5 indicates that the attack packet had a frame number of 54425 and was recorded on March 28, 2018 AT 11:44:50.403257000. The clustering result data and the raw data contained the same information on the packet size and timestamp parameters. Thus, it can be said that the results of the data capture were valid.

F. CLASSIFICATION USING RANDOM TREE ALGORITHM

Classification using the random tree was conducted using the same datasets. These were the attack and combined attack-normal datasets, which had been extracted into CSV files containing 100 000 packets of feature data with 95 573 normal data and 4427 attack data. The random tree algorithm also classifies the packet data into two clusters. Like the K-Means classifier, the Random Tree algorithm also only considers the ping flood attack pattern features/attributes. Figure 6 depicts the classification results. The clustering process generated cluster 0 with 96 290 packets, and 3710 packets in cluster 1. As can be seen from the centroid of cluster 0 with an ip.flags value of 0 and a frame.len value of 42 (which was the same for every packet), it contained the ping flood attack pattern. The cluster 1 centroid had an ip.flags value of 0.0629 and a frame.len value of 56.8429. it can be stated that cluster 0 was the attack packet cluster, while cluster 1 was the normal packet cluster.

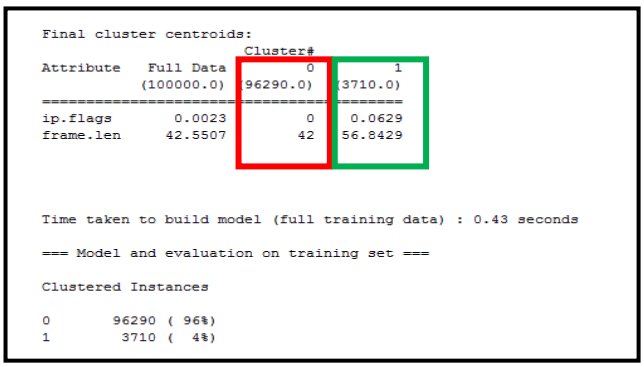

FIGURE 6. Clustering result using random tree algorithm.

G. CONFUSION MATRIX OF CALCULATED RESULTS

To measure the accuracy of the K-Means and Random Tree algorithms, a confusion matrix was used. The initial step in creating a confusion matrix is to break down the clustering results into four categories: (1) true positives (TP), attack packets that were detected correctly; (2) true negatives (TN), normal packets rejected correctly; (3) false positives (FP), normal packets incorrectly identified as attack packets; and (4) false negatives (FN), attack packets not detected. Having obtained the correct values of the four categories, the calculation of the detection rate, false alarm rate, and accuracy can then be made. Both the K-Means algorithm and the random tree algorithm generated the same clustering results for the normal and attack clusters. This fact was caused by the same attributes were used in both clustering processes; they were the attributes selected from the ping flood attack pattern. Table 5 shows the results of the classification using the K-Means and random tree algorithm.

TABLE 5. Classification process result.

	Attack Data		Combined Data	
	Attack	Normal	Attack	Normal
Actual (raw data)	95,511	4,489	96,227	3,773
K-Means	95,573	4,427	96,290	3,710
Random Tree	95,573	4,427	96,290	3,710

TABLE 6. Confusion matrix values.

Metric	K-mean		Random Tree	
	Attack	Combined	Attack	Combined
TP	95,511	96,227	95,511	96,227
FP	62	63	62	63
FN	0	0	0	0
TN	4,427	3,710	4,422	3,707

Table 6 shows the confusion matrix from the K-Means and Random Tree algorithm. The number OF falsely detected attacks was 62 in the attack dataset and was 63 in the combined dataset. The detection metrics are calculated based on the confusion matrix and are presented in Table 7.

TABLE 7. Detection rate.

Metric (%)	K-Means		Random Tree	
	Attack dataset	Normal dataset	Attack dataset	Normal dataset
TPR	100	100	100	100
FPR	1.38	1.67	1.38	1.67
TNR	98.62	98.33	98.12	98.00
FNR	0	0	0	0
Precision	99.94	99.93	99.85	99.80
Accuracy	99.94	99.94	99.88	99.87

Table 7 shows that the false negative rate (FNR) for both the attack and the normal-attack datasets was zero. This value indicates that there were no packets that matched the criteria were missed.

The level of accuracy for both datasets was 99.94%, which indicates that the overall ping flood attack detection results were very good. Figure 7 graphically shows the detection metrics listed in Table 6.

The graph in Figure 7 shows the percentages of the K-Means detection metrics from the attack and normal-attack datasets. The percentages in both datasets were only slightly different. By referring to the information in table 6, the percentage difference in the FPR between the attack and normal-attack datasets was 0.29%, The TNR difference was

−0.29%, and the precision difference between the two was −0.01%, whereas the other values were the same.

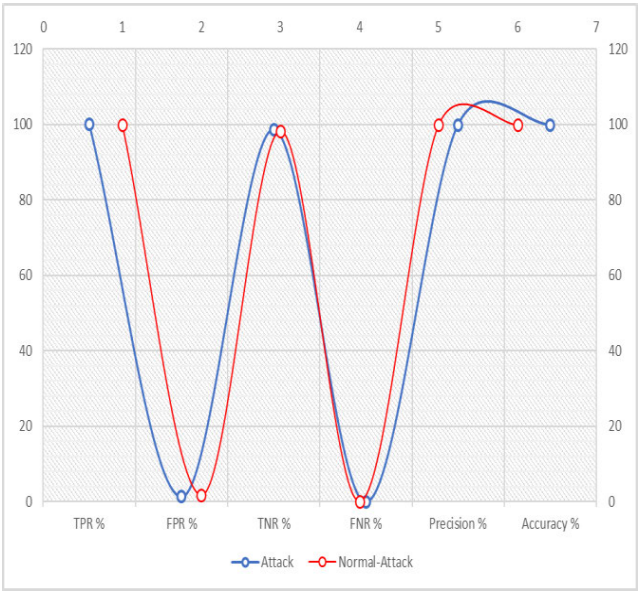

FIGURE 7. Detection rate of confusion matrix of K-Means clustering value.

H. INVESTIGATION ON ENERGY CONSUMPTION

The run time complexity of each iteration in K-Means clustering analysis is calculated by *O(KNT)*, where *K* is the number of clusters, *N* is the number of tuples in the dataset, and *T* is processing time to calculate the distance between two data objects. Hence, for *I* number of iteration, the time complexity of the algorithm is *O(IKNT)* [41].

In this work there are two clusters only (attacks or normal), thus, K = 2, N is the number of attacks data (tuples of frame.len and ip.flags values). T is measured and iteration number I is the number of dataset entries being clustered. We set a pin high at the start of the program and set a pin low, at the end of the program. As long as the program is repeated, the execution time is computed by knowing the time between the two highs. We measure the best, the average and the worst values of T with the following scenarios.

a) Best case scenario: The dataset is arranged with the attack records are placed randomly in the first 30,000 positions with the probability of 95%.
b) Average case scenario: The dataset is arranged with the attack records are placed randomly between 30,000 and 70,000 positions with the probability of 95%.
c) Worst case scenario: The dataset is arranged with the attack records are placed randomly in the last 30,000 positions with the probability of 95%.

Ten runs are carried out for each scenario and the average time is recorded.

Experiment is carried out using computer with Intel Core i7-8086K processor and 16 GB RAM. The processor has raw processing power of 221,720 MIPS at 5.0 GHz frequency. Referring to research work by Oliveira *et al.* [42], energy consumption for the clustering process is estimated as follows. The energy consumption of the processor per million instructions is $(MIPS)^2 * 10^{-8}$ [43]. Hence, the estimated energy consumption for clustering process is calculated by formula in (1).

$$\boldsymbol{Energy} = \{\boldsymbol{MIPS^2} * \boldsymbol{10^{-8}}\} * \boldsymbol{T} \quad (1)$$

The measured average processing time for an iteration of clustering with K-Means and with Random Tree algorithm is 0.0566 seconds and 0.0711 seconds, respectively. Thus, the MIPS for K-Means algorithm is 221,720*0.0566=12,550 and the MIPS for the Random Tree algorithm is 221,720*0.0711=15,765. Computational results are shown in Table 8.

TABLE 8. Energy consumption (in Joule/Second).

Case	K-mean		Random Tree	
	Time (ms)	Energy	Time (ms)	Energy
Best	102.09	0.161563959	103.24	0.256587766
Ave.	114.30	0.180887065	114.98	0.285765802
Worst	171.12	0.270808352	183.67	0.456484648

Observing the computation results on energy consumption of the proposed IDS in Table 8, the energy consumption to detect ICMP flooding attacks is considerably high. Thus, the proposed IDS should be installed on a device with sustain power, such as server or PC, either as centralized IDS or distributed IDS. Nevertheless, the proposed IDS does not involve any collaborative works among nodes, therefore, it is not so beneficial in installing the IDS on each node attached to IoT networks.

V. CONCLUSION

Observation from the experimental results brings us to the conclusion that ping flood attack utilizes the freedom of the ICMP, which allows a user to send a packet echo to a host. This attack utilizes an echo request to flood the victim (in this case, nodes), thus interfering with the victim's network traffic. The attack was recognized through unique attributes of the packet header, i.e.: the length on the frame header and a flag in the IP header. During the experiments, two priority alerts were detected, i.e.: a Priority 2 and Priority 3. A lower priority number indicates a higher level of danger from an attack.

Clustering normal and attack data packets based on ping flood attack pattern attributes by K-Means and by Random Tree algorithm provide similar results. The evaluation of the ping flood attack detection results with the implemented K-Means algorithm can be said to be very good, with accuracy levels of up to 99.94%.

As stated by Khraisat and Alazab [44], there are four important characteristics of reliable IoT IDS development, i.e.: (1) low false alarms rate due to the huge volume of data, (2) very adaptive to extreme IoT communication systems

due to unpredicted sensors' behavior that usually indicating attacks, (3) ability of zero-day attacks detection, and lastly, (4) self-governing ability, thru the use of machine learning and deep learning techniques to learn from big data of IoT. Thus, as a future work, the authors plan to implement different machine learning and deep learning techniques as clustering algorithms, to improve IDS performance.

Machine learning based IDSs adopt supervised methods that deeply depend on human experts' observations for labeling processes, feature extraction and selection of large training data for classification. Therefore, IoT IDSs with huge scale or high dimensional data need unsupervised approach to increase the prediction accuracy of the classification in detecting three (3) most popular types of attacks, i.e.: Flooding, Injection and Impersonate attacks. Such unsupervised approach uses automatic feature extraction and selection methods to replace human intervention and manual labeling process. Thus, the use of feature extraction and feature selection methods to filter significant features from packet's attribute is also considered as one of the future works.

REFERENCES

[1] M. Abdur, S. Habib, M. Ali, and S. Ullah, "Security issues in Internet of Things (IoT): A comprehensive review," *Int. J. Adv. Comput. Sci. Appl.*, vol. 8, no. 6, pp. 359–369 2017, doi: 10.14569/ijacsa.2017.080650.

[2] S. Zahoor and R. N. Mir, "Resource management in pervasive Internet of Things: A survey," *J. King Saud Univ.-Comput. Inf. Sci.*, Sep. 2018, doi: 10.1016/j.jksuci.2018.08.014.

[3] M. Masoud, Y. Jaradat, A. Manasrah, and I. Jannoud, "Sensors of smart devices in the Internet of Everything (IoE) era: Big opportunities and massive doubts," *J. Sensors*, vol. 2019, pp. 1–26, May 2019, doi: 10.1155/2019/6514520.

[4] D. Ciuonzo, P. S. Rossi, and P. K. Varshney, "Distributed detection in wireless sensor networks under multiplicative fading via generalized score tests," *IEEE Internet Things J.*, vol. 8, no. 11, pp. 9059–9071, Jun. 2021.

[5] I. Kök, M. U. Şimşek, and S. Özdemir, "A deep learning model for air quality prediction in smart cities," in *Proc. IEEE Int. Conf. Big Data*, Boston, MA, USA, Dec. 2017, pp. 1983–1990.

[6] Z. A. Almusaylim, A. Alhumam, and N. Z. Jhanjhi, "Proposing a secure RPL based Internet of Things routing protocol: A review," *Ad Hoc Netw.*, vol. 101, Apr. 2020, Art. no. 102096.

[7] A. Triantafyllou, P. Sarigiannidis, and T. D. Lagkas, "Network protocols, schemes, and mechanisms for Internet of Things (IoT): Features, open challenges, and trends," *Wireless Commun. Mobile Comput.*, vol. 2018, Sep. 2018, Art. no. 5349894

[8] C. Pu and T. Song, "Hatchetman attack: A denial of service attack against routing in low power and lossy networks," in *Proc. 5th IEEE Int. Conf. Cyber Secur. Cloud Comput. (CSCloud)*, Shanghai, China, Jun. 2018, pp. 12–17.

[9] P. P. Ioulianou and V. G. Vassilakis, "Denial-of-service attacks and countermeasures in the RPL-based Internet of Things," in *Computing Security* (Lecture Notes in Computer Science), vol. 11980, S. Katsikas, Ed. Cham, Switzerland: Springer, 2020, pp. 374–390.

[10] C. Pu and S. Hajjar, "Mitigating forwarding misbehaviors in RPL-based low power and lossy networks," in *Proc. 15th IEEE Annu. Consum. Commun. Netw. Conf. (CCNC)*, Las Vegas, NV, USA, Jan. 2018, pp. 1–6.

[11] S. M. Muzammal, R. K. Murugesan, and N. Z. Jhanjhi, "A comprehensive review on secure routing in Internet of Things: Mitigation methods and trust-based approaches," *IEEE Internet Things J.*, vol. 8, no. 6, pp. 4186–4210, Oct. 2020.

[12] A. Djenna, S. Harous, and D. E. Saidouni, "Internet of Things meet Internet of Threats: New concern cyber security issues of critical cyber infrastructure," *Appl. Sci.*, vol. 11, no. 10, p. 4580, May 2021, doi: 10.3390/app11104580.

[13] K. Sonar and H. Upadhyay, "An approach to secure Internet of Things against DDoS," *Adv. Intell. Syst. Comput.*, vol. 409, pp. 367–376, Dec. 2016, doi: 10.1007/978-981-10-0135-2.

[14] Y.-M. Kwon, J. Yu, B.-M. Cho, Y. Eun, and K.-J. Park, "Empirical analysis of MAVLink protocol vulnerability for attacking unmanned aerial vehicles," *IEEE Access*, vol. 6, pp. 43203–43212, 2018, doi: 10.1109/ACCESS.2018.2863237.

[15] H. Harshita, "Detection and prevention of ICMP flood DDOS attack," *Int. J. New Technol. Res.*, vol. 3, no. 3, 2017, Art. no. 263333.

[16] R. Doshi, N. Apthorpe, and N. Feamster, "Machine learning DDoS detection for consumer Internet of Things devices," in *Proc. IEEE Secur. Privacy Workshops (SPW)*, San Francisco, CA, USA, May 2018, pp. 29–35, doi: 10.1109/SPW.2018.00013.

[17] A. A. Diro and N. Chilamkurti, "Distributed attack detection scheme using deep learning approach for Internet of Things," *Future Gener. Comput. Syst.*, vol. 82, pp. 761–768, May 2018, doi: 10.1016/j.future.2017.08.043.

[18] D. E. Kouicem, A. Bouabdallah, and H. Lakhlef, "Internet of Things security: A top-down survey," *Comput. Netw.*, vol. 141, pp. 199–221, Aug. 2018, doi: 10.1016/j.comnet.2018.03.012.

[19] B. Bouyeddou, F. Harrou, Y. Sun, and B. Kadri, "Detection of smurf flooding attacks using Kullback-Leibler-based scheme," in *Proc. 4th Int. Conf. Comput. Technol. Appl. (ICCTA)*, Istanbul, Turkey, May 2018, pp. 11–15, doi: 10.1109/CATA.2018.8398647.

[20] K. P. Sinaga and M.-S. Yang, "Unsupervised K-means clustering algorithm," *IEEE Access*, vol. 8, pp. 80716–80727, 2020, doi: 10.1109/ACCESS.2020.2988796.

[21] Y. Jeong, J. Lee, J. Moon, J. H. Shin, and W. D. Lu, "K-means data clustering with memristor networks," *Nano Lett.*, vol. 18, no. 7, pp. 4447–4453, 2018, doi: 10.1021/acs.nanolett.8b01526.

[22] M. J. Rezaee, M. Eshkevari, M. Saberi, and O. Hussain, "GBK-means clustering algorithm: An improvement to the K-means algorithm based on the bargaining game," *Knowl.-Based Syst.*, vol. 213, Feb. 2021, Art. no. 106672, doi: 10.1016/j.knosys.2020.106672.

[23] T. Yu and H. Zhu, "Hyper-parameter optimization: A review of algorithms and applications," 2020, *arXiv:2003.05689*. [Online]. Available: http://arxiv.org/abs/2003.05689

[24] S. Pourahmad, A. Basirat, A. Rahimi, and M. Doostfatemeh, "Does determination of initial cluster centroids improve the performance of K-means clustering algorithm? Comparison of three hybrid methods by genetic algorithm, minimum spanning tree, and hierarchical clustering in an applied study," *Comput. Math. Methods Med.*, vol. 2020, pp. 1–11, Aug. 2020, doi: 10.1155/2020/7636857.

[25] M. Pietrzykowski, "Local regression algorithms based on centroid clustering methods," *Procedia Comput. Sci.*, vol. 112, pp. 2363–2371, Jan. 2017, doi: 10.1016/j.procs.2017.08.210.

[26] Y. Lu, B. Cao, C. Rego, and F. Glover, "A Tabu search based clustering algorithm and its parallel implementation on spark," *Appl. Soft Comput.*, vol. 63, pp. 97–109, Feb. 2018, doi: 10.1016/j.asoc.2017.11.038.

[27] A. K. Mishra, A. K. Tripathy, D. Puthal, and L. T. Yang, "Analytical model for sybil attack phases in Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 379–387, 2019, doi: 10.1109/JIOT.2018.2843769.

[28] C. Pu, S. Lim, B. Jung, and J. Chae, "EYES: Mitigating forwarding misbehavior in energy harvesting motivated networks," *Comput. Commun.*, vol. 124, pp. 17–30, Jun. 2018, doi: 10.1016/j.comcom.2018.04.007.

[29] P. Shukla, "ML-IDS: A machine learning approach to detect wormhole attacks in Internet of Things," in *Proc. Intell. Syst. Conf.*, London, U.K., Sep. 2017, pp. 234–240, doi: 10.1109/IntelliSys.2017.8324298.

[30] H. Qiu, T. Dong, T. Zhang, J. Lu, G. Memmi, and M. Qiu, "Adversarial attacks against network intrusion detection in IoT systems," *IEEE Internet Things J.*, vol. 8, no. 13, pp. 10327–10335, Jul. 2021.

[31] G. Bovenzi, G. Aceto, D. Ciuonzo, V. Persico, and A. Pescape, "A hierarchical hybrid intrusion detection approach in IoT scenarios," in *Proc. IEEE Global Commun. Conf.*, Taipei, Taiwan, Dec. 2020, pp. 7–11, doi: 10.1109/GLOBECOM42002.2020.9348167.

[32] N. Abughazaleh, R. Bin, and M. Btish, "DoS attacks in IoT systems and proposed solutions," *Int. J. Comput. Appl.*, vol. 176, no. 33, pp. 16–19, Jun. 2020, doi: 10.5120/ijca2020920397.

[33] M. F. Elrawy, A. I. Awad, and H. F. A. Hamed, "Intrusion detection systems for IoT-based smart environments: A survey," *J. Cloud Comput.*, vol. 7, no. 1, pp. 1–20, Dec. 2018, doi: 10.1186/s13677-018-0123-6.

[34] E. Hodo, X. Bellekens, A. Hamilton, P.-L. Dubouilh, E. Iorkyase, and C. Tachtatzis, "Threat analysis of IoT networks using artificial neural network intrusion detection system," in *Proc. Int. Symp. Netw., Comput. Commun. (ISNCC)*, Yasmine Hammamet, Tunisia, May 2016, pp. 4–8, doi: 10.1109/ISNCC.2016.7746067.

[35] A. Fadlil, I. Riadi, and S. Aji, "Review of detection DDOS attack detection using naive Bayes classifier for network forensics," *Bull. Electr. Eng. Informat.*, vol. 6, no. 2, pp. 140–148, Jun. 2017, doi: 10.11591/eei.v6i2.605.
[36] F. Hussain, S. G. Abbas, M. Husnain, U. U. Fayyaz, F. Shahzad, and G. A. Shah, "IoT DoS and DDoS attack detection using ResNet," in *Proc. IEEE 23rd Int. Multitopic Conf. (INMIC)*, Bahawalpur, Pakistan, Nov. 2020, pp. 1–6, doi: 10.1109/INMIC50486.2020.9318216.
[37] F. S. de Lima Filho, F. A. F. Silveira, A. M. Brito, G. Vargas-Solar, and L. F. Silveira, "Smart detection: An online approach for DoS/DDoS attack detection using machine learning," *Secur. Commun. Netw.*, vol. 2019, Oct. 2019, Art. no. 1574749, doi: 10.1155/2019/1574749.
[38] Y. Li, D. E. Quevedo, S. Dey, and L. Shi, "A game-theoretic approach to fake-acknowledgment attack on cyber-physical systems," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 3, no. 1, pp. 1–11, Mar. 2017, doi: 10.1109/TSIPN.2016.2611446.
[39] P. Kumar, R. Kumar, G. P. Gupta, and R. Tripathi, "A distributed framework for detecting DDoS attacks in smart contract-based blockchain-IoT Systems by leveraging Fog computing," *Trans. Emerg. Telecommun. Technol.*, vol. 32, no. 6, pp. 1–31, Jun. 2021, doi: 10.1002/ett.4112.
[40] S. Dibb, "Market segmentation: Conceptual and methodological foundations," *J. Targeting, Meas. Anal. for Marketing*, vol. 9, no. 1, pp. 92–93, Aug. 2000.
[41] B. Angelin and A. Geetha, "Outlier detection using clustering techniques–K-means and K-median," in *Proc. 4th Int. Conf. Intell. Comput. Control Syst. (ICICCS)*, Madurai, India, May 2020, pp. 373–378, doi: 10.1109/ICICCS48265.2020.9120990.
[42] H. F. A. Oliveira, A. V. Brito, J. M. F. R. Araujo, and E. U. K. Melcher, "An approach for power estimation at electronic system level using distributed simulation," *J. Integr. Circuits Syst.*, vol. 11, no. 3, pp. 159–170, Dec. 2016.
[43] E. García-Martín, C. F. Rodrigues, G. Riley, and H. Grahn, "Estimation of energy consumption in machine learning," *J. Parallel Distrib. Comput.*, vol. 134, pp. 75–88, Dec. 2019.
[44] A. Khraisat and A. Alazab, "A critical review of intrusion detection systems in the Internet of Things: Techniques, deployment strategy, validation strategy, attacks, public datasets and challenges," *Cybersecurity*, vol. 4, no. 1, pp. 1–27, Dec. 2021, doi: 10.1186/s42400-021-00077-7.

DERIS STIAWAN received the Ph.D. degree in computer science from the Universiti Teknologi Malaysia, Malaysia. He is currently an Associate Professor with the Department of Computer Engineering, Faculty of Computer Science, Universitas Sriwijaya. His research interests include computer networks, intrusion detection/prevention systems, and heterogeneous networks.

MEILINDA EKA SURYANI received the bachelor's degree from the Department of Computer Engineering, Universitas Sriwijaya, Indonesia, in 2018. Since 2017, she has been with the Computer Network and Information Security (COMNETS) Research Group, Faculty of Computer Science, Universitas Sriwijaya. Her research interests include both theoretical and practical aspect of the Internet of Things and computer networks.

SUSANTO received the master's degree in computer science from the Universitas Bina Darma, Palembang, Indonesia. He is currently pursuing the Ph.D. degree with the Faculty of Engineering, Universitas Sriwijaya. He is also a Senior Lecturer with the Faculty of Computer Science, Universitas Bina Insan, Indonesia. His research interests include cryptography, information technology, information security, and network security.

MOHD YAZID IDRIS received the M.Sc. degree in software engineering in 1998, and the Ph.D. degree in information technology (IT) security in 2008. In software engineering, he focuses on the research of designing and development of mobile and telecommunication software. He is currently an Associate Professor with the Faculty of Engineering, School of Computing, Universiti Teknologi Malaysia. His main research activity in IT security is in the area of intrusion prevention and detection (IPD).

MUAWYA N. ALDALAIEN received the Ph.D. degree in computer sciences from the Universiti Sains Malaysia, in 2011. He is currently working with King Hussein School of Computing Sciences, Princess Sumaya University for Technology, Jordan. He has 12 years of experience in the area of scientific research, where he has published a number of journal articles, conference proceedings, and book chapters. His research interests include IMD security, network security protocols, cryptography, cloud computing, the IoT, and cybersecurity law and regulations.

NIZAR ALSHARIF received the M.A.Sc. and Ph.D. degrees from the Department of Electrical and Computer Engineering, University of Waterloo, Canada, in 2010 and 2017, respectively. He is currently an Assistant Professor with the Department of Computer Engineering and Science, Al-Baha University, Al Bahah, Saudi Arabia. His research interests include the IoT systems and security, wireless sensor networks, vehicles networking, and smart city.

RAHMAT BUDIARTO received the M.Eng. and Dr.Eng. degrees in computer science from Nagoya Institute of Technology, Japan, in 1995 and 1998, respectively. He is currently a Full Professor with the Department of Computer Engineering and Science, Al-Baha University, Al Bahah, Saudi Arabia. His research interests include intelligent systems, brain modeling, IPv6, network security, wireless sensor networks, and MANETs.