IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 12, 2017, accepted January 15, 2018, date of publication January 26, 2018, date of current version April 18, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2798626

Compression Header Analyzer Intrusion Detection System (CHA - IDS) for 6LoWPAN Communication Protocol

MOHAMAD NAZRIN NAPIAH, MOHD YAMANI IDNA BIN IDRIS, ROZIANA RAMLI, AND ISMAIL AHMEDY

Department of Computer System and Technology, Faculty of Computer Science and Information Technology, University of Malaya, Kuala Lumpur 50603, Malaysia

Corresponding author: Mohd Yamani Idna Bin Idris (yamani@um.edu.my)

This work was supported by the University of Malaya Research under Grant Scheme RP036 (A, B, C)–15AET.

ABSTRACT Prior 6LoWPAN intrusion detection system (IDS) utilized several features to detect various malicious activities. However, these IDS methods only detect specific attack but fails when the attacks are combined. In this paper, we propose an IDS known as compression header analyzer intrusion detection system (CHA-IDS) that analyzes 6LoWPAN compression header data to mitigate the individual and combination routing attacks. CHA-IDS is a multi-agent system framework that capture and manage raw data for data collection, analysis, and system actions. The proposed CHA-IDS utilize best first and greedy stepwise with correlation-based feature selection to determine only significant features needed for the intrusion detection. These features are then tested using six machine learning algorithms to find the best classification method that able to distinguish between an attack and non-attack and then from the best classification method, we devise a rule to be implemented in Tmote Sky. To ensure the reliability of our proposed method, we evaluate the CHA-IDS with three types of combination attacks known as hello flood, sinkhole, and wormhole. We also compare our results in term of accuracy of detection, energy overhead, and memory consumption with the prior 6LoWPAN-IDS implementation such as SVELTE and Pongle's IDS. The results show that CHA-IDS performs better than the aforementioned methods with 99% true positive rate and consumed low energy overhead and memory that fit in constrained device such Tmote Sky.

INDEX TERMS Internet of Things, security, machine learning, compression header, 6LoWPAN, RPL, routing attack.

I. INTRODUCTION

IPv6 over Low-power Wireless Personal Area Network protocol (6LoWPAN) has been widely used as an adaption layer between the standard IPv6 protocol and IEEE 802.15.4 link layer. Thus, enables the resource constrained devices to effectively transmit information via the standard IPv6. In 6LoWPAN network, Routing Protocol for Low Power and Lossy Network (RPL) has been introduced as a routing protocol to deal with limited memory, power etc. RPL creates Destination Oriented Directed Acyclic Graph (DODAG) [1] and enables the nodes to forward the packets upwards to their parents or downward to their children. However, in such constrained environment, RPL has limited support for security services and are exposed to internal attacks.

There are three main attacks that targeting the RPL protocol in IoT namely hello flood, sinkhole, and wormhole attacks. Several researchers have put their effort to propose 6LoWPAN Intrusion Detection System (6LoWPAN-IDS) to mitigate the aforementioned problems. There are two well-known 6LoWPAN-IDS implementations, namely SVELTE [2] and Pongle's IDS [1]. These methods are efficient in detecting the individual routing attack in which, SVELTE is designed specifically to detect sinkhole attack whereas Pongle's focus on detecting wormhole attack. However, SVELTE and Pongle's IDS can be ineffective against a complex and more destructive attack that consists of multiple or combination of attacks that produce new anomaly attack.

Therefore, in this paper, we propose Compression Header Analyzer Intrusion Detection System (CHA-IDS) to detect individual routing attacks and their combination in 6LoWPAN network. CHA-IDS utilizes compression header of 6LoWPAN as a feature instead of rank and Received Signal Strength Indicator (RSSI) that used in

SVELTE and Pongle's. This is because the compression header of 6LoWPAN is unique and contain essential information needed to detect a complex attack. The framework of the proposed CHA-IDS is divided into four layers; sensor agent (SA), aggregator agent (AGA), analyzer agent (ANA) and actuator agent (ACA). The framework will capture network packet, extract significant features based on normal and abnormal node, analyses data class labelling, and finally alert the user if malicious activity is detected.

The pattern of the 6LoWPAN Compression Header is studied using machine learning algorithm and the best rules are selected. Based on the selected rules, we compare the performance of the proposed CHA-IDS with the existing method (i.e. SVELTE and Pongle's [2], [1]) in term of accuracy. The rest of this paper is organized as follows: types of the attacks and the related works are discussed in Section 2. The framework of the proposed method, evaluation metrics and experimental setup are described in Section 3. Section 4 presents the experiments and the evaluation of the performance of the proposed method and Section 5 discusses the overall findings of the proposed work. Finally, the conclusions are drawn and future works are highlighted in Section 6.

II. RELATED WORK

A. INTRUSION DETECTION SYSTEM (IDS)

Intrusion detection system (IDS) is a system used for monitoring malicious traffic in a network. The IDS act as a second line defense to protect the network from intruders [3]. The majority of IDS techniques proposed in the literature are mainly based on three types of IDS such as signature-based, anomaly-based and hybrid-based.

1) SIGNATURE BASED IDS

Earliest work on IDS techniques have been proposed based on signature. The signature-based method actively compares and matches the events from a network against a predefined attack signature or pattern from a database [3]. This approach needs specific knowledge of an individual attack in order to work properly. Therefore, new attacks are undetectable unless their signatures or patterns are manually added into the database [3]. For this reason, the database needs to be updated frequently with signatures of new attacks. This approach has two main disadvantages: a) malicious data are required to form an attack pattern. b) unable to discover new or unknown attacks.

2) ANOMALY BASED IDS

Contrarily, anomaly-based method (or also known as event-based detection) identifies malicious activities by analyzing the unusual events of a network. This technique starts by defining the normal behavior of a network followed by a comparison between the current protocol specifications with previously defined protocol state [3]. If normal activity changes unexpectedly, they may be marked as an intrusion. The anomaly-based method detects attacks more efficiently than their signature-based counterpart since no predefined signature is needed. Apart from that, anomaly-based method is able to detect new attack patterns. Though significantly known for having these two advantages, anomaly-based method facing difficulties in determining the normal region. This is due to blur boundary between the observation of normal and abnormal behavior. In some cases, an abnormal observation may encroach to the boundary of normal observation, or vice-versa.

3) HYBRID BASED IDS

To overcome signature and anomaly-based weaknesses, a hybrid IDS has been proposed by combining both techniques. This combination enables the hybrid IDS to detect both cases either misuse or anomaly attacks. Aydin *et al.* [4] proposed the hybrid IDS by combining packet header anomaly detection (PHAD) and network traffic anomaly detection using open source tools such as SNORT. In their work, the standard PHAD was altered by modelling the protocols rather than the user behaviors. This model depends on the rapid change of network statistics in a short term.

B. MACHINE LEARNING ALGORITHM IN INTRUSION DETECTION SYSTEM

Generally, machine learning algorithm is an artificial intelligence that learned or adapted to new environment. The machine learning algorithms are widely used in network security for wireless sensor network environment. The algorithm usually operates based on the features that represents the characteristic of the object.

There a several IDS methods that used machine learning algorithm as intrusion indicator. Shamshirband *et al.* [5] proposed a hybrid clustering method namely density-based fuzzy imperialist competitive clustering algorithm (D-FICCA) which is a modification from density-based algorithm and fuzzy logic to enhance the accuracy of malicious detection such as DoS attacks. Then, Shamshirband *et al.* [6] proposed Cooperative fuzzy artificial immune system (Co-FAIS) to mitigate the DoS attacks. D-FICCA and Co-FAIS have the advantages to predict the presence of DoS attack and equipped with the counter-defense mechanism for wireless sensor network. However, energy and memory consumption of D-FICCA and Co-FAIS in constrained device such as WSN node are not provided by the authors.

Selecting suitable features for machine learning algorithm is important to differentiate between normal and anomaly behaviors. Kolias *et al.* [7] used several machine learning algorithms to analyze 156 features collected from 802.11 network. Then, 20 significant features obtained from 156 features are associated according to their specific attacks. This feature selection is performed to select only important features as well as to reduce the consumption of device resources. These significant features can be used as an IDS indicator to detect any attacks. Therefore, in this paper, the machine learning algorithm is applied to find the new

significant features and then classifies the features as intrusion indicator for an IDS.

C. INTRUSION DETECTION SYSTEM FOR 6LoWPAN BASED NETWORK

The rapid advancement of the internet and sensor network technology has made IDS in IoT environment become essential each day. The intrusion in IoT can be categorized as an outside or inside attack. In outside attack, the attack is originated from the outside of the network/internet which connect to 802.11 protocol. Examples of such attacks are fragmentation attack, botnet attack, etc. For the inside attack, the attack can be initiated by compromised or malicious nodes that are part of the network. In this paper, we will concentrate on the inside attacks within the 6LoWPAN protocol. To start with, a brief background of 6LoWPAN protocol will be presented. This is then followed by the review of the most common 6LoWPAN routing attacks such hello flood, sinkhole, wormhole and others attacks.

1) 6LoWPAN BACKGROUND

6LoWPAN is a network architecture for low-power wireless area networks, which is IPv6 stub network [8]. It is part of IP-based infrastructures that uses IPv6 packet to route in 6LoWPAN network [5]. 6LoWPAN fragments and reassembles the datagram because of the limited payload size in the IoT devices [5]. All the device node are connected to the Internet through a getaway known as 6LoWPAN Border Router (6BR) that similar to a sink node in WSN network [5] as shown in Figure 1. It performs the compression or decompression and fragmentation or assembly of IPv6 datagrams that can be implemented on any devices and not restricted to constrained device only.

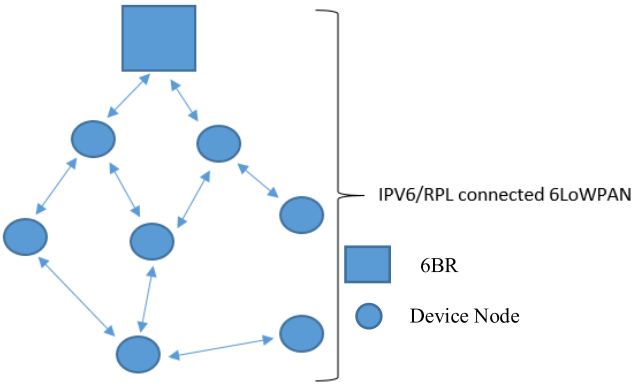


FIGURE 1. Communication of connected device in 6LoWPAN network with 6LoWPAN Border Router.

Destination Oriented Directed Acyclic Graph (DODAG) [1] is IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) that forms like a tree topology with one root known as a sink node. The formation of the topology root node requires the transmission of DODAG Information Object (DIO) messages. After that, other nodes that receive the DIO message will select the parent based on the rank value calculated. If the rank sender is higher, the sender will be selected as parents or vice versa [1]. The distance from the root node and energy of the link may influence the rank value. The network owner can decide the rank value by calculating a parameter or continue to transmit the DIO message to form the tree topology [1].

6BR is assumed to be always accessible from inside and outside the network, hence, end-to-end security is required in the implementation of IoT. Furthermore, sensor nodes in the network are globally identified by an IP address that can be accessed through the Internet. Because of that, implementation of IDS is important in the network to monitor any intrusions.

2) ROUTING ATTACKS IN 6LoWPAN PROTOCOL

The attack occurs on 6LoWPAN network can be classified as external or internal attacks. External attack is the attack initiated from Internet side while the internal attack is initiated from the wireless sensor network side. Examples of the external attack are Brute force attack, malware attack, SSL attack and DNS attack. The internal attack includes the routing attacks such as hello flood, sinkhole and wormhole attacks. The routing attack aims to disrupt the network layer while routing messages from one node to another node in the network. In this paper, we will focus on the routing attacks and discuss the literature that related to the attacks mitigation in the following sub-section.

a: HELLO FLOOD

The hello flood attack utilizes the hello message to make the legitimate nodes believe that the malicious nodes is their neighboring nodes and within their range limit, even though they are far away from the network. This will result in packet loss since the packet travel is deceived by the malicious nodes. Hello flood attack can be identified by utilizing the UDP packet rate. The attack is considered as the Hello Flood attack if the rate of UDP packet is at 30 packets per second [4]. In another work, Sherasiya *et al.* [3] integrated an IDS into the network framework that developed within the EU FP7 project rabbit. Their aims are to detect Hello Flood attacks targeted for 6LoWPAN protocol using an open source Network Intrusion Detection System (NIDS) known as Snort. The Snort will monitor the network traffic and analyzed them against a set of Hello Flood signature rule defined by the researchers. An attack is detected when their signatures are matched with each other.

PalSingh *et al.* [9] on the other hand proposed a method that utilizes the signal strength and distance between two nodes to identify the Hello Flood attack. A Hello Flood Attacker may use a high-quality signal to allure other nodes to use its route even they are not part of the neighbors or they are located at distance apart. For this reason, PalSingh *et al.* [9] sent a test packet and if the packet doesn't come back in a predefined time, they are considered as a stranger. In a more recent study, Grgic *et al.* [10] proposed an IDS based on

distributed algorithms and collective decision-making process. Their methods used the concept of probability estimation to detect the malicious behavior of the nodes. The probability estimation, however, relies on node failure in order to achieve considerable estimation value. This approach might have a high false positive rate since some attacks do not aim to force the node to fail or to become malfunction.

b: SINKHOLE ATTACK

The aim of sinkhole attack is to control packet traffic in a network as much as possible through malicious node. The attacker deceives the legitimate nodes to establish link with the malicious node by pretending having the optimal routes. To mitigate sinkhole attack, Pongle and Chavan [1] proposed SVELTE which monitors the rank information of client node in 6LoWPAN environment. SVELTE is a hybrid of signature-based and anomaly-based IDS method to detect sinkhole and selective forwarding attack. However, SVELTE depends on the availability of the rank information advertised by the client node. The rank information might be lost if an attacker disrupts the network by force. This is in view to the fact that the attack may force the client node to drop their packet.

c: WORMHOLE ATTACK

In wormhole attack, the attacker disrupts the network routing by tunneling a wormhole link between two colluding compromised nodes that are far apart from each other. Pongle and Chavan [1] proposed IDS for wormhole attack detection using RSSI values. The presence of wormhole attack is detected by knowing the location of each node whether the nodes invalid range based on RSSI values received. In another approach, Lai [11] proposed methods to detect wormhole by calculating distance based on the rank of the node defined in the network.

d: OTHER ATTACKS

There are also several other attacks that may disrupt the 6LoWPAN network such as selective forwarding and blackhole. In selective forwarding, malicious nodes selectively forward packets that targeted to disrupt routing paths [6] whereas blackhole silently sucks in all data packets and drops them [5]. All the attacks stated in this section and their respective IDS methods are summarized in Table 1.

III. PROPOSED METHOD

This section is divided into four sub-sections as follows. In the first sub-section, we explain the routing information that available in the 6LoWPAN compression header. The second sub-section described the framework of the proposed method that consists of four layers. Then, we present a set of evaluation criteria to validate the performance of the proposed method. Finally, network configuration for hello flood,

TABLE 1. Summarization of the attacks and their intrusion detection system.

Attacks	Architecture	IDS	Description
Hello Flood	Signature	Ebbit [3]	Count the packets using Snort IDS to validate the attack. **Advantage:** Can determine the attack accurately **Disadvantages:** Unsuitable for constrained devices.
	Anomaly	Co-FAIS [6]	Used Fuzzy Q-Learning to improve detection accuracy. **Advantage:** Predicts DDOS attack. **Disadvantages:** Unsuitable for constrained devices.
	Anomaly	D-FICCA [5]	Proposed a hybrid clustering method known as density-based fuzzy imperialist competitive clustering algorithm. **Advantage:** Predicts DDOS attack. **Disadvantages:** Unsuitable for constrained devices.
Sinkhole	Hybrid	SVELTE [2]	Monitor rank advertises by neighbors. **Advantage:** Suitable for constrained devices and detect anomalies. **Disadvantages:** Unable to detect other attacks such as wormhole attack
Wormhole	Anomaly	Wormhole IDS [1]	Detects wormhole with encapsulation & wormhole with packet relay by monitoring neighbor's RSSI value. **Advantage:** Suitable for constrained devices. **Disadvantages:** Unable to detect anomalies of other attacks such as sinkhole and hello flood.
	Hybrid	Gu Hsin Lai [11]	Detects malicious wormhole node if unreasonable rank values are identified. **Advantage:** Can improve Svelte [1]

TABLE 1. *(Continued.)* **Summarization of the attacks and their intrusion detection system.**

			method on detected sinkhole and wormhole use rank of node as feature detection. **Disadvantages:** Unsuitable for constrained devices.
(Others)	Signature	Giberto Fernades et al [12]	Used DSNSF to predict normal behavior of network traffic activity through historical data analysis. **Advantage:** High true positive in detecting anomalies. **Disadvantages:** Not a real-time detection.
Selective Forwarding and Black Hole (Others)	Anomaly	Faouzi Hidoussi et al [13]	Centralized detection system where base station decides on potential intrusions based on the control packets sent from the cluster heads. **Advantage:** Suitable for constrained devices. **Disadvantages:** Unable to detect anomalies of other attacks such as sinkhole and hello flood.

sinkhole, and wormhole attacks are presented. The configuration is conducted to simulate and collect attacks information that will be used by the proposed method.

A. 6LoWPAN COMPRESSION HEADER

Figure 2 a) shows the layout and b) example of compression header in 6LoWPAN packets. An effective header compression is highly dependent on information pertaining to the entire 6LoWPAN network. 6LoWPAN compression header consist a type of identifier where it can identify with a predefined prefix. In RPL routing protocol, information about routing communication is generated when a packet being transmitted to the destinations in the network. This information is stored in the 6LoWPAN compression header and there are 77 routing information available in 6LoWPAN compression header that can be acquired from Wireshark official website[1] using Wireshark tools. The routing information in 6LoWPAN compression header can potentially be used as features to distinguish between normal and abnormal

[1] www.wireshark.org/docs/dfref/6/6lowpan.html

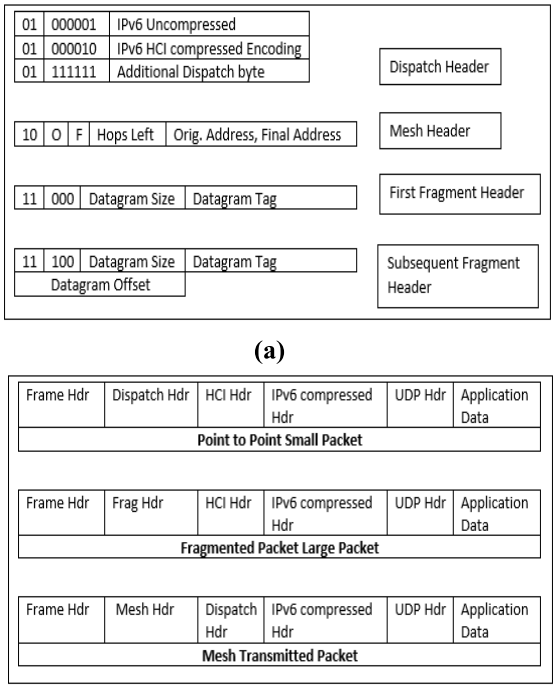


FIGURE 2. (a): 6LoWPAN header layout [14], (b): Example of stack header [14].

activities in 6LoWPAN network due to their unique characteristics. Therefore, in this work, we propose the utilization of routing information in 6LoWPAN compression header as features to detect the routing attacks. However, it should be noted that the features of 6LoWPAN compression header used in this study may differ from others WSN network protocol like Zigbee as this protocol transmitted IPv6 packets over IEEE 802.15.4 networks.

B. PROPOSED SYSTEM FRAMEWORK

Our proposed CHA-IDS is a hybrid based IDS that applied both anomaly and signature based IDS in the scheme. Centralized detection system is implemented in CHA-IDS where all the data will be routed to 6BR device for detection of potential attacks. CHA-IDS utilizes 6LoWPAN compression header as the feature for the machine learning algorithm to learn and classify the type of the attacks. Then, the rule or signature created by the machine learning algorithm is placed at the 6BR. Over time, when a new signature available for the routing attacks, 6BR will be updated with the new rule or signature generated from the new features.

CHA-IDS is divided into four layers as shown in Figure 3. The first layer consists of Sensor Agents (SA) that captures compression header data using Cooja traffic analyzer. The captured data are further analyzed and filtered by Aggregator Agent (AGA) in second layer. The AGA extracts only distinct features that able to distinguish between normal and abnormal network activities. This is followed by the data class

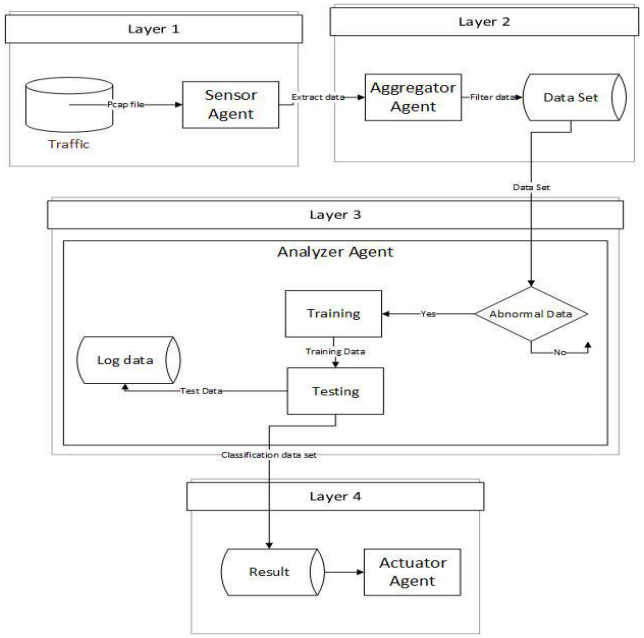


FIGURE 3. Framework of the proposed CHA-IDS.

labelling in Analyzer Agent (ANA) layer. At this layer, data are classified as either normal, hello flood, sinkhole or as wormhole attack. At the final layer, Actuator Agent (ACA) alerts the user if any malicious activities take place. To show how each stage works, a more detailed description of the proposed method framework is described next.

1) LAYER 1, SENSOR AGENT (SA)

SA is responsible for capturing network traffic by collecting received packet data from all nodes in the network. The packet data provide an abstraction of raw and heterogeneous data. In the experiment, we use Cooja traffic analyzer to capture the radio message and save as the packet capture (pcap) file as shown in Figure 4.

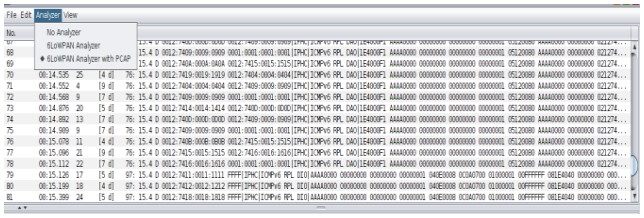


FIGURE 4. 6LoWPAN Analyzer with pcap capture in Cooja simulation tools.

2) LAYER 2, AGGREGATOR AGENT (AGA)

This layer highlights the main contribution of this paper which is finding the significant features that can be used to differentiate between normal and abnormal activities. To perform this task, an experiment that contains normal and abnormal activities are configured. Then, the pcap files for these two activities are filtered to select the data that only related to 6LoWPAN protocol. The filtration process results in 77 routing information in 6LoWPAN compression header. This routing information are saved in Comma Separated Value (CSV) format and then fed into Searching and Feature Selection Algorithm. The Best First Search (BFS) and Greedy Stepwise (GS) are chosen to perform the searching whereas Correlation-based Features Selection (CFS) algorithm are used to evaluate the most significant features that able to differentiate between normal and abnormal network activities (Algorithm 1). It can be noted that we utilizes two different searching algorithms in the experiment. This is to ensure that the selected features are correctly determined which is, the output of the feature selection should be the same for both BFS and GS searching algorithms. If a different output is obtained, the selected features are not significant and may not be able to distinguish between the normal and abnormal activities.

Algorithm 1 Filtering Packet Data Algorithm

1. Input: pcap file
2. get Arguments pcap from network traffic;
3. Loop read each packet;
4. If packet header = 6LoWPAN;
5. Data write 6LoWPAN
6. Choose all info and write to CSV;
7. End if;
8. End loop;

a: FEATURE SELECTION

Best first search with Correlation Features Selection (BFS-CFS) and Greedy stepwise with Correlation Features Selection (GS-CFS) are used to find the significant features to categorize the observed activities into normal and abnormal classes.

Algorithm 2 Greedy Hill Climbing Algorithm [15]

1. Let $s \leftarrow$ start state
2. Expand s by making each possible local change
3. Evaluate each child t of s
4. Let s' child t with highest evaluation $e(t)$
5. If $e(s') \geq e(s)$ then $s \leftarrow s'$, goto 2
6. Return s

BFS will search the space of feature subsets using greedy hill climbing (Algorithm 2). BFS allows backtracking along the search path to move through the search space by making local changes to the current feature subset [15]. The best first search (Algorithm 3) can back-track to a more promising previous subset and continue the search from there [15].

GS performs a greedy forward or backward search through the space of feature subsets. It may start with certain features, all features or randomly point in the space. It will rank the 77 routing information by traversing the space from empty to full (or vice versa) and record the order of the selected

Algorithm 3 Best First Search Algorithm [15]

1. Begin with Open list containing the start state, the CLOSED list empty, and BEST ← start state.
2. Let $s =$ arg max $e(x)$ (get the state from OPEN with the highest evaluation)
3. Remove s from OPEN and add to CLOSED
4. If $e(s) \geq e$ (BEST), then BEST s
5. For each child t of s that is not in the OPEN or CLOSED list, evaluate and add to OPEN
6. If BEST changed in the last set of expansions, goto 2
7. Return BEST

feature [12]. This method (also known as the hill climbing feature selection approach such in Algorithm 2) considers both adding and removing features at each decision point, which allows retracting an earlier decision without keeping explicit track of the search path [12]. It stops when the addition or deletion of any remaining features results in a decrease in the evaluation.

CFS is used to evaluate the worth of a subset of features produce from BFS and GS by considering the individual predictive ability of each feature along with the degree of redundancy between them. Subsets of features that are highly correlated with the class while having a low inter-correlation with each other is formulated in Equation (1).

$$Merit_s = \frac{\overline{kr_{cf}}}{\sqrt{k + k(k-1)_{\overline{r_{ff}}}}} \tag{1}$$

Where $Merit_s$ is the heuristic ''merit'' of an feature subset S containing k feature, $\overline{r_{cf}}$ the average feature-class correlation, and $\overline{r_{ff}}$ the average feature-feature inter-correlation [15], [16].

Equation 1 is a Pearson's correlation coefficient, where the features have been standardized [15]. It shows that the correlation between a class and a feature is a function of the number of features in the composite and the magnitude of the inter-correlations among them, together with the magnitude of the correlations between the components and the outside feature [15].

3) LAYER 3, ANALYZER AGENT (ANA)

Three routing attacks in IoT namely, hello flood, sinkhole and wormhole attack are configured at this layer. The significant features of the attacks as determined from AGA layer are chosen for further analysis. Their packet number are plotted against time to study the behavior of each attack and a set of rule is then devised. Next, classes labelled as ''Normal'', ''Hello Flood'', ''Sinkhole'' and ''Wormhole'' are created based on the revised rule. To classify each attack according to the defined classes, we compare six machine algorithms via WEKA tools to find the best performing algorithm. The algorithms are MLP, SVM, J48, Naïve Bayes, Logistic, and Random Forest.

4) LAYER 4, ACTUATOR AGENT (ACA)

This agent will execute an action by giving an alert to the users. It has simple behavior that responds to the ANA process by comparing them with a threshold value. If the result from the ANA process exceeded the threshold value, an alarm will trigger. In this paper, a threshold value of 10% is set (Algorithm 4) to reduce false alarm.

Algorithm 4 Monitoring & Alert Threshold

1. Check: TP rate percentage
2. If hello flood $>$ 10 %
3. Alert attack occur: hello flood
4. End if
5. If sinkhole $>$ 10 %
6. Alert attack occur: sinkhole
7. End if
8. If wormhole $>$ 10%
9. Alert attack occur: wormhole
10. End if

C. EVALUATION METRICS

To verify the ability of the proposed method, two sets of evaluation metric are devised. The first evaluates the performance of machine learning to classify the routing attacks, and the second measure the energy and memory consumptions.

TABLE 2. Terminology and derivations.

Terminology		Derivations
TP(True Positive)	:	Total examples predicted as true that are actually true
FP(False Positive)	:	Total examples predicted as true that are actually false
TN(True Negative)	:	Total examples predicted as false that are actually false
FN(False Negative)	:	Total examples predicted as false that are actually true

1) MACHINE LEARNING CLASSIFICATIONS

As previously mentioned in Section 3.2.3, six machine learning algorithms (i.e. MLP, SVM, J48, Naïve Bayes, Logistic, and Random Forest) are evaluated to determine the best performing algorithm to classify the routing attacks using WEKA workbench. True Positive rate (TP), False Positive rate (FP) Mean Absolute Error (MAE), Root Means Squared error (RMSE), Relative Absolute error (RAE), precision, recall and accuracy are computed and expressed as in equations (2, 3, 4, 5, 6, 7, 8, and 9). The terminology and derivations of TP, FP, TN and FN are given in Table 2.

$$Prate = \frac{TP}{TP + FN} \tag{2}$$

$$FPrate = \frac{FP}{FP + TN} \tag{3}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n} \left| y_i - \hat{y}_i \right| \tag{4}$$

TABLE 3. Operating conditions of the Tmote sky.

Conditions	Mon	NOM	Max	Unit
Voltage	2.1	–	3.6	V
MCU on, Radio RX	–	21.8	23	mA
MCU on, Radio TX	–	19.5	21	mA
MCU on, Radio off	–	1800	2400	µA
MCU idle, Radio off	–	54.5	1200	µA
MCU standby	–	5.1	21.0	µA

TABLE 4. Memory capacity in Tmote sky

Types of memory	Capacity
RAM	10 Kbytes
ROM	48 Kbytes
External Storage	1 Mbytes

$$y_i = Actual and \hat{y}_i = Predicted$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2} \quad (5)$$

$$RAE_i = \frac{\sum_{j=1}^{n}\left|P_{(ij)} - T_j\right|}{\sum_{j=1}^{n}\left|T_j - \overline{T}\right|} \quad where\ \overline{T} = \frac{1}{n}\sum_{j=1}^{n} T_j \quad (6)$$

$P_{(ij)}$: *value predicted by i for sample case j*

T_j : *target value for sample case j*

$$Precision = \frac{TP}{TP + FP} \quad (7)$$

$$Recall = \frac{TP}{TP + FN} \quad (8)$$

$$Accuracy = \frac{TN + TP}{TP + FN + FP + TN} \quad (9)$$

2) ENERGY & MEMORY CONSUMPTIONS

Typically, the nodes in IoT are battery powered with limited memory. To evaluate the feasibility of the proposed method in IoT environment, a constrained device known as Tmote Sky is utilized. The operating conditions and memory capacity of the Tmote Sky are tabulated in Table 3 and Table 4 [1]. Contiki Powertrace [1], [2] is used to evaluate the power consumed by the proposed CHA-IDS, SVELTE [2] and Pongle's Wormhole IDS [1] implemented in the Tmote Sky. For the measurement of power consumption, we only considered the device for the client node in the network. Equation (10) shows the energy usage for 30 min per node, whereas equation (11) calculates the average of power consumed per second. Lastly, the total size of memory consumed in the experiment is calculated according to Equation (12).

$$Energy\,(mJ) = \begin{pmatrix} Transmit \times 19.5\ mA + listen \times 21.8\ mA \\ + CPU \times 1.8mA + LPM \times 0.0545\ mA \\ \times (3V \div 4096) \times 8 \end{pmatrix} \quad (10)$$

$$Power\,(mW) = \frac{Energy\,(mJ)}{Time\,(s)} \quad (11)$$

$$Total\ size = text + data + bss \quad where\ bss\ is\ prezeroed\ RAM \quad (12)$$

D. EXPERIMENTAL SETUP

The three routing attacks in IoT namely, hello flood, sinkhole, and wormhole attacks are launched in Contiki's network simulator known as Cooja.[2] Due to limited budget, we only implement part of the simulation using real hardware platform provided by the Contiki while the additional nodes were tested via simulation. Contiki has proven to be a powerful toolbox for building complex wireless systems and have shown a realistic result as in the real network [1], [2]. Furthermore, all the data used in the simulation are from the real network environment. In the simulation, Tmote Sky is used as client node and Cooja mote is used as 6BR or sink node. Each attack is triggered separately and the experiment is divided into two parts. Part 1: Data gathering to determine features that can be used to create a dataset. All attacks are simulated for 30 minutes and radio message's communication of each node are logged. Part 2: Dataset is collected at 10, 20, 30 minutes of the simulation period for each attack [1], [2]. In the experiment, a lossy configuration setting is selected to resemble the actual 6LoWPAN network. Other than that, Unit Disk Graph Medium (UDGM) loss model is used as Cooja's default radio model [1], [2]. UDGM models the communication range as a circle in which, only the nodes inside the circle can communicate.

At the beginning of the experiment, a normal 6LoWPAN network is configured without the presence of attacker node. Subsequently, an abnormal 6LoWPAN network is configured with the attacker node. This is done to distinguish between normal and abnormal network activities. Next, the configuration for combined attack are similar with abnormal 6LoWPAN network but have different way of evaluation where three of attacks such hello flood, sinkhole and wormhole are launch in the same time and network randomly to create new anomaly attack pattern in the network. Details configuration of the attacker nodes is described as follows:

Figure 5 shows the configuration of hello flood attack where node 8 is the malicious node broadcasting the hello message. The malicious node can present itself as a neighbor node to numerous nodes by broadcasting the message with robust routing metrics to enter into the network. To start the attack in RPL, attacker broadcasts the information about DODAG [5] using DIO messages.

[2] www.contiki-os.org

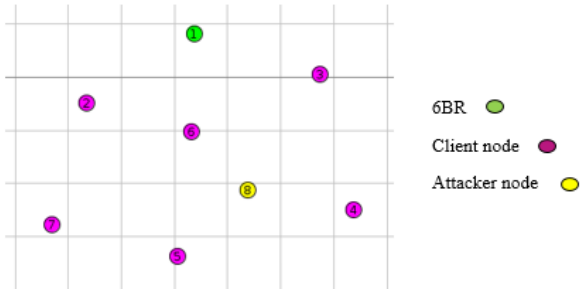


FIGURE 5. Network configuration for hello flood.

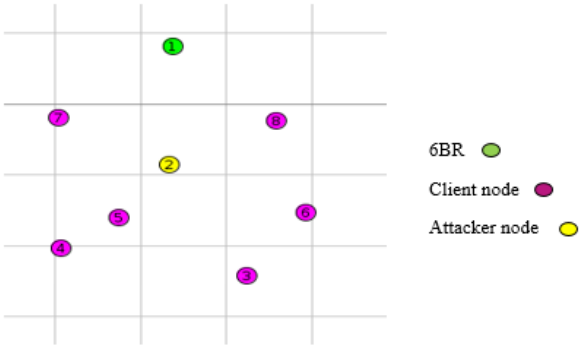


FIGURE 6. Network configuration for sinkhole.

Figure 6, the network configuration for sinkhole is presented. Node 2 which is the attacker node increases its rank and advertises itself to the network in order to convince the nearest neighbor to select node 2 as a parent or sink node. In RPL, routing rule states that rank is strictly increased in the downstream direction and strictly decreased in the upstream direction. These rules are to prevent the nodes from creating a non-optimized path or loop path. The RPL creates node rank as its unique parameter for easily choosing and maintaining the optimized path.

FIGURE 7. Network configuration for wormhole (encapsulation).

Two wormhole attacks using encapsulation and packet relay techniques are configured in Figure 7 and Figure 8. As shown in Figure 7, three nodes are used to create the wormhole encapsulation attack wherein two malicious nodes and one intermediate node are set to establish a tunnel between them [1]. On the other hand, Figure 8 shows how the wormhole using packet relay technique is set up.

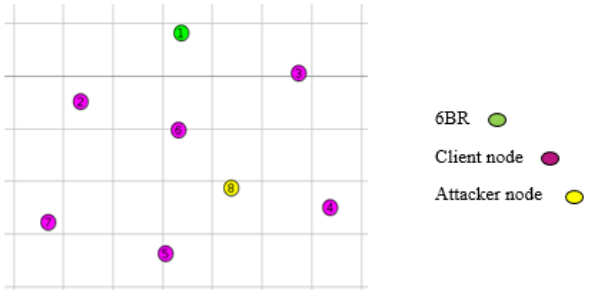


FIGURE 8. Network configuration for wormhole (packet relay).

Malicious node relays the packets between two nearest nodes to convince them that they are neighbors [1].

IV. EXPERIMENTAL RESULTS AND EVALUATIONS

A. SELECTION OF SIGNIFICANT FEATURE

At the AGA layer, 77 routing information of 6LoWPAN compression header are studied as features to detect and classify the routing attacks. However, some of the 77 information are unreliable and contain noises that may lead to the wrong classification of the routing attacks. Therefore, we utilize BFS-CFS and GS-CFS searching algorithms to select the most significant features from the 77 routing information of the 6LoWPAN compression header in AGA layer (Layer 2). The result from the AGA layer shows that only five routing information are significant to detect abnormal activities in the network, thus, they are selected as features in this study as listed in Table 5.

TABLE 5. List of significant features to detect abnormal activities.

Attribute	Description	Type
6LoWPAN.dst	Destination port	Unsigned integer, 2 bytes
6LoWPAN.cid	Context identifier	Boolean
6LoWPAN.dci	Destination context identifier	Unsigned integer, 1 byte
6LoWPAN.next	Next header	Unsigned integer, 1 byte
6LoWPAN.pattern	pattern	Unsigned integer, 1 byte

TABLE 6. List of significant features associated with hello flood, sinkhole and wormhole attacks.

Type of attack	Searching Algorithms	Selected Features
Hello Flood	Best First	dst, pattern
	Greedy Stepwise	dst, pattern
Sinkhole	Best First	dst, cid, dci, next, pattern
	Greedy Stepwise	dst, cid, dci, next, pattern
Wormhole	Best First	dst, dci, cid, pattern
	Greedy Stepwise	dst, dci, cid, pattern

According to the five significant features highlighted in Table 5, we perform a more detail test to associate the significant features with each of the routing attack utilizes in this work i.e. hello flood, sinkhole and wormhole attacks. The results from this test are presented in Table 6. The features

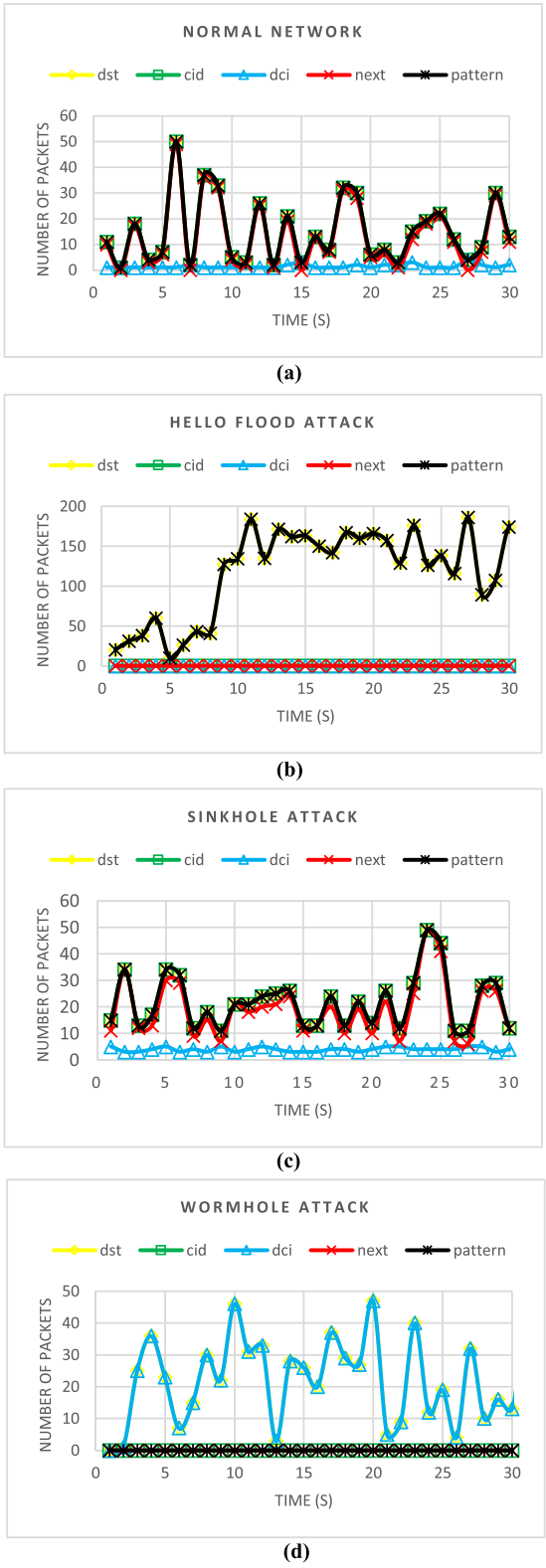


FIGURE 9. (a): Network pattern in normal based on numbers of packets against time, (b): Network pattern in hello flood based on number of packets against time, (c): Network pattern in sinkhole based on number of packets against time, (d): Network pattern in wormhole based on number of packets against time.

for each attack are selected when both BFS and GS searching algorithms results in the same features indicating the features are stable and robust for machine learning algorithm to identify the routing attacks.

Following the selection of significant features for each attack, the duplicated data are removed from the packet. Then, the number of packets of the significant features are plotted against time for normal network and during the routing attacks as depicted in Figure 9 (a, b, c, d). From these figures, the number of packet per second for each significant features changes differently that are unique according to the routing attacks. Thus, we derived a set of condition for each attack based on the number of packet per second in the significant features as summarized in Table 7. These conditions are used to train and guide the machine learning algorithm for classification of routing attacks.

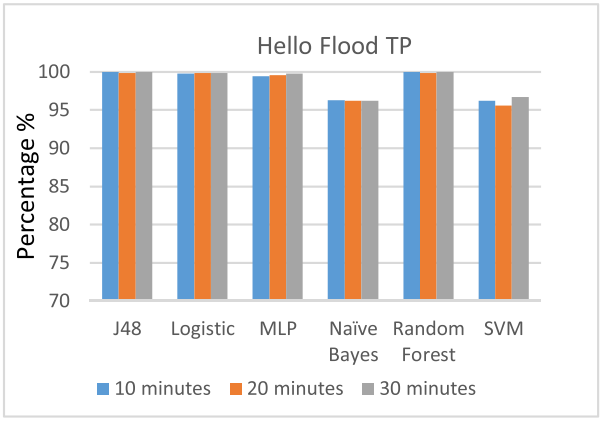
TABLE 7. Summary of condition for each routing attack based on the number of packets in the significant features.

Destination port (dst)	Context identifier (cid)	Destination context identifier (dci)	Next header (next)	Pattern	Class
dst > 40	-	-	-	pattern > 40	Hello Flood
dst > 3	-	6 > dci > 2	next > 5	-	Sinkhole
dst > 0	cid = 0	dci = dst or dci > 0	-	pattern > 0	Wormhole

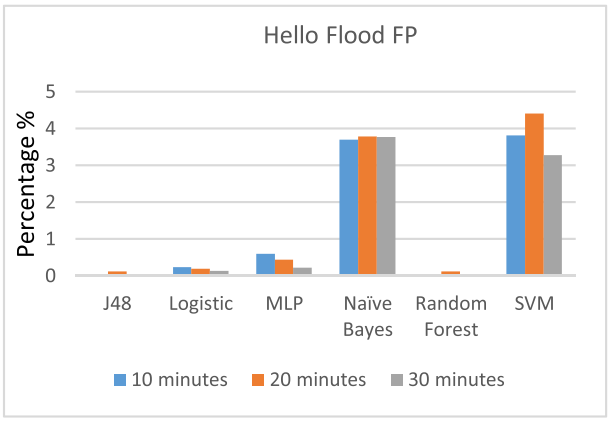
B. SELECTION OF MACHINE LEARNING ALGORITHM FOR ATTACKS CLASSIFICATION

In this study, the best algorithm to classify the routing attacks according to the condition set in Table 7 is obtained by analyzing six machine learning algorithms i.e. MLP, SVM, J48, Naïve Bayes, Logistic, and Random Forest. TP and FP rates are computed to examine whether the features are correctly or incorrectly classified. To start with, the performance of the machine learning algorithms against the individual attack are evaluated as depicted in (Figure 10 (a), (b)), (Figure 11(a),(b)) and (Figure 12(a),(b)). The result shows that J48 and Random Forest has the highest TP rate and manage to achieve 100% detection of Hello Flood as depicted in Figure 10(a). In the Sinkhole attack, Random Forest again attains a 100% of TP rate for the entire simulation period while the performance of J48 increases at 20 and 30 minutes of the simulation period to gain from 99% to 100% (see Figure 11(a)). Similar performances are also observed in Wormhole attack wherein J48 and Random Forest achieve 100% detection for the entire simulation period while other algorithms achieve 99.95% and manage to improve their TP rate at 30 minutes of the simulation period to obtain 99.98% (see Figure 12 (a)).

Following the TP and FP evaluations, other evaluation metrics such as precision, recall and accuracy are also



(a)



(b)

FIGURE 10. (a): TP rate in 10, 20, 30 minutes for hello flood, (b): FP rate in 10, 20, 30 minutes for hello flood.

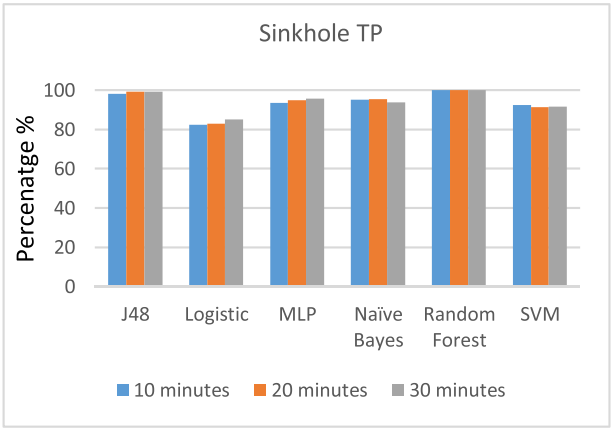
TABLE 8. Hello flood in 30 minutes.

Algorithm	Precision (%)	Recall (%)	Accuracy (%)
J48	100	100	100
Logistic	99.9	99.9	99.9
MLP	99.8	99.8	99.8
Naïve Bayes	97.1	96.2	96.2
Random Forest	100	100	100
SVM	96.8	96.7	96.8

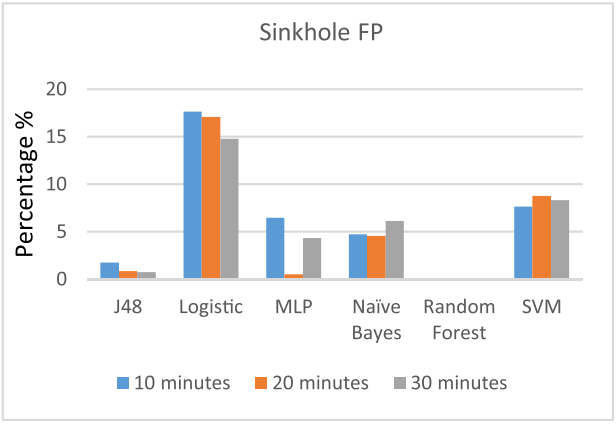
TABLE 9. Sinkhole in 30 minutes.

Algorithm	Precision (%)	Recall (%)	Accuracy (%)
J48	99.3	99.3	99.2
Logistic	77.2	85.2	85.2
MLP	95.6	95.7	96
Naïve Bayes	93.7	93.9	93.8
Random Forest	100	100	100
SVM	91.1	91.7	91.6

considered in the experiment. In Table 8, J48 and Random Forest achieved 100% of precision, recall and accuracy when tested on Hello Flood dataset. Again, Random Forest achieved 100% of precision, recall and accuracy in detecting Sinkhole attack (Table 9). Lastly, all algorithm managed to achieve 100% of precision, recall and accuracy in detecting Wormhole attack as summarized in Table 10.



(a)



(b)

FIGURE 11. (a): TP rate in 10, 20, and 30 minutes for sinkhole, (b): FP rate in 10, 20, and 30 minutes for sinkhole.

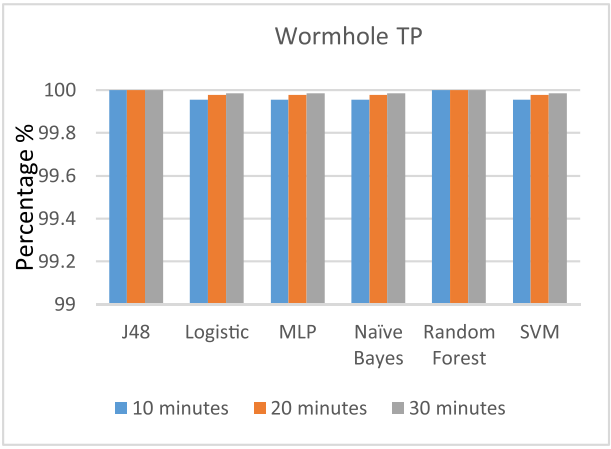
TABLE 10. Wormhole in 30 minutes.

Algorithm	Precision (%)	Recall (%)	Accuracy (%)
J48	100	100	100
Logistic	100	100	100
MLP	100	100	100
Naïve Bayes	100	100	100
Random Forest	100	100	100
SVM	100	100	100

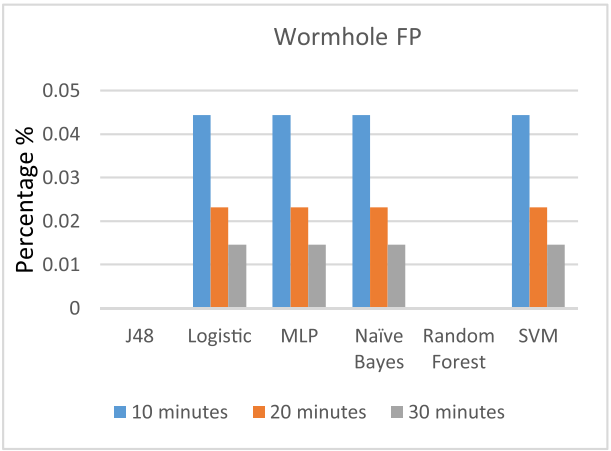
TABLE 11. Evaluations of combined dataset with various algorithms.

Algorithm	Correctly Classified %	Incorrectly Classified %	TP rate	FP rate	MAE	RMSE	RAE %
J48	**99.4444**	**0.5556**	**0.994**	**0.002**	**0.0041**	**0.0529**	**1.259**
Logistic	94.1667	5.8333	0.942	0.042	0.0498	0.1582	15.387
MLP	96.6667	3.3333	0.967	0.018	0.0338	0.1245	10.444
Naïve Bayes	97.2222	2.7778	0.972	0.013	0.0286	0.1155	8.834
Random Forest	99.4444	0.5556	0.994	0.003	0.0087	0.0547	2.680
SVM	93.3333	6.6667	0.933	0.058	0.0333	0.1826	10.292

Apart from the evaluation of individual attack, the combination of the three attacks is also examined. The performance of six machine learning algorithms in detecting the combined attacks is summarized in Table 11. Overall, J48 achieved the



(a)



(b)

FIGURE 12. (a): TP rate in 10, 20, and 30 minutes for wormhole, (b): FP rate in 10, 20, and 30 minutes for wormhole.

highest TP rate with 99.4444% and the lowest MAE, which is 0.0041 among all machine learning algorithms. Random Forest algorithm is ranked in second with a similar TP rate as J48 but obtained a slightly higher MAE with 0.0087. Table 12 (a) to (f) show the confusion matrices of attacks classification for each machine learning algorithm.

From the evaluation of individual and combination attacks, J48 algorithm achieved the highest classification performance among all machine learning algorithms when utilize our proposed features as input. Therefore, J48 is selected to detect the attacks in the proposed CHA-IDS. The rules of J48 implemented in the client node as an IDS signature is depicted in Figure 13.

C. COMPARISON WITH EXISTING APPROACH

Two mitigation approaches i.e. SVELTE [1] and Pongle's IDS [1] are considered in this evaluation. The TP rate, energy and memory consumption of each approach are compared to inspect their feasibility in real environment implementation.

TABLE 12. Confusion matrices of various classification algorithms. (a) J48, (b) Logistic, (c) MLP, (d) Naïve Bayes, (e) Random forest, (f) SVM.

(a)

Normal	Hello Flood	Sinkhole	Wormhole	Classified as
166	0	0	1	Normal
0	**125**	0	0	Hello Flood
0	0	**33**	0	Sinkhole
0	1	0	34	Wormhole

(b)

Normal	Hello Flood	Sinkhole	Wormhole	Classified as
163	0	3	1	Normal
0	**125**	0	0	Hello Flood
16	0	17	0	Sinkhole
1	0	0	34	Wormhole

(c)

Normal	Hello Flood	Sinkhole	Wormhole	Classified as
162	0	5	1	Normal
0	**125**	0	0	Hello Flood
4	0	29	0	Sinkhole
3	0	0	32	Wormhole

(d)

Normal	Hello Flood	Sinkhole	Wormhole	Classified as
162	0	4	1	Normal
0	**125**	0	0	Hello Flood
5	0	28	0	Sinkhole
0	0	0	35	Wormhole

(e)

Normal	Hello Flood	Sinkhole	Wormhole	Classified as
166	0	0	1	Normal
0	**125**	0	0	Hello Flood
0	0	**33**	0	Sinkhole
1	0	0	34	Wormhole

(f)

Normal	Hello Flood	Sinkhole	Wormhole	Classified as
167	0	0	1	Normal
3	122	0	0	Hello Flood
18	0	15	0	Sinkhole
3	0	0	32	Wormhole

1) TRUE POSITIVE RATE

Figure 14 and 15 show the comparison of TP rate between the proposed CHA-IDS with prior methods for individual and combined attacks respectively. The proposed CHA-IDS achieved an overall of 99% of TP rate to detect three types of attack individually. SVELTE obtained 100% of TP rate in detecting hello flood and sinkhole attacks but a lower TP rate with 80% is observed in detecting wormhole attack. Pongle's IDS only capable in detecting wormhole attack but fails to detect hello flood and sinkhole attacks. For combined attack, CHA-IDS scored 99% of TP rate which is the highest

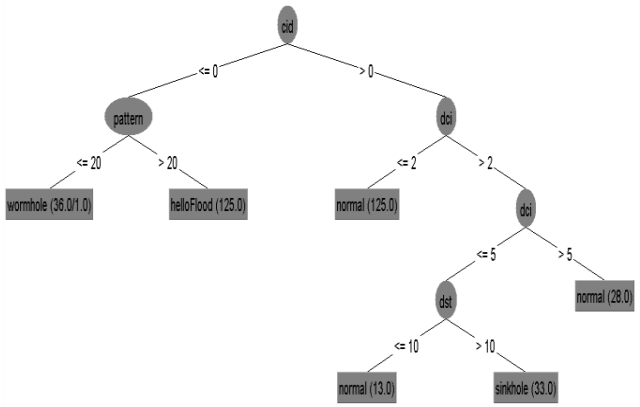


FIGURE 13. J48 classification rules tree.

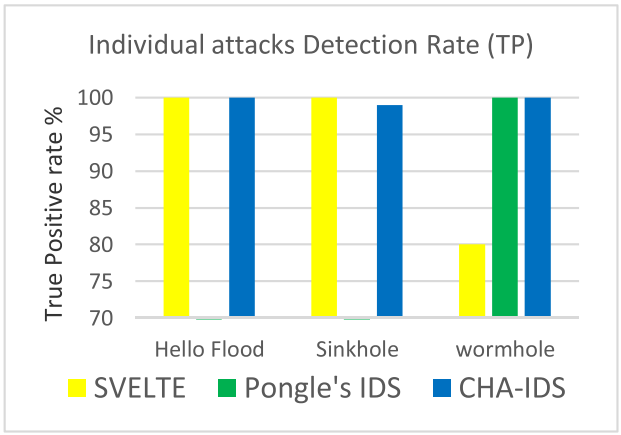


FIGURE 14. Comparison detection for individual attacks.

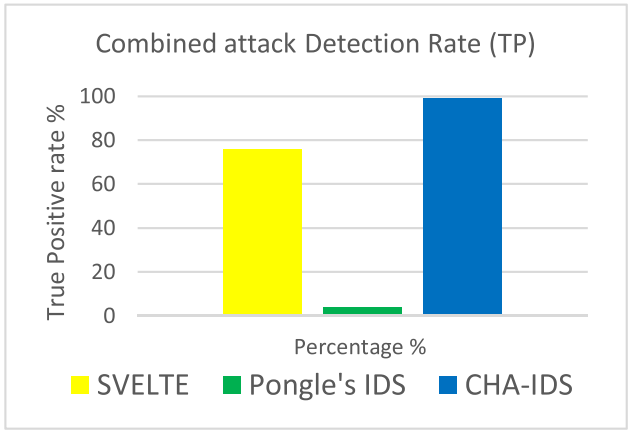


FIGURE 15. Comparison detection for combination attack (hello flood, sinkhole, wormhole).

among all, while SVELTE only detects 76% of the attacks. The lowest performance among all is observed in Pongle's IDS method with minimal TP rate in detecting the combined attacks. Figure 16 shows the detection time for each IDS to trigger the first alert toward the attacks.

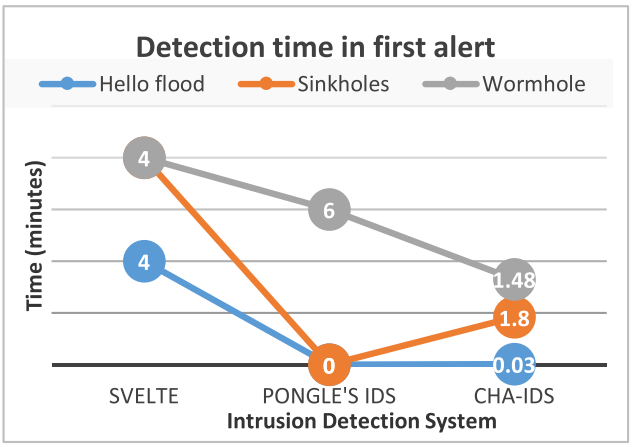


FIGURE 16. Detection time in first alert for each IDS.

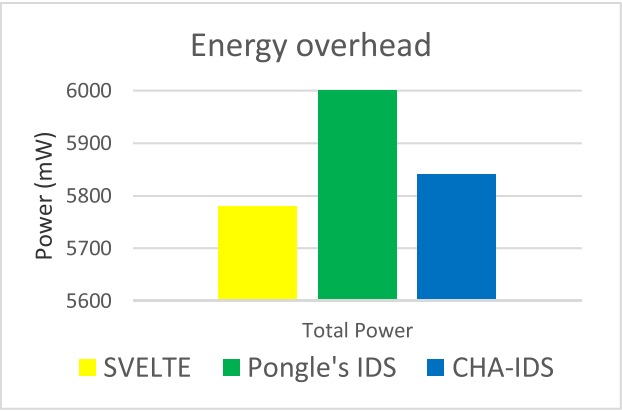


FIGURE 17. Comparison of energy usages.

2) ENERGY OVERHEAD

Evaluation of energy consumption is important in estimating the node lifetime. It is crucial for applications with limited access to continuous power supply. Figure 17 shows the comparison of energy overhead for each IDS in a Tmote Sky node running for 30 minutes. The lowest energy consumption is seen in SVELTE with 5780 mW followed by the proposed CHA-IDS with 5840 mW. The highest energy is consumed by Pongle's Wormhole IDS with more than 6000 mW. SVELTE has the lowest energy overhead because the client node in SVELTE only collects and share information to 6BR. Contrarily, CHA-IDS extracts and filters the number of packets received. The Pongle's IDS, on the other hand, calculates RSSI values, collect and also share information to the 6BR. These processes making them more power hungry compared to the others.

3) MEMORY CONSUMPTION

Typically, constrained devices in IoT applications have limited memory, thus, memory consumption is evaluated to assess the feasibility of IDS methods in constrained devices. In this assessment, SVELTE consumed the lowest memory usage with 44.1Kbytes followed by Pongle's IDS

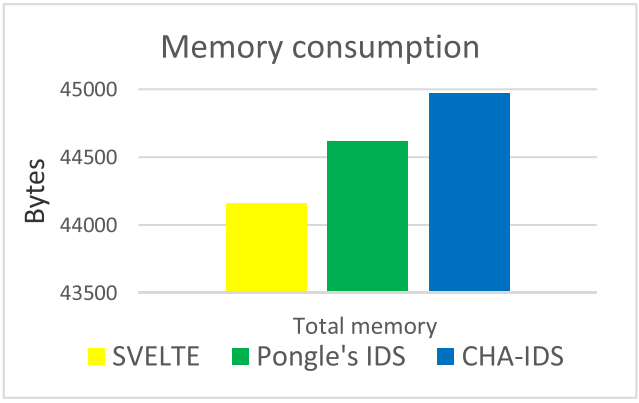


FIGURE 18. Memory consumption usages.

with 44.6Kbytes and CHA-IDS with 44.9Kbytes as shown in Figure 18.

V. DISCUSSIONS

In this paper, we examine the ability of 6LoWPAN compression header as features to detect three types of routing attacks in IoT namely hello flood, sinkhole and wormhole (encapsulation and packet relay) attacks. The features are tested with individual attack and when the attacks are combined. The chosen feature is also compared with two existing methods called SVELTE and Pongle's IDS.

When the individual attack is performed, SVELTE successfully detects hello flood and sinkhole attacks. For wormhole attacks, SVELTE detects 80% of the wormhole encapsulation attack but fails to detect any of the wormhole packet relay. This is because SVELTE is based on RPL and ranking information. The packet relay attack causes the disruption in the network and drop the incoming packet that prevents SVELTE from receiving the information about the client node ranking, thus, fails to work as intended. This also affects the performance of SVELTE in detecting the combination of routing attacks when only detects 76% of the attacks during the experiment.

Pongle's IDS method is based on RSSI value and node's location. It works by comparing the actual RSSI value with the location published by the nodes. Malicious node in wormhole attack interrupts the route selection process by deceiving their location information. However, the signal strength is not easily manipulated by the attack. Therefore, Pongle's IDS can effectively detects the wormhole attack when there is substantial difference in the location information between the nodes. However, Pongle's IDS unable to detect any of the hello flood and sinkhole attacks because these attacks do not manipulate location information. Thus, in the combined attack Pongle's IDS attained the lowest TP rate among all method with only detects 4% of the combined attacks.

The proposed CHA-IDS demonstrate a high capability in detecting both individual and combination of routing attacks during the experiment. This performance is expected since CHA-IDS utilizes part of 6LoWPAN compression header as features to detect the attacks. The 6LoWPAN compression header stores the routing information which affected when the abnormal routing activities exist in the network. This can be seen when the number of packets in the compression header changes with a distinctive pattern according to the type of the attacks. Accordingly, CHA-IDS takes the advantage of the distinctive patterns to detect the abnormalities in the network.

In the experiment, energy overhead and memory consumption of CHA-IDS, SVELTE and Pongle's IDS are also evaluated. Among the methods, it can be observed that CHA-IDS is not the most efficient method in term of energy and memory consumption. However, CHA-IDS offers the best performance to effectively detect the three routing attacks. For SVELTE and Pongle's IDS to achieve the same performance as CHA-IDS, these two attacks need to be combined. Consequently, the combination of SVELTE and Pongle's IDS may consume more energy and memory than CHA-IDS. Therefore, CHA-IDS offers the best balance between the quality of the performance and the consumption of energy and memory.

VI. CONCLUSION & FUTURE WORK

Nodes in the 6LoWPAN network of IoT are exposed to a variety of intrusion threats. In this paper, an IDS to detect routing attacks namely, hello flood, sinkhole and wormhole attacks as well as their combination is proposed. The proposed method named as CHA-IDS utilizes 6LoWPAN compression header as features. The 6LoWPAN compression header consists of 77 information regarding the routing details. From this information, we used BFS-CFS and GS-CFS searching algorithms to select the most significant information as input features into a machine learning algorithm. This results in five features being selected which are destination port, context identifier, destination context identifier, next header and pattern. We tested six machine learning algorithms i.e. MLP, SVM, J48, Naïve Bayes, Logistic, and Random Forest using the five selected features to detect and classify the routing attacks. Among these algorithms, J48 algorithm is chosen for the proposed CHA-IDS as J48 shows the best performance among all algorithms in detecting the routing attacks with 99% of true positive rate. During the experiment, the consumption of energy and memory of CHA-IDS are 5840mW and 44.9kB respectively. CHA-IDS effectively detected both individual and new anomaly attack that created by combination of the routing attacks and outperformed other 6LoWPAN based IDS called SVELTE and Pongle's IDS in which, only detect specific attack.

CHA-IDS has demonstrated a high capability in detecting routing attacks. However the proposed method only managed to detect the occurrence of the attacks but unable to precisely identify the attacker. In future, the proposed CHA-IDS can be extended by collaborating it with distribute IDS to identify the attacker in the 6LoWPAN network. Other than that, the proposed CHA-IDS can be improved to detect other attacks in IoT such as Sybil, Clone ID and Version

Number and Local Repair attacks. Other machine learning algorithms such as fuzzy Q-learning [6] and density-based fuzzy imperialist [5] can be tested with the proposed features to improve the accuracy of the detection.

REFERENCES

[1] P. Pongle and G. Chavan, "Real time intrusion and wormhole attack detection in Internet of Things," *Int. J. Comput. Appl.*, vol. 121, no. 9, pp. 1–9, 2015. [Online]. Available: https://doi.org/10.5120/21565-4589

[2] S. Raza, L. Wallgren, and T. Voigt, "SVELTE: Real-time intrusion detection in the Internet of Things," *Ad Hoc Netw.*, vol. 11, no. 8, pp. 2661–2674, 2013. [Online]. Available: https://doi.org/10.1016/j.adhoc.2013.04.014

[3] T. Sherasiya, H. Upadhyay, and H. B. Patel, "A survey: Intrusion detection system for Internet of Things," *Int. J. Comput. Sci. Eng.*, vol. 5, no. 2, pp. 91–98, 2016. [Online]. Available: http://www.iaset.us/view_archives.php?year=2016&id=14&jtype=2&page=2

[4] M. A. Aydın, A. H. Zaim, and K. G. Ceylan, "A hybrid intrusion detection system design for computer network security," *Comput. Elect. Eng.*, vol. 35, no. 3, pp. 517–526, 2009. [Online]. Available: https://doi.org/10.1016/j.compeleceng.2008.12.005

[5] S. Shamshirband, A. Amini, N. B. Anuar, M. L. M. Kiah, Y. W. Teh, and S. Furnell, "D-FICCA: A density-based fuzzy imperialist competitive clustering algorithm for intrusion detection in wireless sensor networks," *Meas., J. Int. Meas. Confederation*, vol. 55, pp. 212–226, Sep. 2014. [Online]. Available: https://doi.org/10.1016/j.measurement.2014.04.034

[6] S. Shamshirband *et al.*, "Co-FAIS: Cooperative fuzzy artificial immune system for detecting intrusion in wireless sensor networks," *J. Netw. Comput. Appl.*, vol. 42, pp. 102–117, Jun. 2014. [Online]. Available: https://doi.org/10.1016/j.jnca.2014.03.012

[7] C. Kolias, G. Kambourakis, A. Stavrou, and S. Gritzalis, "Intrusion detection in 802.11 networks: Empirical evaluation of threats and a public dataset," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 184–208, 1st. Quart., 2016. [Online]. Available: https://doi.org/10.1109/COMST.2015.2402161

[8] Z. Shelby and C. Bormann, *6LoWPAN: The Wireless Embedded Internet*, vol. 43. West Sussex, U.K.: Wiley, 2011.

[9] V. P. Singh, A. S. A. Ukey, and S. Jain, "Signal strength based hello flood attack detection and prevention in wireless sensor networks," *Int. J. Comput. Appl.*, vol. 62, no. 15, pp. 1–6, 2013. [Online]. Available: https://doi.org/10.5120/10153-4987

[10] K. Grgic, D. Zagar, and V. K. Cik, "System for malicious node detection in IPv6-based wireless sensor networks," *J. Sensors*, vol. 2016, 2016, Art. no. 6206353. [Online]. Available: https://doi.org/10.1155/2016/6206353

[11] G.-H. Lai, "Detection of wormhole attacks on IPv6 mobility-based wireless sensor network," *EURASIP J. Wireless Commun. Netw.*, vol. 2016, Dec. 2016, Art. no. 274. [Online]. Available: https://doi.org/10.1186/s13638-016-0776-0

[12] G. Fernandes, Jr., J. J. P. C. Rodrigues, and M. L. Proença, Jr., "Autonomous profile-based anomaly detection system using principal component analysis and flow analysis," *Appl. Soft Comput.*, vol. 34, pp. 513–525, Sep. 2015. [Online]. Available: https://doi.org/10.1016/j.asoc.2015.05.019

[13] F. Hidoussi, H. Toral-Cruz, D. E. Boubiche, K. Lakhtaria, A. Mihovska, and M. Voznak, "Centralized IDS based on misuse detection for cluster-based wireless sensors networks," in *Wireless Pers. Commun.*, vol. 85, no. 1, pp. 207–224, 2015. [Online]. Available: https://doi.org/10.1007/s11277-015-2774-2

[14] Mulligan, G., "The 6LoWPAN architecture," in *Proc. 4th Workshop Embedded Netw. Sensors (EmNets)*, 2007, pp. 78–82. [Online]. Available: https://doi.org/10.1145/1278972.1278992

[15] M. A. Hall, "Correlation-based feature selection for machine learning," The Univ. Waikato, Hamilton, New Zealand, Tech. Rep. 1999, Apr. 1999.

[16] M. A. Hall and G. Holmes, "Benchmarking attribute selection techniques for discrete class data mining," *IEEE Trans. Knowl. Data Eng.*, vol. 15, no. 6, pp. 1437–1447, Nov./Dec. 2003. [Online]. Available: https://doi.org/10.1109/TKDE.2003.1245283

[17] A. Rghioui, A. Khannous, and M. Bouhorma, "Denial-of-service attacks on 6LoWPAN-RPL networks: Issues and practical solutions," *J. Adv. Comput. Sci. Technol.*, vol. 3, no. 2, p. 143, 2014. [Online]. Available: https://doi.org/10.14419/jacst.v3i2.3321

[18] P. Kasinathan, C. Pastrone, M. A. Spirito, and M. Vinkovits, "Denial-of-service detection in 6LoWPAN based Internet of Things," in *Proc. IEEE 9th Int. Conf. Wireless Mobile Comput., Netw. Commun.*, Oct. 2013, pp. 600–607. [Online]. Available: https://doi.org/10.1109/WiMOB.2013.6677419

[19] P. Pongle and G. Chavan, "A survey: Attacks on RPL and 6LoWPAN in IoT," in *Proc. Int. Conf. Pervas. Comput., Adv. Commun. Technol. Appl. Soc. (ICPC)*, Jan. 2015, pp. 1–6. [Online]. Available: https://doi.org/10.1109/PERVASIVE.2015.7087034

[20] L. Wallgren, S. Raza, and T. Voigt, "Routing attacks and countermeasures in the RPL-based internet of things," *Int. J. Distrib. Sensor Netw.*, vol. 9, no. 8, 2013, Art. no. 794326. [Online]. Available: https://doi.org/10.1155/2013/794326

[21] A. Morais and A. Cavalli, "A distributed intrusion detection scheme for wireless ad hoc networks," in *Proc. SAC*, 2012, pp. 556–562. [Online]. Available: https://doi.org/10.1145/2245276.2245382

[22] Z. Yu and J. J. P. Tsai, "A framework of machine learning based intrusion detection for wireless sensor networks," in *Proc. IEEE Int. Conf. Sensor Netw., Ubiquitous, Trustworthy Comput. (SUTC)*, Jun. 2008, pp. 272–279. [Online]. Available: https://doi.org/10.1109/SUTC.2008.39

[23] M. Mantere, M. Sailio, and S. Noponen, "A module for anomaly detection in ICS networks," in *Proc. 3rd Int. Conf. High Confidence Netw. Syst. (HiCoNS)*, 2014, pp. 49–56. [Online]. Available: https://doi.org/10.1145/2566468.2566478

[24] M. Di and E. M. Joo, "A survey of machine learning in wireless sensor netoworks from networking and application perspectives," in *Proc. 6th Int. Conf. Inf., Commun. Signal Process. (ICICS)*, 2007, pp. 1–5. [Online]. Available: https://doi.org/10.1109/ICICS.2007.4449882

[25] G. W. Kibirige and C. Sanga, "A survey on detection of sinkhole attack in wireless sensor networks," *Int. J. Comput. Sci. Inf. Secur.*, vol. 13, no. 5, pp. 48–52, May 2015.

[26] S. M. H. Bamakan, H. Wang, T. Yingjie, and Y. Shi, "An effective intrusion detection framework based on MCLP/SVM optimized by time-varying chaos particle swarm optimization," *Neurocomputing*, vol. 199, pp. 90–102, Jul. 2015. [Online]. Available: https://doi.org/10.1016/j.neucom.2016.03.031

[27] M. Xie, S. Han, B. Tian, and S. Parvin, "Anomaly detection in wireless sensor networks: A survey," *J. Netw. Comput. Appl.*, vol. 34, no. 4, pp. 1302–1325, 2011. [Online]. Available: https://doi.org/10.1016/j.jnca.2011.03.004

[28] Z. K. Wazir, X. Yang, M. Y. Aalsalem, and A. Quratulain, "Comprehensive study of selective forwarding attack in wireless sensor networks," *Int. J. Comput. Netw. Inf. Secur.*, vol. 3, no. 1, pp. 1–10, 2011.

[29] S. K. Saini and M. Gupta, "Detection of malicious cluster head causing hello flood attack in LEACH protocol in wireless sensor networks," *Int. J. Appl. Innov. Eng. Manage.*, vol. 3, no. 5, pp. 384–391, 2014.

[30] J. Granjal, E. Monteiro, and J. Sa Silva, "Enabling network-layer security on IPv6 wireless sensor networks," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, Dec. 2015, pp. 1–6.

[31] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Generat. Comput. Syst.*, vol. 29, no. 7, pp. 1645–1660, 2013. [Online]. Available: https://doi.org/10.1016/j.future.2013.01.010

[32] D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac, "Internet of things: Vision, applications and research challenges," *Ad Hoc Netw.*, vol. 10, no. 7, pp. 1497–1516, 2012. [Online]. Available: https://doi.org/10.1016/j.adhoc.2012.02.016

[33] R. K. Sundararajan and U. Arumugam, "Intrusion detection algorithm for mitigating sinkhole attack on LEACH protocol in wireless sensor networks," *J. Sensors*, vol. 2015, Feb. 2015, Art. no. 203814.

[34] R. S. Hassoubah, S. M. Solaiman, and M. A. Abdullah, "Intrusion detection of hello flood attack in WSNs using location verification scheme," *Int. J. Comput. Commun. Eng.*, vol. 4, no. 3, pp. 156–165, 2015. [Online]. Available: https://doi.org/10.17706/ijcce.2015.4.3.156-165

[35] N. A. Alrajeh, S. Khan, and B. Shams, "Intrusion detection systems in wireless sensor networks: A review," *J. Distrib. Sensor Netw.*, vol. 9, no. 5, 2013, Art. no. 167575. [Online]. Available: http://www.hindawi.com/journals/ijdsn/2013/167575/abs/

[36] S. Raza, H. Shafagh, K. Hewage, R. Hummen, and T. Voigt, "Lithe: Lightweight secure CoAP for the Internet of Things," *IEEE Sensors J.*, vol. 13, no. 10, pp. 3711–3720, Oct. 2013. [Online]. Available: https://doi.org/10.1109/JSEN.2013.2277656

[37] A. Abduvaliyev, A. S. K. Pathan, J. Zhou, R. Roman, and W. C. Wong, "On the vital areas of intrusion detection systems in wireless sensor networks," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 3, pp. 1223–1237, 3rd Quart., 2013. [Online]. Available: https://doi.org/10.1109/SURV.2012.121912.00006
[38] S. Suthaharan and T. Panchagnula, "Relevance feature selection with data cleaning for intrusion detection system," in *Proc. IEEE Southeastcon*, Mar. 2012, pp. 1–6. [Online]. Available: https://doi.org/10.1109/SECon.2012.6196965
[39] S. Gunduz, B. Arslan, and M. Demirci, "A review of machine learning solutions to denial-of-services attacks in wireless sensor networks," in *Proc. IEEE 14th Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2015, pp. 150–155. [Online]. Available: https://doi.org/10.1109/ICMLA.2015.202
[40] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Comput. Netw.*, vol. 54, no. 15, pp. 2787–2805, Jun. 2010.
[41] W. Li, W. Meng, X. Luo, and L. F. Kwok, "*MVPSys*: Toward practical multi-view based false alarm reduction system in network intrusion detection," *Comput. Secur.*, vol. 60, pp. 177–192, Jul. 2016. [Online]. Available: https://doi.org/10.1016/j.cose.2016.04.007
[42] I. Almomani, B. Al-Kasasbeh, and M. Al-Akhras, "WSN-DS: A dataset for intrusion detection systems in wireless sensor networks," *J. Sensors*, vol. 2016, Aug. 2016, Art. no. 4731953. [Online]. Available: https://doi.org/10.1155/2016/4771953
[43] H. H. Bosman, G. Iacca, A. Tejada, H. J. Wörtche, and A. Liotta, "Spatial anomaly detection in sensor networks using neighborhood information," *Inf. Fusion*, vol. 33, pp. 41–56, Jan. 2017. [Online]. Available: https://doi.org/10.1016/j.inffus.2016.04.007
[44] T. Winter *et al.*, *RPL: IPv6 Routing Protocol for Low Power and Lossy Networks*, document Work In Progress, Jul. 2011, pp. 1–164. [Online]. Available: http://tools.Ietf.Org/html/draft-Ietf-Roll-Rpl-19 and https://doi.org/10.2313/NET-2011-07-1
[45] A. A. Gendreau and M. Moorman, "Survey of intrusion detection systems towards an end to end secure Internet of Things," in *Proc. IEEE 4th Int. Conf. Future Internet Things Cloud (FiCloud)*, Aug. 2016, pp. 84–90. [Online]. Available: https://doi.org/10.1109/FiCloud.2016.20
[46] F. Afifi, N. B. Anuar, S. Shamshirband, and K.-K. R. Choo, "DyHAP: Dynamic hybrid ANFIS-PSO approach for predicting mobile malware," *PLoS ONE*, vol. 11, no. 9, pp. e0162627-1–e0162627-21, 2016. [Online]. Available: https://doi.org/10.1371/journal.pone.0162627
[47] G. Mulligan, "The 6LoWPAN architecture," in *Proc. 4th Workshop Embedded Netw. Sensors (EmNets)*, 2007, pp. 72–78. [Online]. Available: https://doi.org/10.1145/1278972.1278992
[48] A. K. Sen and A. Bagchi, "Fast recursive formulations for best-first search that allow controlled use of memory," in *Proc. Int. Joint Conf. Artif. Intell.*, vol. 1. 1989, pp. 297–302.
[49] R. Dechter and J. Pearl, "Generalized best-first search strategies and the optimality of A*," *J. ACM*, vol. 32, no. 3, pp. 505–536, 1985. [Online]. Available: https://doi.org/10.1145/3828.3830
[50] G. Bendall and F. Margot, "Greedy-type resistance of combinatorial problems," *Discrete Optim.*, vol. 3, no. 4, pp. 288–298, 2005.
[51] S. Raza, S. Duquennoy, J. Höglund, U. Roedig, and T. Voigt, "Secure communication for the Internet of Things—A comparison of link-layer security and IPsec for 6LoWPAN," *Secur. Commun. Netw.*, vol. 7, no. 12, pp. 2654–2668, 2014, doi: 10.1002/sec.406.

MOHAMAD NAZRIN NAPIAH received the bachelor's degree in computer science (computer system and network) from the University of Malaya, where he is currently pursuing the master's degree in computer science by research with the Department of Computer System and Technology, Faculty Science Computer and Information Technology. His research interest is in the area of Internet of Things, information security, machine learning, anomaly detection, and network communication.

MOHD YAMANI IDNA BIN IDRIS received the Ph.D. degree in electrical engineering and has vast experience in research. He is currently an Associate Professor with the Faculty of Computer Science and Information Technology, University of Malaya. His expertise is in the area of sensor network, security system, and signal/image processing.

ROZIANA RAMLI received the bachelor's and master's degrees in engineering from the University of Malaya, where she is currently pursuing the Ph.D. degree with the Faculty of Computer Science and Information Technology. Her current research interest includes image/signal processing and analysis.

ISMAIL AHMEDY received the Ph.D. degree in computer science. He is currently a Senior Lecturer with the Department of Computer System and Technology, Faculty Science Computer and Information Technology, University of Malaya. His area of expertise are underwater acoustic sensor networks, embedded system, and wireless sensor networks.