IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Mathematical Modelling of Engineering Problems
Vol. 10, No. 2, April, 2023, pp. 639-646
Journal homepage: http://iieta.org/journals/mmep

Traffic Classification of IoT Devices by Utilizing Spike Neural Network Learning Approach

Ahmed R. Zarzoor[1*], Nadia Adnan Shiltagh Al-Jamali[2], Ibtesam R.K. Al-Saedi[3]

[1] Directorate of Inspection, Ministry of Health, Baghdad 10047, Iraq
[2] Department of Computer Engineering, University of Baghdad, Baghdad 10071, Iraq
[3] Communication Engineering Department, University of Technology, Baghdad 10066, Iraq

Corresponding Author Email: ahmed.arjabi@mizan.edu.iq

https://doi.org/10.18280/mmep.100234

Received: 18 December 2022
Accepted: 13 March 2023

Keywords:
Spike Neural Network (SNN), Internet of Things (IoTs), Support Vector Machine (SVM), Deep Neural Network (DNN)

ABSTRACT

Whenever, the Internet of Things (IoT) applications and devices increased, the capability of the its access frequently stressed. That can lead a significant bottleneck problem for network performance in different layers of an end point to end point (P2P) communication route. So, an appropriate characteristic (i.e., classification) of the time changing traffic prediction has been used to solve this issue. Nevertheless, stills remain at great an open defy. Due to of the most of the presenting solutions depend on machine learning (ML) methods, that though give high calculation cost, where they are not taking into account the fine-accurately flow classification of the IoT devices is needed. Therefore, this paper presents a new model based on the Spike Neural Network (SNN) called IoT-Traffic Classification (IoT-TCSNN) to classify IoT devices traffic. The model consists of four phases: data preprocessing, feature extraction, classier and evaluation. The proposed model performance is evaluated according to evaluation metrics: accuracy, precision, recall and F1-score and energy usage in comparison with two models: ML based Support Vector Machine IoT-TCSVM and ML based Deep Neural Network (IoT-TCDNN). The evaluations result has been shown that IoT-TCSNN consumes less energy in contrast to IoT-TCDNN and IoT-TCSVM. Also, it gives high accuracy in comparison with IoT-TCSVM.

1. INTRODUCTION

In the last decade a new technology called an Internet of Things (IoTs) has become a favorable infrastructure to answer to the on-request needs of the users. An IoT creates a network of billions smart devices (such as cameras, smartphones, printers, TVs, …, etc.) that interconnect with each other and exchanged data without human intervention (Machine to Machine) M2M [1]. This can lead to a massive amount of exchanged data, that cause a traffic congestion causes a significant network performance bottleneck in diverse layers of an end point to end point (P2P) communication route, involving of the resource needs for handling IoT data at the border or cloud. Also, it causes delay in case of dangerous emergency scenarios [2, 3]. To solve this problem, many researchers [4-7] presents the Internet Traffic Control (ITC) framework to handle the network performance degradation. The ITC framework classifies network either on the type of application or the type of IoT devices. However, ITC is classified into four approaches: port-based technique, payload-based technique, statistical based technique, trace based (Machine Learning ML) technique. In the port number method, the network traffic is classified based on the port information (such as data segment, transmission unit) to allocated the traffic. The main advantage port approach is that traffic classification process fast and low cost in contrast to other techniques. The disadvantage, its limitation for applications which can run in whatever port number [8].

While, the payload approach based on the isolation of the traffic IP in the application layer, in order to characterize Point to Point (P2P) application traffic. In this technique, payload information (such as number of sessions, the arrival time of the packet, …, etc.) is used to check the packet payload and match it with saves signature in the database. So, when it matches the traffic is classified. The main disadvantage of this technique, it consumes more memory and need more calculation process besides it ignores the encrypted traffic [9]. To overcome the problem of payload approach with encrypted traffic, the statistical based technique has been utilized to classify network traffic based on the statistical features [10] (i.e., characteristics) of the traffic such as traffic influx, packet volume, optimal time and interval of the packet. The gauges of features can characterize different applications among each other as applications with unique type. The main problem of the statistical approach, that it required high resource usage (memory and data processing time). In contrast, ML based approach utilizes features to classify network traffic. The main idea is about using ML (supervised, unsupervised) algorithms to trace the features (such as packet length, source port number, destination port number, inter-arrival time, Media Access Control (MAC) address, …, etc.) from large network data influx and afterwards utilize these features to characteristic and predict the network traffic and IoT devices characteristics [11]. This technique gives high accuracy of traffic characteristics in comparison with port and payload approaches. In contrast, the main deficiency of ML based approach is the high calculation cost due to not consider the fine precision of trace traffic features [12].

However, the increase of applications and devices in the Internet of Things (IoT) often makes us rethink their access capabilities, and the end-to-end network performance is also the focus of attention. So, machine learning method has been widely used to solve the above problems, but machine learning method does not take the flow classification of the IoT into account. So how to use neural network method to investigate stream classification in IoT devices is an urgent problem to be solved and considered. Therefore, contributions of this paper are as follows: (1) A new model based on Spike Neural Network (SNN) is proposed, which uses neural network SNN learning method to classify the traffic of IoT devices according to the peak value called IoT-TCSNN model; (2) The traffic can be classified by using packet capacity characteristics and packet capture time information. At the same time, the rank correlation coefficient (Spearman) of SNN can be used as the classifier of crop networking device traffic to extract the best feature group from the global feature group. The IoT-TCSNN applied-on dataset "IoT Traffic Traces" that's available on repository of University of New Souths Wales so as to train IoT-TCSNN model [13]. The proposed model is divided into four phases: Data preprocessing, the feature selection based on two statistical approaches: Correlation coefficient and Spearman rank correlation, classifier of IoT traffic devices by utilizing SNN learning approach, and evaluation of IoT-TCSNN model via using four metrics (accuracy, precision, recall and F1_score). The rest of this paper is organized as follow: section 2, explores the related works that utilize ML to classify IoT devices traffic, section 3 describes the IoT-TCSNN, section 4 presents the implements and results discussion. Finally, section 5 includes study's conclusion.

2. RELATED WORK

In the last few years, many researchers have been utilized a trace based (ML) approach to classify it traffic. For instance, the authors [14] classified internet traffic by using multilayer deep neural network (DNN) with the cross-entropy approach to classify internet traffic. The DNN is used to extract features from the traffic feature influx. While, the cross-entropy approach is utilized as classifies for internet traffic by obtaining maximum entropy classifier value for the traffic. Researchers [15] analyze the smart home traffic against cyber-attacks by suggesting a tool called a botnet based on auto-encoder neural network (ANL) with "gradient boosting decision tree" GBDT algorithm. They used ANL to select features based on the characteristics of communication behavior among network nodes. The GBDT algorithm is used to train unusual traffic disclosure model to enhance the detection of lopsided botnet data. Also, "Time Convolutional Network" (TCN) tool based on a multi class neural network is used to classify cyberattack on the IoT device traffic [16]. Where, the traffic features flow is extracted based on "Deep Packet Inspection" DPI approach and used them as input to TCN to classify malicious traffic from normal traffic. Researchers [17] used two ML algorithms "adaboost and Xgboost" and DNNs learning method to segregation the heavy IoT traffic into four classes: frequent traffic, incident-based traffic, inquiry based on traffic and malignant traffic. In order to optimize the throughput of the network and minimizes the congestion on the network channels.

The study [18] presents a framework for characteristics and revealing of IoT devices via using Hierarchical DNN (HDNNs). The HDNN is utilized to extract a feature set (such as source port number, source MAC address, total forward packet, …etc.) from IoT traffic flow so as to classify IoT devices from non IoT devices. Also, the study [19] utilized a framework consists of seven supervised ML algorithms (Linear Discriminant Analysis, KNN, Random Forests, Multilayer Perceptron, Ada Boosting, Decision Tree and eXtreme Gradient Boosting) to designate IoT devices from network traffic in smart home according to an application type. While, in the study [20] authors identified IoT device classes according to on traffic flux characteristics (such as source IP address, MAC address, destination IP address, source port number, destination port number). Thus, they identified devices behaviors, by measuring the amount of alteration of the sent and received data rate ("Cu index") through a given interval of time. Thus, when Cu is nigh to zero that means IoT device has less amount of received and sent data beside it has a high level of expecting IoT device behavior in comparison to other devices that have a greater value of Cu. The research [21] presents two phases learning approach to classify IoT devices. In their method, they extracted features by using a correlation coefficient approach to utilize it as input to the DL algorithm so as to categorize the devices into four classes: "Triby Speaker", "Natatmo Welcome", "Next Smoke Alarm", "Belkin wemo Switch".

Research [22] exams information-theoretic borders on the prophesy of the IoT traffic device by using Also, they utilized Analysis of variance (ANOVA) method and "Auto-Correlation Function" (ACF) technique, to select a feature. Thereafter comparing the performance attains for five machine learning approaches: Logistic Regression, 1-Dimensional Convolutional Neural Network (1D CNN), Multi-Layer Perceptron (MLP) and Long Short-Term Memory (LSTM) and LR algorithm to identify local ideal value time window for the predicate on odd IoT traffic device. The study [23] has been presenting a new technique called "Cost-aware IoT devices classification" via utilizing cross entropy (CE) established on random optimization approach. The main idea of CE is about selecting the optimal features group from public features group to minimize the misclassification for IoT device traffic within a specific limit. For SNN based methods, researchers [24] have used feedforward supervised SNN learning method to classify four encrypted internet traffic (File transfer, VoIP, chat and browsing). In the SNN approach the traffic is classified based on the information on packet volume features and packet capture time. Where, in this study an SNN is used as classifier for IoT device traffic within a rank correlation coefficient (Spearman) method to extract the optimal feature group from global feature group, see Table 1.

3. STUDY METHOD

In this study IoT-TCSNN model is used to classify IoT devices traffic by using trace-based on Spike Neural Network SNN learning approach. The model consists of four phases: data preprocessing, feature extraction, classification and model evaluation, see Figure 1.

3.1 Data pre-processing

In the IoT-TCSNN, the [13] dataset is used in this study, it includes 1,229,103 records that represent a real network trace traffic for IoT devices. Where, each record consisted of: MAC

address for the source (SRC) and destination (DST) device, packet interarrival time (PIT) and timestamp. The PIT is the total time that terminates which, among two sequential packet receptions, while the timestamp keeps track of normal distribution with an intermediate rate of 1 (i.e., only one packet receives at every time unit 1), to compute interarrival time. Besides, other packet information such as the port number of the SRC and DST, Window size, protocol type, Time to Live (TTL) information, packet size and packet ID, see Table 2. The number of features is twelve (f_1, f_2, f_3, ..., f_{11}). Since the SNN deals with numeric value only, some features such as IP address (f_2 and f_3) and MAC address (f9 and f10) are converted to numeric value by using extraction accuracy of a Spearman correlation coefficient method (SCC).

Table 1. A summary comparison of existing approaches

Study	Year	ML based technique	Summary
[14]	2022	DNN learning approach with cross entropy approach	-Extract features from the traffic feature influx. -Cross-entropy approach is utilized as classifier of the internet traffic by obtaining maximum entropy classifier value for the traffic
[15]	2021	Auto-encoder neural network (ANL) with "gradient boosting decision tree" GBDT algorithm	-ANL to select features based on the characteristics of communication behavior among network nodes. -The GBDT algorithm is used to train unusual traffic disclosure model to enhance the detection of lopsided botnet data
[16]	2022	multi class neural network	- Features selected based on "Deep Packet Inspection" DPI approach and used them as input to TCN to classify malicious traffic from normal traffic
[17]	2021	(adaboost and Xgboost) and DNNs	-Segregation the heavy IoT traffic into four classes: frequent traffic, incident-based traffic, inquiry based on traffic and malignant traffic.
[18]	2022	Hierarchical DNN (HDNNs)	-Extract feature set (such as source port number, source MAC address, total forward packet, …etc.) from IoT traffic flow so as to classify IoT devices from non IoT devices
[19]	2022	Supervised ML algorithm	- Designation IoT devices from non IoT devices from network traffic in smart home based on an application type utilized seven supervised ML algorithms
[20]	2021	ML algorithm	- Identify IoT device classes according to on traffic flux characteristics
[21]	2022	DL algorithm	-Extracted feature by using a correlation coefficient approach and obtained it as input to the DL algorithm to category the devices to four classes.
[22]	2021	Five ML methods and ACF and ANOVA	- ACF and ANOVA to select a feature and 1D CNN, MLP and Long LSTM and LR algorithm to identify local ideal value time window for the predicate on odd IoT traffic device
[23]	2021	Cross Entropy (CE)	-Selecting optimal features group from public features group to minimize the misclassification of IoT device traffic within a specific limit.
[24]	2022	SNN	-Traffic is classified based on the information on packet volume features and packet capture time.
Propose study		SNN	- SNN is used as classifier for IoT device traffic based on a rank correlation coefficient (Spearman) method to extract the optimal feature group from global feature group.

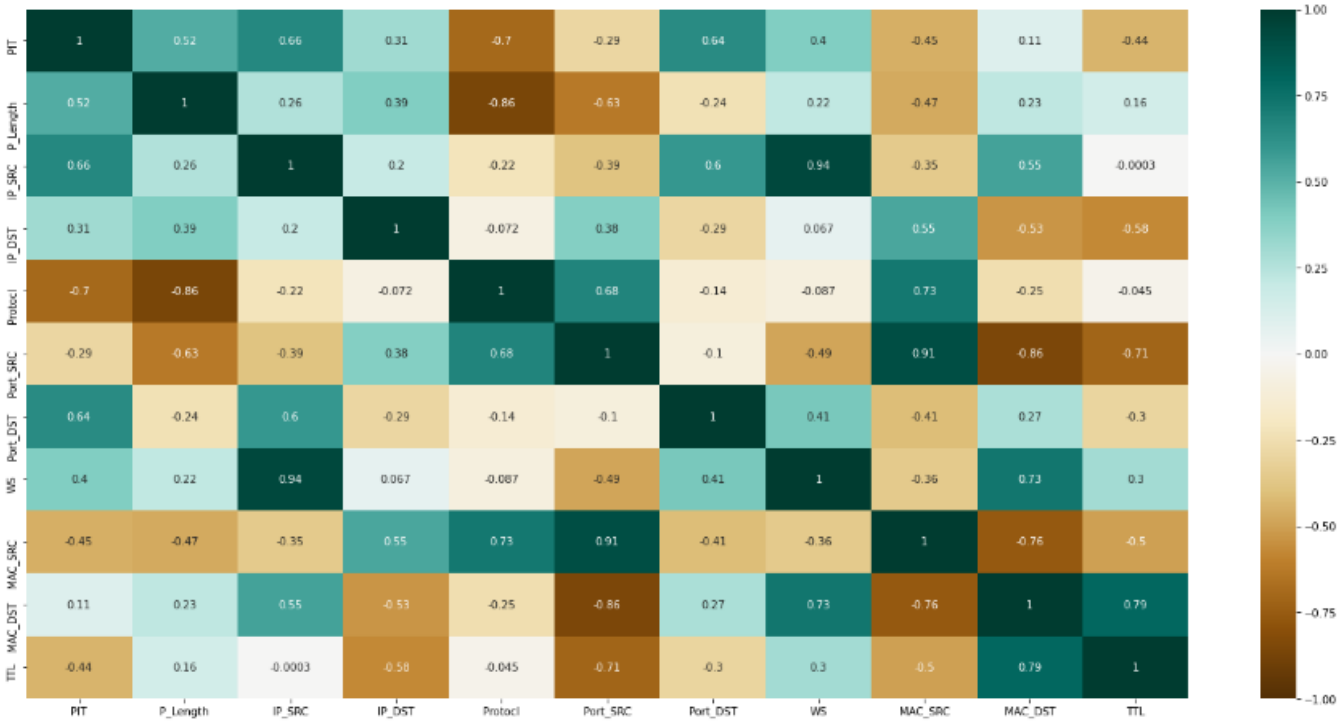

Figure 1. Linear relationship between IoT traffic features

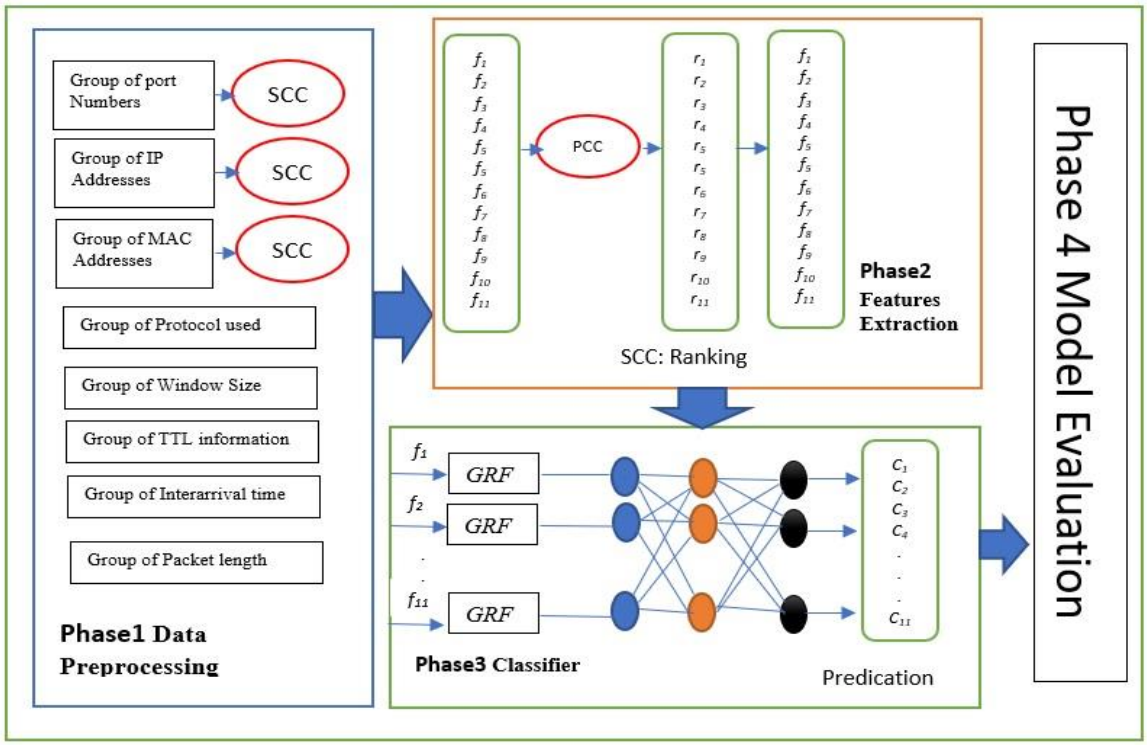

Figure 2. IoT-TCSNN model

Table 2. Description of dataset

Feature	Description	Variable
PIT	Packet intermediate time among two sequential packet receptions	f_1
P_Length	Packet Length	f_2
IP.SRC	IP source address	f_3
IP.DST	IP destination address	f_4
Protocol	Protocol utilized by the flow	f_5
Port.SRC	Source port number	f_6
Port.DST	Destination port number	f_7
WS	Window size	f_8
MAC.SRC	Source MAC Address	f_9
MAC.DST	Destination MAC Address	f_{10}
TTL	Heights number of hop that required for each packet to reach destination	f_{11}

3.2 Feature extraction

In this phase, a Pearson correlation coefficient (PCC) [25] approach is used to measure the linear relationship between the features set ($f_1, f_2, \ldots, f_{11}$). The PCC measures the intensity of difference values between the two features (variables) based on association among them. The coefficient is computed by using Eq. (1). Where, a and b two features, *a*' represent the mean of a, *b*' represent the mean of b, a_i and *bi* are variations of a and b values. To increase feature extraction accuracy a Spearman correlation coefficient (SCC) [26] approach is utilized to compute the range (increase or decrease) of feature in association with other feature (in between two ordinary variables (features) (i.e., considering the rank of the feature), the rank of SCC is calculated by using Eq. (2). Where, R represents the SCC rank value, d_i^2 variation among two ranks (R_{f1}-R_{f2}) of each observation and n is the number of arranged paired. The CC value is ranged between -1 (strong negative correlation) and 1 (strong positive correlation). To visualized the linear relationship between the features set ($f_1, f_2, \ldots, f_{11}$) the heatmap utilized, see Figure 1.

$$\frac{\sum(a_i-a')(b_i-b')}{\sqrt{\sum(a_i-a)^2\sum(b_i-b)^2}} \quad (1)$$

$$R = 1 - \frac{6\sum d_i^2}{n(n-1)} \quad (2)$$

The heatmap visualization graph shows strong positive correlation among the WS (f_8) with IP_SRC (f_3), WS with MAC_DST (f_{10}), WS (f_8) with IP_SRC (f_3). Also, strong positive correlation between MAC_DST (f_{10}) and TTL(f_{11}), MAC_DST (f_{10}) and IP_SRC (f_3), Port_SRC (f_6) with protocol (f_5), packet length (f_2) with PIT (f_1) and Port_DST (f_7) with PIT (f_1). Besides, strong negative correlation among MAC_SRC (f_9) with MAC_DST (f_{10}), MAC_SRC (f_9) with Port_SRC (f_6), Protocol (f_5) with packet length (f_2), MAC_DST(f_{10}) with IP_DST (f_4), protocol (f_5) with PIT (f_1). In our case, the elected features are the independent variables and IoT devices calssifies (e.g., cam, hub, etc.) are the dependent variables. So, to estimate the probability p for a combination of selected features (independent variables) is performed by using the logit function Eq. (3) [21]. Where, *ln* is the natural logorithm, *p* logarithm of selected features that enables to predict regression coefficients (i.e., first temporary persecution).

$$Logit\ (p) = ln\frac{p}{1-p} \quad (3)$$

3.3 Classifier

In this phase an SNN is utilized as classier for the input features set from phase 3 to give a prediction set of classes (C_1, C_2, C_3, …, C_n). Where, each C represents a IoT Device. The SNN consists of a number of biological "synaptic" neurons. Each neuron is able to receive an input signal and produces an output signal, regardless of the actions for the rest of the other

neurons (i.e., neurons have an interior dynamic that causes biological "synaptic" neurons modify through time). So, when, the time exceeds the neuron rest to emptying and minimizing its membrane possibility. Subsequently, divides input spike must not reason a synaptic neuron to spike or fire [27, 28]. The neuron interconnects with each other via synaptic side with the weights. So, according to the synaptic weight modification, the learning process is performed v by using either supervised or an unsupervised approach [29]. However, the most public model that used to train SNN is "Synaptic Time Dependent Plasticity" STDP unsupervised method [30-32]. The STDP is used beside restrained fit spiking threshold to learn a representation for an input spike model that suitable for classification. The spikes are encoded via transforming the input signal into a series of spikes called "spiketrains" in a process called "encoding". The encoded process in STDP is trained via using "Leaky integrate neurons Fire" (LIF) [33, 34]. The temporal coding (rank code order (ROC)) method is used in this step. Where, in the temporal coding method, accurate timing of spikes and among action potentials isused to encode information. This involves the order that a group of nodes produces specific spikes. In the ROC is a method that 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). So, the ROC algorithm is applied on the output layer to get the output value, the order is calculated by utilizing Eq. (4), where *ne* is the elected output node, *nj* is the input node, the mod is the modulation factor that gives value in the range (0,1) and order(ne) is *nj* s' spiking order value, that established as results of the V encoding. To illustrate, let V=0.5, $W_{0,}$ ne=0.5 and order n_0 =4. Thus, predicted value (PV) 0.52=0.25 and according to the PV the device type 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 devices. Otherwise, the highest PV will be selected to identify the device type [34].

$$PV = \sum_{j=0}^{Winow\ time\ size-1} Wnj, ne\ mod^{order(ne)} \quad (4)$$

$$a_i(V) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{(x-\mu)^2}{\sigma^2}\right) \quad (5)$$

$$\mu_i = V_{min} + (V_{max} - V_{min}) . \frac{i}{n-1}\ for\ i = 0\ to\ n-1 \quad (6)$$

$$T = \max(a_i(V)) \quad (7)$$

$$W_i = W_i - b(\frac{\partial Error}{\partial Wi}) \quad (8)$$

In this study, the SNN is consisted of an input layer, one hidden layer and the output layer. To train the selected features value that obtained for the input layer a "Gaussian Receptive Fields" GRF algorithm, is utilized to encode information into firing times for the input layer by utilizing Eq. (5). Where, input value between (minimum data value (Vmin) and maximum data value Vmax) with σ is centered by utilizing Eq. (6). The spike timing is arranged between (0 to T), the *T* value is computed by using Eq. (7). The σ is allocated by the passing points of the V with corresponding Gaussian summits: the i-th input receives a spike at T-$a_i(V)$. So, when $a_i(V)$>0.01 and no spikes, then the nearest value of v to the σ will be taken, while the SNN modified synaptic weights in this study according to localized learning rules of STDP model. Also, the Backpropagation method is used in the hidden layer to update weight so as to alleviate error, see Eq. (8). Where, W_i, represents a new weight and (b) represent the learning rate (the minimum value) for the error function. The output layer gives the predicated classes (C_1, C_2, C_3, …, C_n), see Figure 2.

3.4 Evaluation model

To evaluate IoT-TCSNN model two ML approach are utilized: Support Vector Machine (SVM) and Deep learning Convolutional Neural Networks (CNN) in order to compare their performance with IoT-TCSNN model. The SVM is a supervised ML method, which make a binary classification from complicated nonlinear problem [35]. It needs data features (samples) to make a hyperplane, resolution surface and increasing the margin round it. Its subject training phase in which every data feature called as xi is allocated to the class called yi (predicted value). Thus, the training a group is labeled as (xi,yi), i=1,2,3,..., n where x ∈ $\mathbb{R}^n$ and y {-1,1}. So, the outcome of SVM is a group of support vectors which make the ideal hyperplane and the W (weight) that responsible to each input data feature which utilized to predicate the y value. While, CNN is an unsupervised ML model that consisted of a convolution layer (CL), Rectified linear unit (ReLU), Pooling layer (PL) and fully connected layer (FCL). In CL a series of filters are applied on input feature data to generate various output vectors for each filter together with a one weight. In ReLU layer, a negative value is maintained so as only positive vale are forward to the PL. In the PL, a pooling form from nonlinear down-sampling. The main aim of the PL is to reduce the number of parameters that network require to learn [36]. However, the performance of the three models (IoT-TCSVM, IoT-TCSNN and IoT-TCDNN) by is evaluated by using metrics: accuracy, precision, recall and F1_score that described in the next section.

4. RESULTS AND DISCUSSIONS

The IoT-TCSNN model have implemented on laptop type Lenovo (CPU speed 2.8 GHz Intel Core i7, RAM 8GB and operation system MS Window 10). Three scenarios have been conducted in order to, evaluate the performance of study method: one for an IoT-TCSNN model, the second one for the IoT-TCSVM method based on SVM approach and third scenario for the IoT-TCDNN on based CNN approach. The three scenarios implemented via using python language libraries: SNNTroch to implement (IoT-TCSNN) method and Tensorflow and panda libraries to implement IoT -TCSVM and IoT-TCDNN. For the first scenario, Threshold voltage Vth of input layer node (20 mV), Threshold voltage Vth of hidden/output layer node (65 mV) has been specified based on [34] study, see Table 3. While, For IoT-TCDNN, number of input neurons 120, hidden neurons 40, activation function (Relu), see Table 4. The evaluation of network performance has been performed via utilizing evaluation metrics: accuracy, precision, recall F1_score and energy usage that computed using Eqns. (9), (10), (11), (12), (13) and (14) respectively [37, 38]. Where, *TP* true positive, *TN* true negative, *FN* false negative, false positive *FP*, $Energy_{Tx}$ describes the amount of energy usage, which needed to convey (k) data packet for (d) distance between a pair of nodes, $Energy_{Rx}$ describes the amount of energy usage that required to get (*k*) for (*d*) between a pair of nodes.

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

$$Precision = \frac{TP}{TP+FP} \quad (10)$$

$$Recall = \frac{TP}{TP+FN} \quad (11)$$

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision+Recall} \quad (12)$$

$$Energy_{Tx}(d,k) = \begin{cases} kElec + k\varepsilon amp\ d^2, d < d_0 \\ kElec + k\varepsilon amp\ d^4, d \geq d_0 \end{cases} \quad (13)$$

$$Energy_{Rx}(k) = kElec + kEpa \quad (14)$$

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	20 mV
Threshold voltage V_{th} of hidden/output layer node	65 mV
Membrane resistance (all nodes)	1 MΩ
Membrane time constant (all nodes)	20 ms

Table 4. Parameters details for IDS-DNN

Parameter	Value
Input neuron	120
Hidden neuron	40
Activation function	Relu
Epochs	120/20
Batch size	64
Optimizer	Adam
Dropout rate	0.9

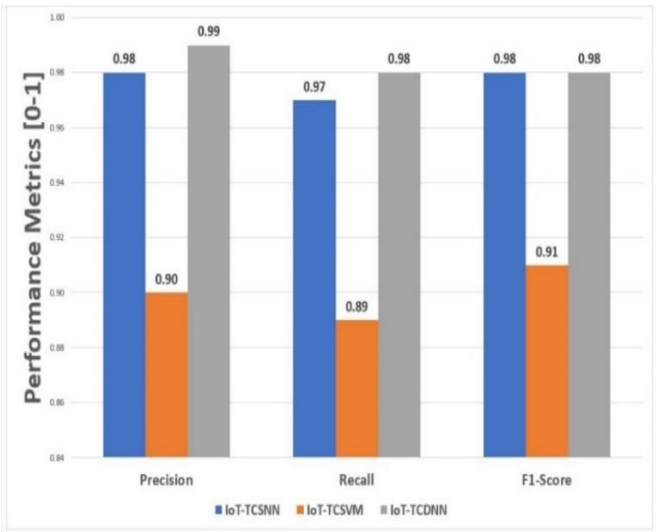

Figure 3. Performance evaluation based on precision, recall and F1-score

Table 5. Performance evaluation metric of classifiers

Model	Training Accuracy	Testing Accuracy
IoT-TCSNN	99.00	0.98
IoT-TCSVM	89.00	90.80
IoT-TCDNN	99.80	99.75

The performance metrics have shown that IoT-TCSNN gives model higher precision (value 0.98), Recall (value= 0.97) and F1-Score (value=0.98) in comparison with IoT-TCSVM where precision (value 0.90), Recall (value= 0.89) and F1-Score (value=0.91). While, IoT-TCDNN model gives the same F1-Score value of IoT-TCSNN model and higher value precision (value 0.99), Recall (value= 0.98) in contrast of IoT-TCSNN model, see Figure 3. For Accuracy, the IoT-TCSNN gives higher (training accuracy= 99.00 and testing accuracy= 99.80) in comparison with IoT-TCSVM (training accuracy= 89.00 and testing accuracy= 90.80). On the other side, the IoT-TCSNN gives less accuracy than the IoT-TCDNN model (where, training accuracy= 99.80 and testing accuracy= 99.75), see Table 5. For energy usage, the IoT-TCSNN consumed less energy in comparison with IoT-TCDNN and IoT-TCSVM model, see Figure 4.

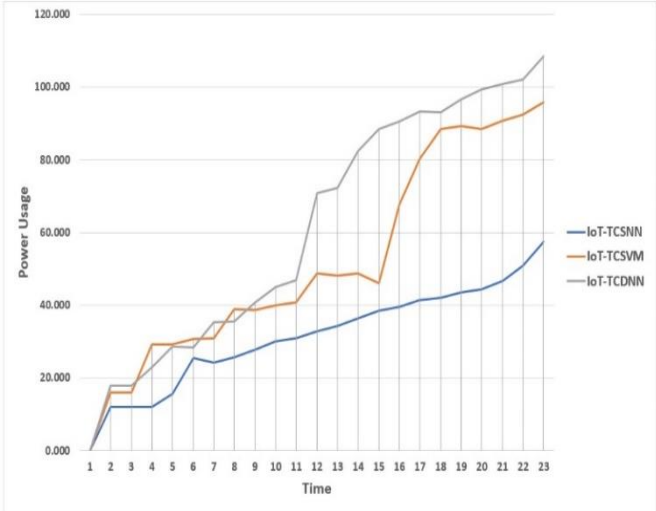

Figure 4. Performance evaluation for power usage

5. CONCLUSION

This study proposes a new model IoT-TCSNN for classifying IoT devices network traffic so as to optimize the network performance. The model classifies IoT devices traffic based on SNN and statistical characteristics of the devices' features. The model consists of four levels: data preprocessing (cleaning data and converting string data to number), feature extraction according to the linear relationship among features by using PCC and SNN method, classier devices level (predicates IoT-device via utilizing SNN) and finally model has evaluated according to the performance evaluation metrics: accuracy, precision, recall and F1-score with two models: IoT-TCSVM and IoT-TCDNN. However, three scenarios have been implemented for the three models using python language. The evaluation results have been shown that IoT-TCSNN consumes less energy in comparison with IoT-TCSVM and IoT-TCDNN. Also, IoT-TCSNN model shows high accuracy in comparison with IoT-TCSVM model and less accuracy in comparison with IoT-TCDNN.

ACKNOWLEDGMENT

We would like to appreciate all the excellent suggestions of anonymous reviewers to enhance the quality of this paper.

REFERENCES

[1] Castilho, S.D., Godoy, E.P., Salmen, F. (2020). Implementing security and trust in IoT/M2M using middleware. In 2020 International Conference on Information Networking (ICOIN), Barcelona, Spain, pp. 726-731. https://doi.org/10.1109/ICOIN48656.2020.9016435

[2] Lilhore, U.K., Imoize, A.L., Li, C.T., Simaiya, S., Pani, S.K., Goyal, N., Kumar, A., Lee, C.C. (2022). Design and implementation of an ML and IoT based Adaptive Traffic-management system for smart cities. Sensors, 22(8): 2908. https://doi.org/10.3390/s22082908

[3] Yue, Y., Li, S., Legg, P., Li, F. (2021). Deep learning-based security behaviour analysis in IoT environments: A survey. Security and Communication Networks, 2021: 1-13. https://doi.org/10.1155/2021/8873195

[4] Abbasi, M., Shahraki, A., Taherkordi, A. (2021). Deep learning for network traffic monitoring and analysis (NTMA): A survey. Computer Communications, 170: 19-41. https://doi.org/10.1016/j.comcom.2021.01.021

[5] Neelakandan, S., Berlin, M.A., Tripathi, S., Devi, V.B., Bhardwaj, I., Arulkumar, N. (2021). IoT-based traffic prediction and traffic signal control system for smart city. Soft Computing, 25(18): 12241-12248. https://doi.org/10.1007/s00500-021-05896-x

[6] Li, Y., Su, X., Ding, A.Y., Lindgren, A., Liu, X., Prehofer, C., Riekki, J., Rahmani, R., Tarkoma, S., Hui, P. (2020). Enhancing the internet of things with knowledge-driven software-defined networking technology: Future perspectives. Sensors, 20(12): 3459. https://doi.org/10.3390/s20123459

[7] Aouedi, O., Piamrat, K., Parrein, B. (2022). Intelligent traffic management in next-generation networks. Future Internet, 14(2): 44. https://doi.org/10.3390/fi14020044

[8] Ibrahim, H.A.H., Zuobi, O.R.A., Abaker, A.M., Alzghoul, M.B. (2021). A hybrid online classifier system for internet traffic based on statistical machine learning approach and flow port number. Applied Sciences, 11(24): 12113. https://doi.org/10.3390/app112412113

[9] Lim, H.K., Kim, J.B., Kim, K., Hong, Y.G., Han, Y.H. (2019). Payload-based traffic classification using multi-layer lstm in software defined networks. Applied Sciences, 9(12): 2550. https://doi.org/10.3390/app9122550

[10] Tahaei, H., Afifi, F., Asemi, A., Zaki, F., Anuar, N.B. (2020). The rise of traffic classification in IoT networks: A survey. Journal of Network and Computer Applications, 154: 102538. https://doi.org/10.1016/j.jnca.2020.102538

[11] Balamurugan, N.M., Adimoolam, M., Alsharif, M.H., Uthansakul, P. (2022). A Novel method for improved network traffic prediction using enhanced deep reinforcement learning algorithm. Sensors, 22(13): 5006. https://doi.org/10.3390/s22135006

[12] Jadav, N., Dutta, N., Sarma, H.K.D., Pricop, E., Tanwar, S. (2021). A machine learning approach to classify network traffic. In 2021 13th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), Pitesti, Romania, pp. 1-6. https://doi.org/10.1109/ECAI52376.2021.9515039

[13] University of New Souths Wales. IoT Traffic Traces. https://iotanalytics.unsw.edu.au/iottraces, accessed on Jul. 27, 2022.

[14] Umair, M.B., Iqbal, Z., Bilal, M., Almohamad, T.A., Nebhen, J., Mehmood, R.M. (2021). An efficient internet traffic classification system using deep learning for IoT. Computers, Materials & Continua (CMC), 71(1): 407-420. https://doi.org/10.32604/cmc.2022.020727

[15] Duan, L., Zhou, J., Wu, Y., Xu, W. (2022). A novel and highly efficient botnet detection algorithm based on network traffic analysis of smart systems. International Journal of Distributed Sensor Networks, 18(3): 1-17. https://doi.org/10.1177/15501477211049910

[16] Xin, L., Ziang, L., Yingli, Z., Wenqiang, Z., Dong, L., Qingguo, Z. (2022). TCN enhanced novel malicious traffic detection for IoT devices. Connection Science, 34(1): 1322-1341. https://doi.org/10.1080/09540091.2022.2067124

[17] Khedkar, S.P., Ramalingam, A.C. (2021). Identification of network traffic over IOT platforms. Revue d'Intelligence Artificielle, 35(4): 349-357. https://doi.org/10.18280/ria.350410

[18] Zahid, H.M., Saleem, Y., Hayat, F., Khan, F.Z., Alroobaea, R., Almansour, F., Ahmad, M., Ali, I. (2022). A Framework for Identification and Classification of IoT Devices for Security Analysis in Heterogeneous Network. Wireless Communications and Mobile Computing, 2022: 8806184. https://doi.org/10.1155/2022/8806184

[19] De Resende, A.A., De Melo, P.H., Souza, J.R., Cattelan, R.G., Miani, R.S. (2022). Traffic classification of home network devices using supervised learning. I In Proceedings of the 14th International Conference on Agents and Artificial Intelligence (ICAART 2022), pp. 114-120. https://doi.org/10.5220/0010785500003116

[20] Cvitić, I., Peraković, D., Periša, M., Stojanović, M.D. (2021). Novel classification of IoT devices based on traffic flow features. Journal of Organizational and End User Computing (JOEUC), 33(6): 1-20. https://doi.org/10.4018/JOEUC.20211101.oa12

[21] Hameed, A., Violos, J., Leivadeas, A. (2022). A deep learning approach for IoT traffic multi-classification in a smart-city scenario. IEEE Access, 10: 21193-21210. https://doi.org/10.1109/ACCESS.2022.3153331

[22] Nakip, M., Gül, B.C., Rodoplu, V., Güzeliş, C. (2022). Predictability of Internet of Things traffic at the medium access control layer against information-theoretic bounds. IEEE Access, 10: 55602-55615. https://doi.org/10.1109/ACCESS.2022.3174126

[23] Chakraborty, B., Divakaran, D.M., Nevat, I., Peters, G.W., Gurusamy, M. (2021). Cost-aware feature selection for IoT device classification. IEEE Internet of Things Journal, 8(14): 11052-11064. https://doi.org/10.1109/JIOT.2021.3051480

[24] Rasteh, A., Delpech, F., Aguilar-Melchor, C., Zimmer, R., Shouraki, S.B., Masquelier, T. (2022). Encrypted internet traffic classification using a supervised spiking neural network. Neurocomputing, 503: 272-282. https://doi.org/10.1016/j.neucom.2022.06.055

[25] Jia, S., Ma, Y., Xue, J., Zhu, A. (2022). Securing AI-powered Internet of Things (IoT) ecosystems. Wireless Communications and Mobile Computing, 2022: 9058048.

[26] Alghamdi, R., Bellaiche, M. (2022). Evaluation and selection models for ensemble intrusion detection systems in IoT. IoT, 3(2): 285-314. https://doi.org/10.3390/iot3020017

[27] Ali Abd Al-Hameed, K. (2022). Spearman's correlation

coefficient in statistical analysis. International Journal of Nonlinear Analysis and Applications, 13(1): 3249-3255. https://doi.org/10.22075/ijnaa.2022.6079

[28] He, W., Wu, Y., Deng, L., Li, G., Wang, H., Tian, Y., Ding, W., Wang, W.H., Xie, Y. (2020). Comparing SNNs and RNNs on neuromorphic vision datasets: Similarities and differences. Neural Networks, 132: 108-120. https://doi.org/10.1016/j.neunet.2020.08.001

[29] Dora, S., Kasabov, N. (2021). Spiking neural networks for computational intelligence: an overview. Big Data and Cognitive Computing, 5(4): 67. https://doi.org/10.3390/bdcc5040067

[30] Schwab, B.C., König, P., Engel, A.K. (2021). Spike-timing-dependent plasticity can account for connectivity aftereffects of dual-site transcranial alternating current stimulation. NeuroImage, 237: 118179. https://doi.org/10.1016/j.neuroimage.2021.118179

[31] Aljamali, N.A.S. (2020). Convolutional multi-Spike Neural Network as intelligent system prediction for control systems. Journal of Engineering, 26(11): 184-194. https://doi.org/10.31026/j.eng.2020.11.12

[32] Majeed, A.D., Al-Jamali, N.A.S. (2021). Spike Neural Network as a controller in SDN network. Journal of Engineering, 27(9): 64-77. https://doi.org/10.31026/j.eng.2021.09.06

[33] Samardzic, N.M., Bajic, J.S., Sekulic, D.L., Dautovic, S. (2022). Volatile memristor in leaky integrate-and-fire neurons: Circuit simulation and experimental study. Electronics, 11(6): 894. https://doi.org/10.3390/electronics11060894

[34] Zarzoor, A.R., Al-Jamali, N.A.S., Qader, D.A.A. (2023). Intrusion detection method for internet of things based on the spiking neural network and decision tree method. International Journal of Electrical and Computer Engineering, 13(2): 2278-2288. https://doi.org/10.11591/ijece.v13i2.pp2278-2288

[35] Ioannou, C., Vassiliou, V. (2021). Network attack classification in IoT using support vector machines. Journal of Sensor and Actuator Networks, 10(3): 58. https://doi.org/10.3390/jsan10030058

[36] Lakshmanna, K., Kaluri, R., Gundluru, N., Alzamil, Z.S., Rajput, D.S., Khan, A.A., Haq, M.A., Alhussen, A. (2022). A review on deep learning techniques for IoT data. Electronics, 11(10): 1604. https://doi.org/10.3390/electronics11101604

[37] Alani, M.M., Miri, A. (2022). Towards an explainable universal feature set for IoT intrusion detection. Sensors, 22(15): 5690. https://doi.org/10.3390/s22155690

[38] Zarzoor, A.R. (2022). Enhancing IoT performance via using Mobility Aware for dynamic RPL routing protocol technique (MA-RPL). International Journal of Electronics and Telecommunications, 68(2): 187-191. https://doi.org/10.24425/ijet.2022.139866