

## Intrusion detection method for internet of things based on the spiking neural network and decision tree method

Ahmed R. Zarzoor<sup>1</sup>, Nadia Adnan Shiltagh Al-Jamali<sup>2</sup>, Dina A. Abdul Qader<sup>2</sup>

<sup>1</sup>Directorate of Inspection, Ministry of Health, Baghdad, Iraq

<sup>2</sup>Department of Computer Engineering, University of Baghdad, Baghdad, Iraq

### Article Info

#### Article history:

Received Apr 25, 2022

Revised Oct 16, 2022

Accepted Nov 8, 2022

#### Keywords:

Deep neural network

Internet of things

Intrusion detection system

Spike neural network

### ABSTRACT

The prevalence of using the applications for the internet of things (IoT) in many human life fields such as economy, social life, and healthcare made IoT devices targets for many cyber-attacks. Besides, the resource limitation of IoT devices such as tiny battery power, small storage capacity, and low calculation speed made its security a big challenge for the researchers. Therefore, in this study, a new technique is proposed called intrusion detection system based on spike neural network and decision tree (IDS-SNNDT). In this method, the DT is used to select the optimal samples that will be hired as input to the SNN, while SNN utilized the non-leaky integrate neurons fire (NLIF) model in order to reduce latency and minimize devices' power usage. Also, a rand order code (ROC) technique is used with SNN to detect cyber-attacks. The proposed method is evaluated by comparing its performance with two other methods: IDS-DNN and IDS-SNNTLF by using three performance metrics: detection accuracy, latency, and energy usage. The simulation results have shown that IDS-SNNDT attained low power usage and less latency in comparison with IDS-DNN and IDS-SNNTLF methods. Also, IDS-SNNDT has achieved high detection accuracy for cyber-attacks in contrast with IDS-SNNTLF.

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



### Corresponding Author:

Ahmed R. Zarzoor

Directorate of Inspection, Ministry of Health

Baghdad, Iraq

Email: Ahmed.Arjabi@gmail.com

## 1. INTRODUCTION

Internet of things (IoT) smart devices are interconnected with each other, and to the internet via using protocols. Also, these devices are expanding rapidly and playing a pivotal role in human daily life. They have been used in many applications such as smart city, home, and car applications [1]–[3]. Consequently, there will be a community of interconnected smart things sharing and exchanging data in the world. Cisco company has foreseen that above than 200 billion smart things will be communicated to the internet via 2030 [4]. So, IoT devices are vulnerable to attacks besides their resource limitation making their data security the main challenge for researchers [5]. Moreover, it makes the security methods for key management, cyber-attacks detection, and trust management among the significant defies of the IoT network [6]. For instance, some researchers are handling security problems and defying the IoT network by using intrusion detection systems (IDS) [7]. The traditional IDS works on two levels: host level and network level [8]. The IDS works on the network level and is considered the most suitable secure method for the IoT network [9] due to the limitation of the IoT nodes' resources (such as the low battery power and small storage capacity). Besides, the IoT network needs to be trained in either online traffic (i.e., live traffic) or offline (i.e., suitable dataset) in order to predict cyber-attacks. However, most researchers preferred to use the offline one to train the network because of the high cost of the online one [10].

However, there are techniques for identifying cyber-attack used by IDS which are: the IDS-based signature method, IDS-based anomaly method, and hybrid IDS [11], [12]. The signature technique is more appropriate to detect known attacks by utilizing a known pattern store in the database (i.e., supervised learning). Thus, the signature method is not suitable to detect unknown attacks. Consequently, the IDS-based anomaly method is used to detect unknown attacks (i.e., unsupervised learning). While the hybrid method used both techniques: signature and anomaly to identify the cyber-attacks. Many researchers worked with anomaly methods via utilizing machine learning (ML) algorithms [13]–[15]. One of the common learning algorithms that are used by IDS based anomaly method is the deep learning algorithm. The DL scheme consists of an input layer, more than one hidden layer, and an output layer. So, the significant features are extracted from the input data via passing the data in multi-hidden layers, and learning is achieved by updating the weight in order to classify output data [16]–[18]. The main disadvantage of the DL is the overfitting of training besides the high exhaustion of network resources [19]. Therefore, the third generation of neural network (NN) called spike neural network (SNN) is used in [20]–[22] to enhance power usage and reduce attack detection time. Nevertheless, the main disadvantage of the SNN is that it is hard to train in comparison with the NN [23].

The SNN is created from a neural network that is spotted in biology, where the biological neurons depend on tentative dimensions. Also, the biological “synaptic” neurons are able to take an input signal and make an output signal, in any case of the action for the remainder of the neurons. In another word, they have internal dynamics which reason biological neurons modify through time. So, with time bypassing the neuron resort to emptying and reducing their membrane possibility. Thence, scattered input spike shall not reason a biological neuron to spike or fire [23], [24]. The biological neuron connected with each other by synaptic parts with weights. So, SNN learning is achieved by modifying the synaptic weights by utilizing either an unsupervised or supervised method [25]. The most common model to train SNN is called synaptic time-dependent plasticity (STDP) unsupervised approach [26]. The STDP is utilized along with the side restrained fit spiking threshold to learn exemplification for input spike paradigms which are appropriate for classification [27]. The spikes are encoded by converting the input wave signal into a sequence of spikes “spiketrains” in a process called “encoding”. There are two types of encoding: rate code and temporal code. In the rate code, the firing rate is counted and stored in a counter, while in the temporal code the spike information is saved at the timing of a fire. The encoded spike in STDP is trained by using leaky integrate neurons fire (LIF) as a model for representing the membrane possibility. The main problem LIF is hard to train therefore in this study the non-LIF (NLIF) is utilized due to its simplicity to train and gives high performance in comparison with LIF [20].

Therefore, the main contribution of this study is to propose IDS based on the SNN algorithm with the decision tree algorithm as a new method called IDS-SNNDT to detect cyber-attacks in the IoT, where DT is utilized to select the optimal samples that attain input value to SNN, while the SNN is trained via using the NLIF model on the offline dataset (IoT Botnet 2020) and uses rank code order (ROC) method to detect cyber-attack. The rest of this paper organizes: section 2 explores the related studies, section 3 describes the IDS-SNNDT method, and section 4 discusses the implementation of the proposed method and results. The final section includes the study conclusion.

## 2. RELATED WORKS

The SNN has been utilized by IDS to detect attacks on IoT devices by researchers due to its usage of less energy and achieving minimum latency in comparison with DNN. For instance, Johnson *et al.* [28] used SNN and “glial cell” to detect the Trojan attack via using agreeable firing or “spiking rate” on IoT hardware devices. So, when the spiking rate value is not in the acceptable range that means the Trojan detection of the IoT device otherwise, no attack is identified in the IoT device. Maciąg *et al.* [29] used unsupervised anomaly identification in an IoT data stream from online Yahoo datasets called OeSNN. The core idea of OeSNN is about utilizing an input encoding layer that operated on single time series via using gaussian receptive fields (GRF) to simplify the SNN train, so as to identify abnormal data stream modification. In [30], a semi-supervised abnormal detection technique is created according to the evolving SNN (eSNN) called “Gryphon”. In order to detect manifold behaviors and abnormalities related to cyber-attacks that are recognized as advanced persistent threats (APT). In eSNN, one class is utilized to categorize true valued datasets, where each data pattern is a series of spikes via using rank order population encoding (ROPE). So, in this technique they used eSNN to make a decision rule, that correctly specifies the label (class) to new unlabeled data.

In [31], a supervised method is proposed for near-sensor abnormality detection via using a long-shortened time period long short-term memory spiking neural networks (LSNN), approach. In LSNN, two classes of signals are classified: healthy and sanitary by using the backpropagation through time (BPTT) technique. Xing *et al.* [32] proposed a real time eSNN method bounded by Boltzmann machine technique to detect abnormal modification in data streams. The main issue is about using a Boltzmann machine method to increase the categorize accuracy and at the same time reduce the computational resources demands. Jaoudi *et al.* [33] utilized SNNs to detect cyber-attack in vehicles based on the support vector machine (SVM) method.

They convert autoencoder to SNN by applying an adjustment process which obtained the weight and biases. Also, SVM method is used in the training process to trace the vector distance between input and output patterns instead of using labels, so as to learning the data packets kinds. Thus, the threshold is computed by taking the mean loss or error for all training patterns. Consequently, if the reform loss for the pattern is less than calculate threshold values then identifies the pattern as a normal, otherwise identifies it as an anomaly.

Yusob *et al.* [34] proposed a technique to detect anomaly data samples based on the SNN. The technique consisted of three phases: the first phase is used to initialize the weight values utilizing the ROPE approach, the second phase is used to represent the real input data to spike values by utilizing the GRF approach, and the last phase is used to detect the anomaly data pattern. The anomaly data detection process occurs only when the neuron in SNN is spiked. Sahu *et al.* [35] utilized SNN to identify anomaly movement for automatic EEG movement during schizophrenia. They used two techniques: temporal contrast and Poisson probability to find the probability of abnormality emptying of each channel. Also, in our study, the IDS-SNNDT method utilized the Poisson encoder but with temporal-based ROC so as to detect cyber-attacks in the offline dataset IoT Botnet 2020.

### 3. STUDY METHOD

The IDS-SNNDT method is based on the SNN network to detect cyber-attacks in IoT devices based on the Poisson encoding and temporal coding ROC techniques. The encoding process in SNN is the process of transforming wave signal to spikes values so as to be utilized as an input value of a node. For SNN, there are two types of encoding approaches: rate code and temporal code [36]. The rate codes firm the information in the coverage rate of spike obstetrics of one or set of nodes in a way that drives to a value that characterizes the activity of the nodes. In the temporal coding method, accurate timing of spikes and among action potentials is used to encode information. This involves the full timing details in relevance to a proportional timing of spikes released via different nodes or just the order that a group of nodes produces specific spikes. In IDS-SNNDT a method of temporal code called ROC is used to detect attacks in offline datasets. The ROC is a method that is established according to the firing order of a group of nodes in relation to the universal reference (i.e., considering the accuracy timing of the spikes) [37], while the Poisson encoding [38] process, a value of wave signal that is taken as an input value of a node, is normalized among the high and low value. The normalized value represents a probability (P) over a time window, where the lower timestamp TS the resulting sequence of spike “spike train” of the encoded wave signal at each TS has a likelihood P, which contains a spike. So, when likelihood P is high then it means more fires “spiketrain” will have. Thus, the information will be encoded more precisely. Figure 1 demonstrates the IDS-SNNDT technique.

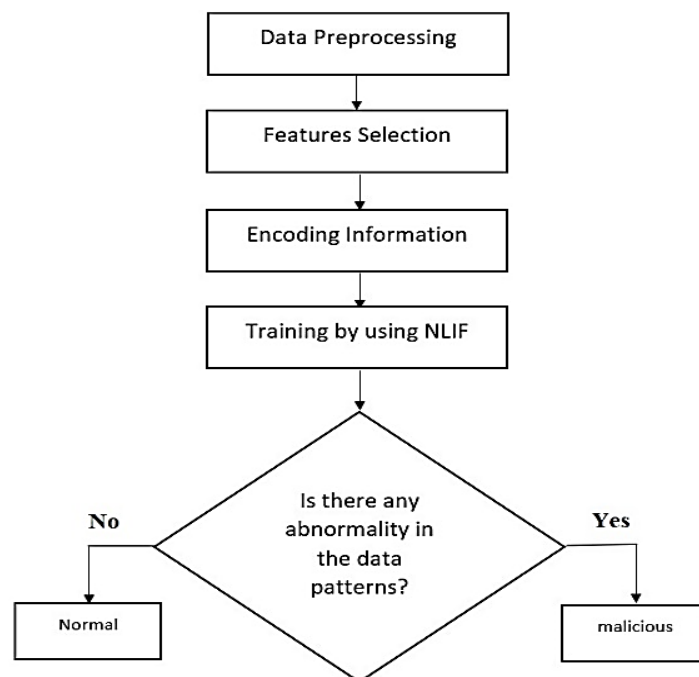


Figure 1. Illustrate IDS-SNNDT method

In the data preprocessing, the IoT Botnet dataset 2020 [39] is used in this study for the training and testing of the IDS-SNN method, where 70% of the dataset is utilized for training and 30% of the dataset is used for testing. The dataset includes more than 72 million records that contain cyber-attacks such as disk operating systems (DoS), distributed denial-of-service (DDoS), and service scan attacks. In this step, the clean data process is performed by removing redundant data and ignoring empty space. Besides, converts data types (float) to the value in the range [0,1] so as to avoid errors. The data in the dataset is scaled to the value between [0,1] by using the “min-max normalization” function (1) [40]. In order to ensure that the SNN training process is not biased to a specific class and guarantees uniformities of learning.

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (1)$$

In the second step, the entropy and information gain are used with a decision tree (DT) [41], [42], where the algorithm uses select specific features from the IoT Botnet features. The DT is based on a tree structure, in which the whole dataset is divided into two subsets, and a subset is divided into two subsets until reaching the final data. The process of dividing the DT is performed by using the entropy method, which is used to measure uncertainty in a dataset of observations. The entropy and information gain (IG) are calculated by utilizing (2) and (3) [43]. However, the total number of features in the IoT Botnet dataset 2020 dataset is 49 with 2 labels (1, 0) that contain 1, 940, 389 records. The final features count according to the entropy and GI with DT are 19 feature selections, see Table 1.

$$Entropy(S) = \sum_{i=1}^c -P_i \log_2 P_i \quad (2)$$

$$IG(Y, X) = E(Y) - E(Y|X) \quad (3)$$

Table 1. IoT Botnet dataset 2020 dataset final features selection

No	Feature	Description
1	pkSeqID	Row Identifier
2	Proto	Textual representation of transaction protocols presents in network flow
3	Saddr	Source IP address
4	Sport	Source port number
5	Daddr	Destination IP address
6	Dport	Destination port number
7	state_number	Numerical representation of feature state
8	Seq	Argus sequence number
9	Mean	Average duration of aggregated records
10	Stddev	Standard deviation of aggregated records
11	Min	Minimum duration of aggregated records
12	Max	Maximum duration of aggregated records
13	N_IN_Conn_P_DstIP	Number of inbound connections per destination IP
14	N_IN_Conn_P_SrcIP	Number of inbound connections per source IP
15	Srate	Source-to-destination packets per second
16	Drate	Destination-to-source packets per second
17	Attack	Class label: 0 for Normal traffic, 1 for Attack Traffic
18	Category	Traffic category
19	Subcategory	Traffic subcategory

In the next step, the NLIF model is used to train SNN. The model represents by using (4), where  $I(t)$  is the input current,  $V$  represents the membrane voltage of neuron  $j$  that asses in time during energizing with an  $I(t)$ , where  $W_{ji}$  is the weight of the synaptic linkage between input node  $i$  and output node  $j$ ,  $t_i$  is the spiking time of  $i$ , while  $g(t)$  is the ‘spike’ or high waveform in this study the  $g(t)$  is assumed equal to zero for  $t < 0$  or  $t < T$  (timestep). So, when  $I(t)$  is applied the  $V$  maximizes with time till it reaches a steady threshold voltage ( $V_{th}$ ). At this point, a spike occurs and  $V$  resets to its restarting potential point, after that the NLIF persists to run. Also, a refractory period ( $trp$ ) is utilized to the boundary spiking frequency of a node by stopping it from spiking over that period. For input, steady input  $I(t)=I$  is the threshold voltage. The spiking frequency for constant  $I(t)$  is calculated using (5). To illustrate how nodes “neurons” operates in the NLIF model. Figure 2(a) demonstrates input for four input nodes at the spike time ( $t_1$ ,  $t_2$ ,  $t_3$ , and  $t_4$ ). Figure 2(b) shows synaptic current for the four nodes that are represented by  $g(t-t_i)$  hops on time  $t_i$ , while Figure 2(c) demonstrates how  $V_j(t)$  increases the firing threshold. Finally, Figure 2(d) shows how the output node  $j$  sends a spike when the  $V_j$  threshold is passed. So, the node emits a spike early than in the LIF model since it waits after  $t_4$  to increase the  $V_j(t)$ . Therefore, the NLIF reduces delay and consumes less energy in comparison with the LIF model.

$$I(t) = C \frac{dV_j(t)}{dt} = \sum_i W_{ji} g(t - t_i) \tag{4}$$

$$S(I) = \frac{I}{cv_{th} + trp I} \tag{5}$$

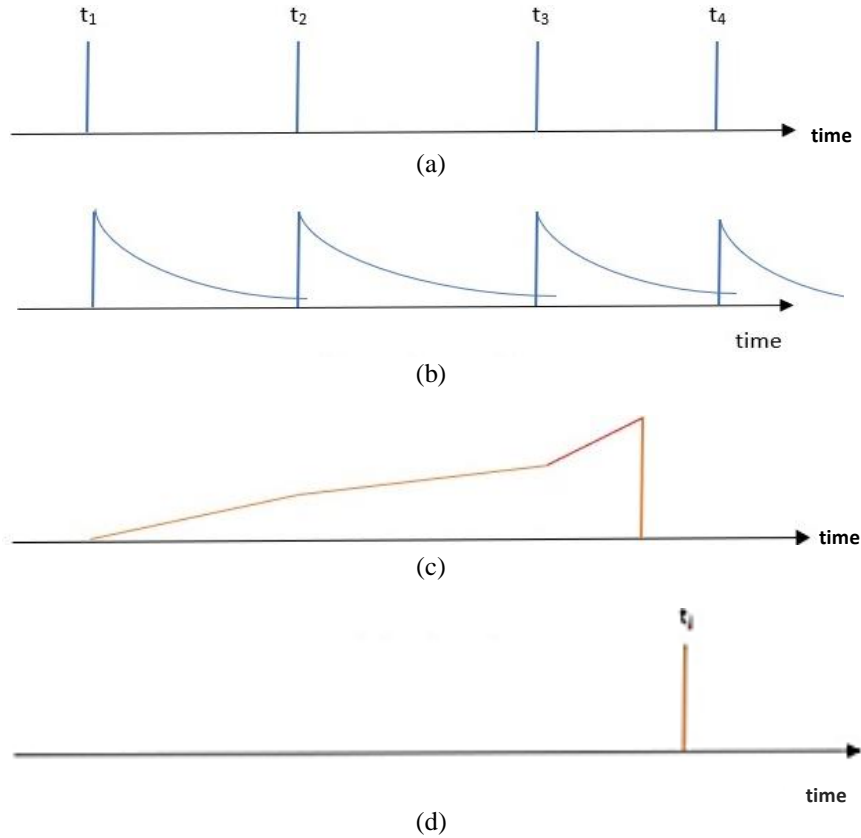


Figure 2. NLIF model: (a) four input time spikes, (b) the synaptic current of the four spikes time, (c) how membrane voltage of node  $j$  ( $V_j(t)$ ) increases the firing threshold, and (d) how the output neuron  $j$  sends a spike when the  $V_j$  threshold is passed

In this study, the SNN consisted of an input layer, two hidden layers, and an output layer as in Figure 3. The GRF method is used to encode information into firing times for the input layer by utilizing (6), where each input information is ranged between (minimum value  $V_{min}$  and maximum value  $V_{max}$ ) with  $\sigma$  is centered by using (7). The spike timing ranges from 0 to T. The T value is calculated by using (8). The  $\sigma$  is specified by the crossing points of the V with identical Gaussian summits: the  $i^{th}$  input receives a spike at  $T - a_i(V)$ . So, if  $a_i(V) > 0.01$  and no spikes then the nearest value of  $v$  to the  $\sigma$  will be taken, as shown in Figure 3, where  $V=0.5$ . Also, the two hidden layers are used to update leaning weight via using a backpropagation algorithm to alleviate error and computed by using (9), where  $W_i$  is the new weight and (b) is the learning rate that represents the minimum value of the error function. Besides, the ROC algorithm is applied on the output layer to get the output value, the order is calculated by utilizing (10), ne is the elected output node,  $n_j$  is the input node, the mod is the modulation factor that gives value in the range (0,1) and order( $n_e$ ) is  $n_j$ 's spiking order value, which established as results of the V encoding. To demonstrate, let  $V=0.5$ ,  $W_0, n_e=0.5$ , and order  $n_0=4$ . Thus, the predicted value (PV)  $0.52=0.25$  and according to the PV, the cyber-attack will be detected where the PV is in the range (0,1). Also, when all PV values are less than 0.5 the output node will not detect any type of attack. Otherwise, the highest PV will be selected to identify the attack type, as shown in Figure 4. For instance, in Figure 5, the DoS attack is identified, it has maximum PV in comparison with PV of other attacks (DDoS, scan OS, scan services, theft data refiltration (TDF) any theft keylogging (TK)).

$$a_i(V) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{\sigma^2}\right) \tag{6}$$

$$\mu_i = V_{min} + (V_{max} - V_{min}) \cdot \frac{i}{n-1} \quad \text{for } i = 0 \text{ to } n - 1 \tag{7}$$

$$T = \max(a_i(V)) \tag{8}$$

$$W_i = W_i - b \left( \frac{\partial Error}{\partial W_i} \right) \tag{9}$$

$$PV = \sum_{j=0}^{Window\ time\ size-1} Wnj, ne\ mod^{order(ne)} \tag{10}$$

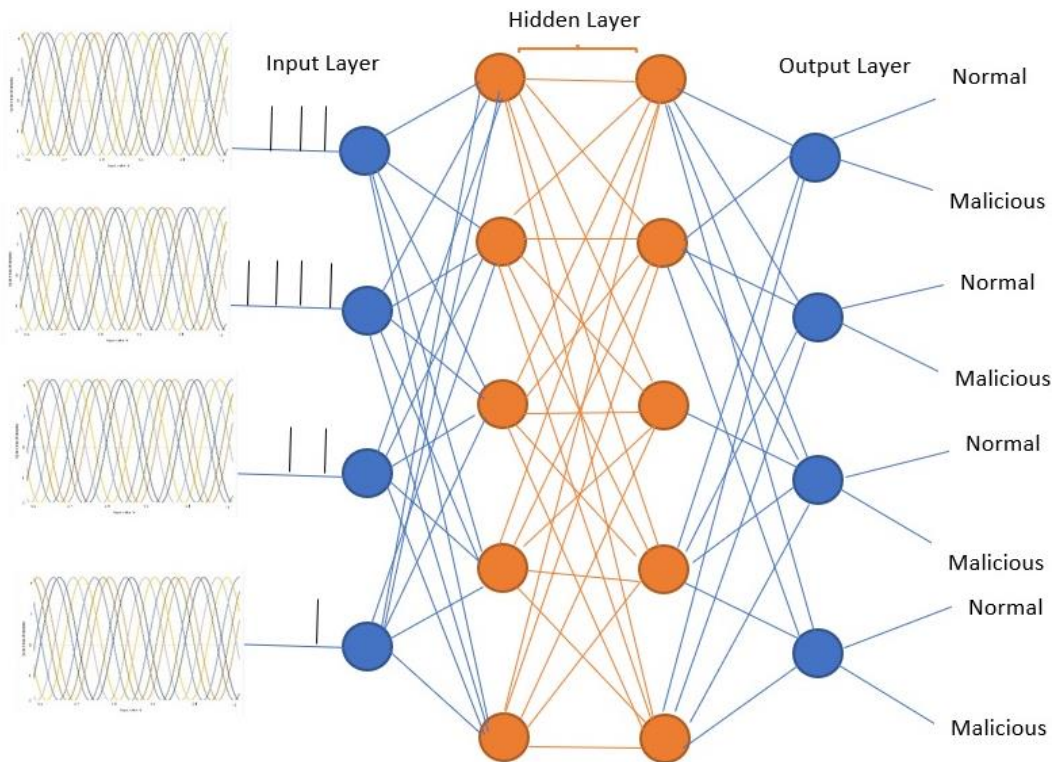


Figure 3. IDS-SNNDT structure

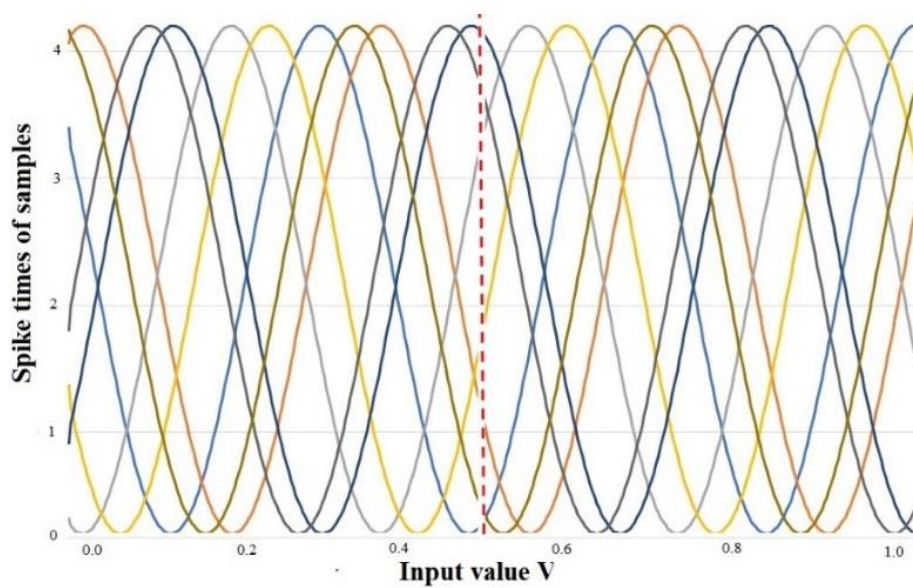


Figure 4. GRF code method

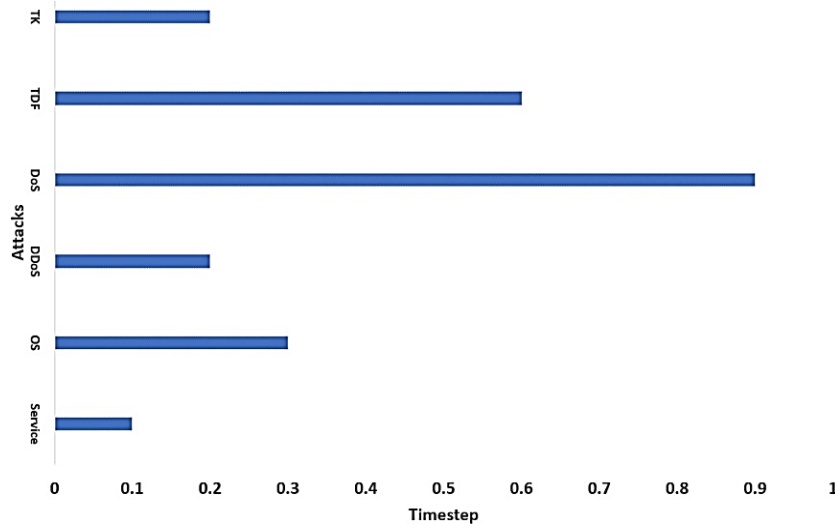


Figure 5. The DoS is identified by utilizing ROC method

#### 4. RESULTS AND DISCUSSION

The study method is implemented on the personal laptop type Lenovo, having a 2.6 GHz Intel Core i5 8<sup>th</sup> generation processor, 4 GB RAM, and Windows 10 operating system. Three scenarios were performed to evaluate the performance of the study method: one for an IDS-SNNDTLF method using the LIF model, the second one for the IDS-SNNDT method using the NLIF model, and the third scenario for the IDS-DNN. The three scenarios were implemented via using python language libraries: *snmTroch* to implement (IDS-SNNDT and IDS-SNNTLF) method and TensorFlow and panda libraries to implement IDS-DNN. For IDS-DNN, the numbers of nodes used are 100 for the input layer and 40 for the hidden layer, as shown in Table 2. For the IDS-SNNTLF and IDS-DNNDT, the ( $V_{th}=65$  mV) for the input layer node, while ( $V_{th}=65$  mV) for the hidden layer node and output layer node and learning rate 0.001 and the max depth of the decision tree is 3, as shown in Table 3. The performance of three scenarios has been evaluated by utilizing three metrics: accuracy of detection (AD), latency, and energy usage. The three metrics have been calculated using (11) to (15), respectively [44], where, in (13) to (15), the  $E_{energy\_Tx}$  represents the amount of power usage required to transmit data (k) with distance (d) from one node to another one,  $E_{energy\_Rx}$  represents the amount of power usage which required to receive data (k) with distance (d) from other nodes..

Table 2. Parameters details for IDS-DNN

Parameter	Value
Input neuron	100
Hidden neuron	20
Activation function	Rectified linear unit (ReLU)
Epochs	100/10
Batch size	64
Optimizer	Adam
Dropout rate	0.9

Table 3. Parameters details for IDS-SNNDT

Parameter	Value
$max\_depth$ for DT	3
learning rate	0.001
batch size	64
Threshold voltage $V_{th}$ of input layer node	15 mV
Threshold voltage $V_{th}$ of hidden/output layer node	65 mV
Membrane resistance (all nodes)	1 M $\Omega$
Membrane time constant (all nodes)	20 ms

$$AD = \frac{True\ Negative + True\ Positive}{True\ Negative + True\ Positive + False\ Negative + False\ Positive} \quad (11)$$

$$Latency = \frac{\sum \text{prior of time that need to deliver data packet to the target node}}{\text{number of received data packet at target node}} \quad (12)$$

$$Node\ Energy\ usage = \text{inital energy} - |E_{energy_{Tx}} + E_{energy_{Rx}}| \quad (13)$$

$$E_{energy_{Tx}}(d, k) = \begin{cases} kE_{elec} + k\epsilon_{amp} d^2, & d < d_0 \\ kE_{elec} + k\epsilon_{amp} d^4, & d \geq d_0 \end{cases} \quad (14)$$

$$E_{energy_{Rx}}(k) = kE_{elec} + kE_{pa} \quad (15)$$

The three scenarios are applied to the dataset IoT Botnet 2020 (where 70% of the dataset is utilized for training and 30% of the dataset is used for testing). The dataset includes more than 72,000,000 records that contain attacks such as DoS, DDoS, Scan OS, TDF, and TK attacks. The results have shown for the accuracy of detection metric, the IDS-DNN achieves high accuracy of detection for training and testing of DoS (95.59%, 95.59%), DDoS ( 92.18 % , 92.11%), TDF (94.26%, 94.15%), TK (99.44%, 99.55%), OS (99.90%, 99.90%) in comparison with IDS-SNNDT DoS (94.00%, 94.50%), DDoS ( 91.99 % , 90.04%), TDF (93.22%, 98.03%), TK (99.66%, 99.80%), OS (99.88%, 99.70%) and IDS-SNNTLF, DoS (90.00%, 95.59%), DDoS (90.90 % , 89.90%), TDF (88.22%, 90.10%), TK (94.66%, 92.33%), OS (95.88%, 96.20%), as shown in Table 4 and Figure 6. Nevertheless, the IDS-SNNDT method gives high accuracy of detection in contrast with IDS-SNNTLF. On the contrary, for latency metric, the IDS-SNNDT method achieves less delay in comparison with IDS-DNN and IDS-SNNTLF, see Figure 7. Also, for energy usage metric the IDS-SNNDT method consumes less power in contrast to the IDS-DNN and IDS-SNNTLF method, see Figure 8.

Table 4. Accuracy of detection details

Method	Accuracy of Detection									
	DoS		DDoS		TDF		TK		OS	
	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
IDS-DNN	95.59	95.59	92.18	92.11	94.26	94.15	99.55	99.66	99.90	99.98
IDS-SNNDT	94	94.5	91.99	90.04	99.66	98.03	99.88	99.80	99.88	99.70
IDS-SNNTLF	90	92.54	90.90	89.90	88.22	90.10	94.66	92.34	95.88	96.20

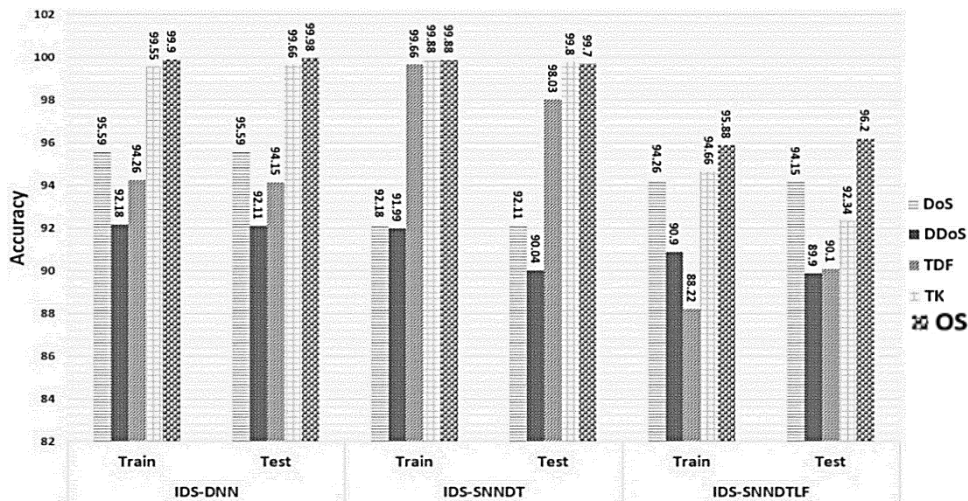


Figure 6. Illustrates accuracy of detection for IDS-DNN, IDS-SNNDT and IDS-SNNTLF method

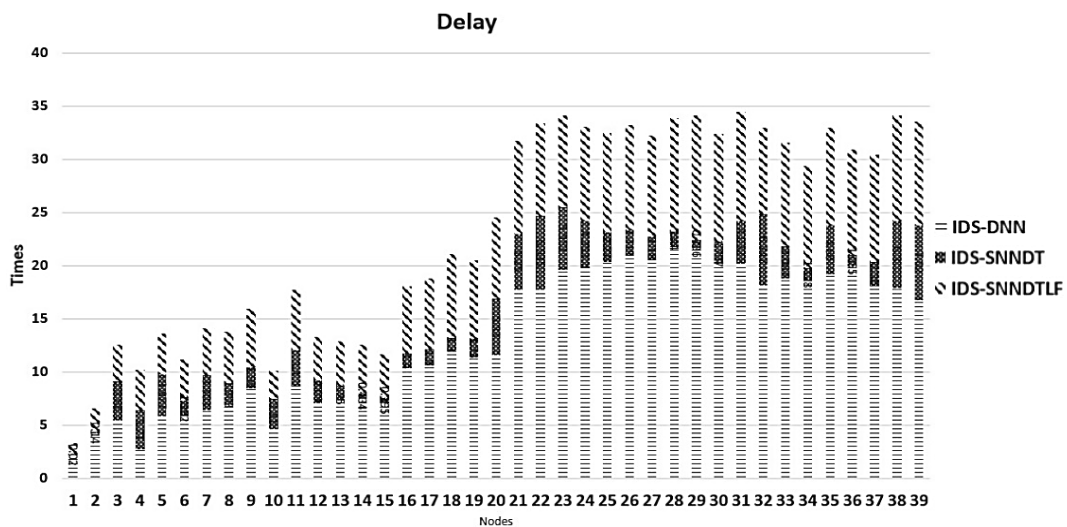


Figure 7. Illustrates latency for IDS-DNN, IDS-SNNDT and IDS-SNNTLF method



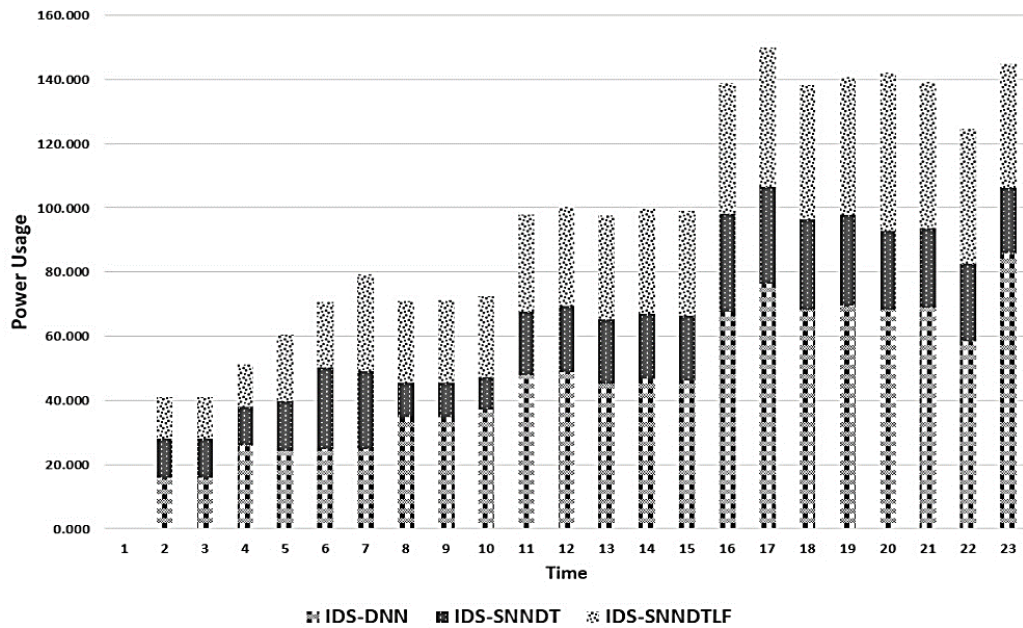


Figure 8. Power usage for IDS-DNN, IDS-SNNNT and IDS-SNNNTLF method

## 5. CONCLUSION

The IDS-SNNNT method is proposed in this study to improve the performance of IDS in a way that meets the IoT network resources restriction. The method has been established based on the DT and SNN. The DT is used to select the features and SNN is utilized to detect cyber-attacks. The SNN is established by the NLIF model in order to minimize the delay and reduce devices' energy consumption. The SNN consisted of three layers: an input layer, two hidden layers, and an output layer. The GRF algorithm has been used to encode selected features to hire them as the input values for the input layer, while the ROC method has been used to detect cyber-attack based on the PV.

However, the study method has been implemented by using Python language and applied to the IoT Botnet 2020 dataset. Also, the method is evaluated with two methods via utilizing three metrics: accuracy of detection and latency and energy usage on three scenarios: IDS-DNN, IDS-SNNNT, and IDS-SNNNTLF. The implementation results have shown that IDS-SNNNT gives low power usage and less latency in comparison with IDS-SNNNTLF and IDS-DNN. Besides, its success in achieving higher accuracy of cyber-attack detection in comparison with IDS-SNNNTLF.

## ACKNOWLEDGEMENTS

The authors would like to appreciate all the excellent suggestions of anonymous reviewers to enhance the quality of this paper. Also, the authors received no financial support for the research, authorship, and/or publication of this article.

## REFERENCES




- [1] S. K. Routray, K. P. Sharmila, E. Akansha, A. D. Ghosh, L. Sharma, and M. Pappa, "Narrowband and IoT (NB-IoT) for smart cities," in *2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)*, Feb. 2021, pp. 393–398. doi: 10.1109/ICICV50876.2021.9388513.
- [2] F. Alshaym, T. Al-Hadhrami, F. Saeed, and K. Awson-David, "Toward home automation: an IoT based home automation system control and security," in *2021 International Congress of Advanced Technology and Engineering (ICOTEN)*, Jul. 2021, pp. 1–11. doi: 10.1109/ICOTEN52080.2021.9493464.
- [3] M. Patil, V. Chakole, and K. Chetepawad, "IoT based economic smart vehicle parking system," in *2020 3rd International Conference on Intelligent Sustainable Systems (ICISS)*, Dec. 2020, pp. 1337–1340. doi: 10.1109/ICISS49785.2020.9315919.
- [4] Cisco, "Cisco annual internet report (2018-2023) white paper," Cisco 2020. Accessed Dec 1, 2021. [Online]. Available: <https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html>
- [5] S. Hameed, F. I. Khan, and B. Hameed, "Understanding security requirements and challenges in internet of things (IoT): a review," *Journal of Computer Networks and Communications*, pp. 1–14, Jan. 2019, doi: 10.1155/2019/9629381.
- [6] R. F. Ali, A. Muneer, P. D. D. Dominic, S. M. Taib, and E. A. A. Ghaleb, "Internet of things (IoT) security challenges and solutions: a systematic literature review," in *Communications in Computer and Information Science*, Springer Singapore, 2021, pp. 128–154. doi: 10.1007/978-981-16-8059-5\_9.

- [7] F. Hussain *et al.*, “A framework for malicious traffic detection in iot healthcare environment,” *Sensors*, vol. 21, no. 9, Apr. 2021, doi: 10.3390/s21093025.
- [8] S. Carta, A. S. Podda, D. R. Recupero, and R. Saia, “A local feature engineering strategy to improve network anomaly detection,” *Future Internet*, vol. 12, no. 10, Oct. 2020, doi: 10.3390/fi12100177.
- [9] S. Ghazanfar, F. Hussain, A. U. Rehman, U. U. Fayyaz, F. Shahzad, and G. A. Shah, “IoT-flock: an open-source framework for iot traffic generation,” in *2020 International Conference on Emerging Trends in Smart Technologies (ICETST)*, Mar. 2020, pp. 1–6. doi: 10.1109/ICETST49965.2020.9080732.
- [10] N. Widiyasono, I. A. Dwi Giriantari, M. Sudarma, and L. Linawati, “Detection of Mirai malware attacks in IoT environments using random forest algorithms,” *TEM Journal*, pp. 1209–1219, Aug. 2021, doi: 10.18421/TEM103-27.
- [11] X.-H. Nguyen, X.-D. Nguyen, H.-H. Huynh, and K.-H. Le, “Realguard: a lightweight network intrusion detection system for IoT gateways,” *Sensors*, vol. 22, no. 2, Jan. 2022, doi: 10.3390/s22020432.
- [12] M. A. Alsoufi *et al.*, “Anomaly-based intrusion detection systems in IoT using deep learning: a systematic literature review,” *Applied Sciences*, vol. 11, no. 18, Sep. 2021, doi: 10.3390/app11188383.
- [13] E. M. Campos *et al.*, “Evaluating federated learning for intrusion detection in internet of things: review and challenges,” *Computer Networks*, vol. 203, Feb. 2022, doi: 10.1016/j.comnet.2021.108661.
- [14] N. Islam *et al.*, “Towards machine learning based intrusion detection in IoT networks,” *Computers, Materials and Continua*, vol. 69, no. 2, pp. 1801–1821, 2021, doi: 10.32604/cmc.2021.018466.
- [15] N. Awadallah Awad, “Enhancing network intrusion detection model using machine learning algorithms,” *Computers, Materials and Continua*, vol. 67, no. 1, pp. 979–990, 2021, doi: 10.32604/cmc.2021.014307.
- [16] J. Lansky *et al.*, “Deep learning-based intrusion detection systems: a systematic review,” *IEEE Access*, vol. 9, pp. 101574–101599, 2021, doi: 10.1109/ACCESS.2021.3097247.
- [17] Z. K. Maseer, R. Yusof, S. A. Mostafa, N. Bahaman, O. Musa, and B. Ali Saleh Al-rimy, “DeepIoT.IDS: hybrid deep learning for enhancing IoT network intrusion detection,” *Computers, Materials and Continua*, vol. 69, no. 3, pp. 3945–3966, 2021, doi: 10.32604/cmc.2021.016074.
- [18] A. Aldweesh, A. Derhab, and A. Z. Emam, “Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues,” *Knowledge-Based Systems*, vol. 189, Feb. 2020, doi: 10.1016/j.knosys.2019.105124.
- [19] S. Tsimenidis, T. Lagkas, and K. Rantos, “Deep learning in IoT intrusion detection,” *Journal of Network and Systems Management*, vol. 30, no. 1, Jan. 2022, doi: 10.1007/s10922-021-09621-9.
- [20] S. Zhou and X. Li, “Spiking neural networks with single-spike temporal-coded neurons for network intrusion detection,” in *2020 25th International Conference on Pattern Recognition (ICPR)*, Jan. 2021, pp. 8148–8155. doi: 10.1109/ICPR48806.2021.9412580.
- [21] A. D. Majeed and N. A. S. Al-Jamali, “Spike neural network as a controller in SDN network,” *Journal of Engineering*, vol. 27, no. 9, pp. 64–77, Sep. 2021, doi: 10.31026/j.eng.2021.09.06.
- [22] N. A. S. Aljamali, “Convolutional multi-spike neural network as intelligent system prediction for control systems,” *Journal of Engineering*, vol. 26, no. 11, pp. 184–194, Nov. 2020, doi: 10.31026/j.eng.2020.11.12.
- [23] W. He *et al.*, “Comparing SNNs and RNNs on neuromorphic vision datasets: Similarities and differences,” *Neural Networks*, vol. 132, pp. 108–120, Dec. 2020, doi: 10.1016/j.neunet.2020.08.001.
- [24] S. Dora and N. Kasabov, “Spiking neural networks for computational intelligence: an overview,” *Big Data and Cognitive Computing*, vol. 5, no. 4, Nov. 2021, doi: 10.3390/bdcc5040067.
- [25] W. Zhang and P. Li, “Temporal spike sequence learning via backpropagation for deep spiking neural networks,” *Prepr. arXiv2002.10085*, Feb. 2020.
- [26] B. C. Schwab, P. König, and A. K. Engel, “Spike-timing-dependent plasticity can account for connectivity aftereffects of dual-site transcranial alternating current stimulation,” *NeuroImage*, vol. 237, Aug. 2021, doi: 10.1016/j.neuroimage.2021.118179.
- [27] S. G. Hu, G. C. Qiao, T. P. Chen, Q. Yu, Y. Liu, and L. M. Rong, “Quantized STDP-based online-learning spiking neural network,” *Neural Computing and Applications*, vol. 33, no. 19, pp. 12317–12332, Oct. 2021, doi: 10.1007/s00521-021-05832-y.
- [28] A. P. Johnson, H. Al-Aqrabi, and R. Hill, “Bio-inspired approaches to safety and security in iot-enabled cyber-physical systems,” *Sensors*, vol. 20, no. 3, Feb. 2020, doi: 10.3390/s20030844.
- [29] P. S. Maciag, M. Kryszkiewicz, R. Bembenik, J. L. Lobo, and J. Del Ser, “Unsupervised anomaly detection in stream data with online evolving spiking neural networks,” *Neural Networks*, vol. 139, pp. 118–139, Jul. 2021, doi: 10.1016/j.neunet.2021.02.017.
- [30] K. Demertzis, L. Iliadis, and I. Bougoudis, “Gryphon: a semi-supervised anomaly detection system based on one-class evolving spiking neural network,” *Neural Computing and Applications*, vol. 32, no. 9, pp. 4303–4314, May 2020, doi: 10.1007/s00521-019-04363-x.
- [31] F. Barchi *et al.*, “Spiking neural network-based near-sensor computing for damage detection in structural health monitoring,” *Future Internet*, vol. 13, no. 8, Aug. 2021, doi: 10.3390/fi13080219.
- [32] L. Xing, K. Demertzis, and J. Yang, “Identifying data streams anomalies by evolving spiking restricted Boltzmann machines,” *Neural Computing and Applications*, vol. 32, no. 11, pp. 6699–6713, Jun. 2020, doi: 10.1007/s00521-019-04288-5.
- [33] Y. Jaoudi, C. Yakopcic, and T. Taha, “Conversion of an unsupervised anomaly detection system to spiking neural network for car hacking identification,” in *2020 11th International Green and Sustainable Computing Workshops (IGSC)*, Oct. 2020, pp. 1–4. doi: 10.1109/IGSC51522.2020.9291232.
- [34] B. Yusob, Z. Mustaffa, and J. Sulaiman, “Anomaly detection in time series data using spiking neural network,” *Advanced Science Letters*, vol. 24, no. 10, pp. 7572–7576, Oct. 2018, doi: 10.1166/asl.2018.12980.
- [35] R. Sahu, S. R. Dash, L. A. Cacha, R. R. Poznanski, and S. Parida, “Classifier implementation for spontaneous EEG activity during schizophrenic psychosis,” *Computación y Sistemas*, vol. 25, no. 3, Sep. 2021, doi: 10.13053/cys-25-3-3874.
- [36] G. Datta, S. Kundu, and P. A. Beerel, “Training energy-efficient deep spiking neural networks with single-spike hybrid input encoding,” in *2021 International Joint Conference on Neural Networks (IJCNN)*, Jul. 2021, pp. 1–8. doi: 10.1109/IJCNN52387.2021.9534306.
- [37] D. Auge, J. Hille, E. Mueller, and A. Knoll, “A survey of encoding techniques for signal processing in spiking neural networks,” *Neural Processing Letters*, vol. 53, no. 6, pp. 4693–4710, Dec. 2021, doi: 10.1007/s11063-021-10562-2.
- [38] Y. Wang *et al.*, “A low-cost hardware-friendly spiking neural network based on binary MRAM synapses, accelerated using in-memory computing,” *Electronics*, vol. 10, no. 19, Oct. 2021, doi: 10.3390/electronics10192441.
- [39] N. Koroniotis, N. Moustafa, and E. Sitnikova, “A new network forensic framework based on deep learning for Internet of Things networks: A particle deep framework,” *Future Generation Computer Systems*, vol. 110, pp. 91–106, Sep. 2020, doi: 10.1016/j.future.2020.03.042.
- [40] X. Larriva-Novo, V. A. Villagrà, M. Vega-Barbas, D. Rivera, and M. Sanz Rodrigo, “An IoT-focused intrusion detection system approach based on preprocessing characterization for cybersecurity datasets,” *Sensors*, vol. 21, no. 2, Jan. 2021, doi: 10.3390/s21020656.




- [41] X. Zhao and X. Nie, "Splitting choice and computational complexity analysis of decision trees," *Entropy*, vol. 23, no. 10, Sep. 2021, doi: 10.3390/e23101241.
- [42] B. Charbuty and A. Abdulazeez, "Classification based on decision tree algorithm for machine learning," *Journal of Applied Science and Technology Trends*, vol. 2, no. 01, pp. 20–28, 2021.
- [43] N. Paedeheh and K. Ghiasi-Shirazi, "Improving the backpropagation algorithm with consequentialism weight updates over mini-batches," *Neurocomputing*, vol. 461, pp. 86–98, Oct. 2021, doi: 10.1016/j.neucom.2021.07.010.
- [44] A. R. Zarzoor, "Enhancing dynamic source routing (DSR) protocol performance based on link quality metrics," in *2021 International Seminar on Application for Technology of Information and Communication (iSemantic)*, Sep. 2021, pp. 17–21. doi: 10.1109/iSemantic52711.2021.9573233.

## BIOGRAPHIES OF AUTHORS






**Ahmed R. Zarzoor**    received his M.Sc. degree in software engineering from University of Bradford, UK, Bradford in 2006, and a Ph.D. degree in computer science from Post-Graduation Studies Iraqi Commission for Computer and informatics, Baghdad, Iraq. He is currently a Director of Information Technology at the Ministry of Health, Baghdad, Iraq. His main interest includes WSN, IoTs, MANET, computer networks and security, and soft computing. He can be contacted at [Ahmed.Arjabi@gmail.com](mailto:Ahmed.Arjabi@gmail.com).



**Nadia Adnan Shiltagh Al-Jamali**    received a B.Sc. degree in control and systems engineering, an M.Sc. degree in control engineering, and a Ph.D. degree in computer engineering from the University of Technology, Baghdad, Iraq. Her fields of interest are computer control, wireless sensor networks, intelligent systems, neural networks, and robotics. She can be contacted at [nadia.aljamali@coeng.uobaghdad.edu.iq](mailto:nadia.aljamali@coeng.uobaghdad.edu.iq).



**Dina A. Abdul Qader**    received a B.Sc. degree in computer engineering from Baghdad University, Iraq, in 2006 and an M.S. degree in Computer Engineering from the University of Baghdad, Iraq, in 2012. Currently, as an assistant lecturer, Dina is working as a faculty member in the College of Engineering at the University of Baghdad, Iraq. Her research interests include machine learning, neural network, artificial intelligence, convolutional neural network, image processing, regression, classification, robot, control, and FPGA. She can be contacted at [dina\\_aldaloo@coeng.uobaghdad.edu.iq](mailto:dina_aldaloo@coeng.uobaghdad.edu.iq).