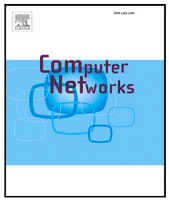
Computer Networks 237 (2023) 110072

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet



TNN-IDS: Transformer neural network-based intrusion detection system for MQTT-enabled IoT Networks

Safi Ullah [b,e], Jawad Ahmad [a,*], Muazzam A. Khan [b,c], Mohammed S. Alshehri [d], Wadii Boulila [e,f], Anis Koubaa [e], Sana Ullah Jan [a], M Munawwar Iqbal Ch [g]

[a] School of Computing, Engineering and The Built Environment, Edinburgh Napier University, Edinburgh, EH10 5DT, United Kingdom
[b] Department of Computer Science, Quaid-i-Azam University, Islamabad 44000, Pakistan
[c] Pakistan Academy of Sciences, Islamabad 44000, Pakistan
[d] Departments of Computer Science, College of Computer Science and Information Systems, Najran University, Najran 61441, Saudi Arabia
[e] Robotics and Internet-of-Things Laboratory, Prince Sultan University, Riyadh 12435, Saudi Arabia
[f] RIADI Laboratory, University of Manouba, Manouba 2010, Tunisia
[g] Institute of Information Technology, Quaid-i-Azam University, Islamabad 44000, Pakistan

ARTICLE INFO

Dataset link: https://ieee-dataport.org/open-access/mqtt-iot-ids2020-mqtt-internet-things-intrusion-detection-dataset

Index terms:
Intrusion detection
MQTT
Transformer neural network

ABSTRACT

The Internet of Things (IoT) is a global network that connects a large number of smart devices. MQTT is a de facto standard, lightweight, and reliable protocol for machine-to-machine communication, widely adopted in IoT networks. Various smart devices within these networks are employed to handle sensitive information. However, the scale and openness of IoT networks make them highly vulnerable to security breaches and attacks, such as eavesdropping, weak authentication, and malicious payloads. Hence, there is a need for advanced machine learning (ML) and deep learning (DL)-based intrusion detection systems (IDS). Existing ML-based IoT-IDSs face several limitations in effectively detecting malicious activities, mainly due to imbalanced training data. To address this, this study introduces a transformer neural network-based intrusion detection system (TNN-IDS) specifically designed for MQTT-enabled IoT networks. The proposed approach aims to enhance the detection of malicious activities within these networks. The TNN-IDS leverages the parallel processing capability of the Transformer Neural Network, which accelerates the learning process and results in improved detection of malicious attacks. To evaluate the performance of the proposed system, it was compared with various IDSs based on ML and DL approaches. The experimental results demonstrate that the proposed TNN-IDS outperforms other systems in terms of detecting malicious activity. The TNN-IDS achieved optimum accuracies reaching 99.9% in detecting malicious activities.

1. Introduction

The Internet of Things (IoT) is one of the most advanced technologies that have grabbed the curiosity of industrial researchers and practitioners. The growing propagation of the IoT has motivated the integration of IoT features in various domains, including agriculture, health, smart security, air, and water pollution, vehicles, smart cities, and smart homes [1–3]. Indeed, the IoT has grown into a worldwide network that connects smart devices anywhere in the World at any time by providing them with a unique identity [4–8]. The functioning of IoT is made possible by the implementation of several protocols [9].

Currently, there are several IoT protocols, such as the Advanced Message Queuing Protocol (AMQP), the Message-Queuing Telemetry Transport (MQTT) protocol, the Constrained Application Protocol (CoAP), and the Extensible Messaging and Presence Protocol (XMPP) in which MQTT is widely used in various IoT applications [10–12]. The MQTT protocol was first developed in 1999 by the International Business Machines (IBM) Corporation and Eurotech developers. It is a lightweight protocol with a quick response that supports low-power smart devices [13]. MQTT enables smooth communication with low bandwidth in an IoT network, regardless of the number of connected devices. This protocol consists of four major components: IoT nodes, the topic, the message, and a broker. The IoT nodes can be publishers, subscribers, or both. The publisher writes a message on a specific topic and publishes it via a broker. The subscribers receive informational messages about subscribed topics from the broker [14]. The architecture of the MQTT-based IoT network is shown in Fig. 1.

When the channel connects IoT components, various attacks are possible, more specifically on secret information such as confidential

* Corresponding author.
E-mail address: j.ahmad@napier.ac.uk (J. Ahmad).

https://doi.org/10.1016/j.comnet.2023.110072
Received 21 February 2023; Received in revised form 27 September 2023; Accepted 15 October 2023
Available online 16 October 2023

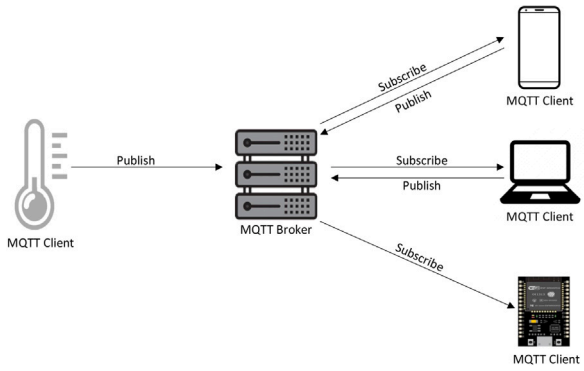


Fig. 1. MQTT protocol-enabled IoT architecture.

military information and private patient data [15–18]. In this context, the number of cyberattacks is increasing with the proliferation of IoT devices [19–21]. Accordingly, network security is a critical factor in deploying IoT networks. In this context, an IDS is often used to analyze network traffic to detect malicious activity [22–25]. In recent years, the ML and DL-based IDS have become more popular than those trained by real-time network data and used for auto-detection of cyberattacks [26,27]. DL is a subset of ML that consists of several layers and computational neurons that extract high-level features from the input set. It can handle new problems and achieve high-performance results in several areas of research [28–33]. While traditional ML techniques can achieve high accuracy on certain data, they have limitations when it comes to modeling complex relationships between inputs and outputs. However, the existing IDS for IoT networks face difficulties in effectively detecting malicious activities. Several existing systems are designed to handle intrusion detection tasks with balanced datasets. However, they may not be effective in dealing with the challenges posed by multi-class learning with imbalanced datasets [34]. Additionally, one of the most common challenges faced by existing systems is the selection of significant features for model training. Moreover, current systems often exhibit slow learning power and typically require multiple iterations to complete the training process. To overcome these problems, this paper proposes a Transformer Neural Network-based intrusion detection system (TNN-IDS) for the MQTT-enabled IoT networks to improve malicious activity detection in these networks. The significant advantage of the TNN has the ability to model large amounts of data and learn complex relationships between inputs and outputs, leading to improved accuracy and performance on many tasks. Multi-head attention layer divides input features into an equal number of heads and processes them in parallel, which provides improved outcomes [35,36].

The MQTT messages are typically represented as sequences of packets. The TNN architecture is well-suited for sequence modeling tasks, as it has the ability to capture dependencies between different segments of a sequence. These qualities of TNN make them capable of detecting anomalies in MQTT message sequences. The performance of the proposed TNN-IDS was evaluated on the MQTT-IoT-IDS2020 dataset. This dataset was the first dataset to simulate an MQTT-based network. It consists of three types of intuitive-level (also referred to as abstract-level) features: unidirectional flow (Uni-Flow), bidirectional flow (Bi-Flow), and Packet-Flow features [37]. Additionally, the MQTT-IoT-IDS2020 dataset contains five classes of data, one corresponding to normal activity (normal) and four corresponding to four sorts of malicious activities, or network attacks, which are aggressive-scan attacks, User Datagram Protocol scan (UDP-scan) attacks, Sparta SSH-brute-force attacks, and MQTT brute-force attacks. Moreover, the experiments were conducted for traditional ML and other DL systems, and their results were compared with those of the proposed system. The main contributions of this work are:

- A TNN-based IDS model is developed for the MQTT protocol that can detect intrusions with higher accuracy due to its potential for dividing input features into equal heads and processing them in parallel. A multi-head attention layer is used to process multiple vectors in parallel.
- The MQTT-IoT-IDS2020 dataset is preprocessed for missing values that generally negatively impact the performance of the IDS. Many cleaning techniques are used in the data preparation step to remove missing values.
- The features extraction method is adapted to select the most discriminating features among all that help improve the performance of malicious activity detection. The extra tree classifier (ETC) method is used to extract features.
- Experiments were conducted to prove the efficacy of the proposed approach against other ML and DL methods. All preprocessing steps used in comparing the proposed and other models were the same.

The rest of this article is arranged as follows. Section 2 presents a review of relevant literature and a table summary of the reviewed literature. Details on the proposed system are given in Section 3. Section 4 presents the results of experiments on the proposed IDS and compares them with those of other ML and DL-based systems. Finally, Section 5 concludes the article.

2. Related work

In recent years, many researchers worked on detecting intrusions in IoT networks. This section reviews some of their efforts and findings. To start with, Mehedi et al. [38] presented a residual neural network paradigm (P-ResNet) for detecting malicious activities in IoT networks. They focused primarily on feature engineering and a small amount of label data. The experimentation results uncovered that their proposed system outperformed other systems in detecting malicious activities in IoT networks. Wahab [39] proposed a dynamic deep neural network (DDNN) IDS for IoT networks. They used the principal component analysis (PCA) technique to observe the variance in the intrusion data features. They employed the DS2OS dataset in their experiments to evaluate the performance of their proposed system. Rashid et al. [40] introduced a stacking ensemble technique (SET) to identify harmful activities in IoT networks. Two public-domain datasets, that is, NSL-KDD and UNSW-NB15, were used in experiments in order to assess the performance of the SET. Additionally, these researchers used feature engineering to increase the efficiency of the proposed system. Gyamfi et al. [41] suggested an online incremental support vector data description (OI-SVDD) system for intrusion detection in industrial IoT and multi-access edge computing servers. They employed the UNSW-NB15 and self-generated datasets to evaluate the efficiency of their proposed system. Liu et al. [42] introduced a particle swarm optimization-based gradient descent (PSO-LightGBM) system for detecting malicious attacks in IoT network communications. This system was basically used for feature engineering, and a support vector machine (SVM) was employed for the prediction. These researchers used the UNSW-NB15 dataset to evaluate the performance of the proposed system. Attota et al. [43] designed a multi-view federated learning (MV-FLID) system that trains on distributed data and detects malicious activities in an IoT network. They improved the learning efficacy of the IDS for different classes of attacks. The proposed system outperformed traditional non-Federal Learning (non-FL) centralized approaches. Khan et al. [12] proposed a deep neural network (DNN) to classify normal data flow and malicious attacks in IoT networks. They ran this system on the MQTT-IoT-IDS2020 data to evaluate its efficacy and compared it with those of traditional ML-based IDS. Mosaiyebzadeh et al. [44] suggested three DL-based IDS for the prediction of intrusions in MQTT-enabled networks, which are DNN, long short-term memory (LSTM), and a combination of convolutional and recurrent neural networks (CNN+RNN+LSTM). They used the MQTT-IoT-IDS2020 dataset to evaluate these systems.

3. Proposed approach

This section introduces the system proposed in this study for detecting malicious activity in MQTT-enabled IoT networks. The section, first, describes the MQTT-IoT-IDS2020 dataset. Next, it explains the preprocessing and feature engineering techniques. Then, the proposed Transformer Neural Network (TNN) model and the adopted evaluation metrics are described in detail.

3.1. The MQTT-IoT-IDS2020 dataset

The MQTT-IoT-IDS2020 dataset was the first dataset to simulate an MQTT-based network [37]. It was generated by using 12 MQTT sensors, a broker, a simulating camera, and an attacker. Data were collected based on five scenarios: normal activity, aggressive-scan attack, UDP-scan attack, Sparta SSH-brute-force attack, and MQTT brute-force attack scenarios. For each scenario, data were recorded separately in pcap files. Three types of intuitive-level feature data (Uni-Flow, Bi-Flow, and Packet-Flow features) were extracted from the pcap files [45]. Each feature type contains 'csv' files for each of the aforementioned five scenarios. The Uni-Flow, Bi-Flow, and Packet-Flow feature data have 19, 32, and 31 features, respectively, including the label.

3.2. Preprocessing steps

Three preprocessing steps were applied in our experiments: data preparation, feature engineering, and normalization.

3.2.1. Data preparation

Data preparation is the initial stage of preprocessing in which the dataset is checked for internal problems. In general, if a dataset contains missing values, it may not be suitable for training a model as it can lead to inaccurate predictions [46]. Missing data can disrupt the learning process and impede the model's ability to recognize important patterns and relationships, which are essential for making accurate predictions [47]. Therefore, it is crucial to handle missing data appropriately before utilizing it in a prediction model [48]. The utilized dataset was thoroughly inspected for missing values, as they have the potential to introduce errors and inconsistencies in the prediction model, ultimately leading to biased outcomes. The employed dataset contained no missing values in the Uni-Flow and Bi-Flow features in the current study. However, the Packet-Flow features included missing values and mislabeled entries like 'is attack' string instead of 0 or 1 entries. All mislabeled entries were removed using the Python Data Analysis Library (Pandas) library of python. After that, every attribute of the dataset was scanned for missing values using the 'percent missing' method of the Pandas library.

Scanning unclosed nine attributes (mqtt messagetype, mqtt messagelength, mqtt flag name, mqtt flag passwd, mqtt flag retain, mqtt flag qos, mqtt flag willflag, mqtt flag clean, and mqtt flag reserved), each containing more than 80% missing values. These are the attributes common to all classes in a Packet-Flow. Filling in those missing values can lead to ambiguity, owing to that all empty cells will be filled with the same values. To tackle this problem, these nine attributes were removed from the dataset. The Packet-Flow file (scan sU) contains about 99% missing values in six attributes (tcp flag urg, tcp flag ack, tcp flag push, tcp flag reset, tcp flag syn, and tcp flag fin). Removing these attributes will create the problem of feature imbalance with other classes. To address this dilemma, the missing values were replaced with the mean of the same class in the attribute. In the Packet-Flow features, less than 8% of the values are missing in the remaining attributes of the other files. These missing value instances were removed from the dataset using the Pandas 'dropna' function. Distribution of the raw and clean data in the MQTT-IoT-IDS2020 dataset is shown in Table 1.

The MQTT-IoT-IDS 2020 dataset has some categorical features; it contains several data categories. The label encoder method was used in the present study to encode categorical features as numerical features. The dataset has binary-labeled data (0 and 1), where 0 represents normal activity, and 1 denotes an attack. The label was replaced with a unique number for each class. Then, all the unique numbers were merged into a single data frame in which the five classes of normal activity, aggressive-scan attack, UDP- scan attack, Sparta SSH-brute-force attack, and MQTT brute-force attack, were represented by 0, 1, 2, 3, and 4, respectively.

3.2.2. Feature engineering

Feature engineering is the process of reducing the overfitting, underfitting, unnecessary computational power, and extra memory consumption of the ML and DL-based IDS [49,50]. The goal of feature engineering is to enhance the performance of the classifier. In this study, the feature-filtering approach was used to extract the optimal set of features. In this approach, the ETC was employed to extract the features that had greater than 0 gain values. The gain value represents the effect of the feature on the output label. After filtering the features, exactly 18, 29, and 14 features were selected from the Uni-Flow, Bi-Flow, and Packet-Flow features, respectively, excluding the labels.

3.2.3. Normalization

Normalization is a preprocessing step that transforms all dataset attributes into a common scale. In the case of the ML and DL algorithms, not every dataset needs to be normalized. Instead, normalization is only necessary when the feature ranges vary. We initially examined the dataset for outliers using the Random Forest (RF) method, but no outliers were detected.

The dataset utilized in our experiment consists of various features with a wide range of values. Such variations in feature values can potentially have a negative impact on the model's performance [51]. For example, the 'tcp flag ack' feature of Packet Flow has a range between 0–1, while the 'ttl' feature range lies between 50–100. To address the differences in ranges of dataset features, the data were normalized to the range of 0–1 using the min–max normalization approach as demonstrated in Eq. (1). The min–max method is often utilized because of its simplicity and ease of implementation. Additionally, it preserves the original shape of the distribution, making it a desirable normalization.

$$X_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (1)$$

Afterwards, the normal and normalized data were split into a training sub-set of 80% of the dataset and a testing sub-set consisting of the rest 20% of the dataset to validate the proposed IDS. The stratified sampling method was used to split the data into homogeneous sub-sets.

3.3. Architecture of the proposed transformer neural network

The TNN was developed in the area of natural language processing (NLP) in 2017 [52]. It is a novel DL technique that was originally designed to solve sequence-to-sequence problems and deal with long-term dependencies [53]. It utilizes a self-attention method to accomplish effective parallel computation, which improves the learning process. By design, the TNN performs encoding and decoding processes. In NLP, encoding is implemented to encode one language, and decoding is performed to compute the probability of another language with the help of the previous output. Intrusion detection is a challenging task that requires careful consideration of the neural network architecture to achieve considerable performance compared to baselines. While off-the-shelf transformer models such as BERT and GPT have been successful in NLP tasks, they may not perform optimally in intrusion detection tasks because of the differences in the data and the objectives. BERT and GPT both utilize transformers, which consist of embedding layers, positional encoding, encoders, and decoders. In these models, the decoding block relies on the preceding output to generate words sequentially. However, not all of these components are necessary for

Table 1
Distribution of raw and clean data in the MQTT-IoT-IDS2020 dataset.

Data flow type	*Raw data*				*Clean data*	
	Features	Files	Classes	Instances	Clean features	Clean instances
Uni-Flow	19	normal	Normal	171836	19	171836
		mqtt_bruteforce	Attack	28874		28874
			Normal	4205		4205
		scan_A	Attack	39797		39797
			Normal	11561		11561
		scan_sU	Attack	34409		34409
			Normal	22436		22436
		sparta	Attack	154175		154175
			Normal	28232		28232
Bi-Flow	32	normal	Normal	86008	32	86008
		mqtt_bruteforce	Attack	14544		14544
			Normal	2152		2152
		scan_A	Attack	19907		19907
			Normal	5786		5786
		scan_sU	Attack	22434		22434
			Normal	17230		17230
		sparta	Attack	77202		77202
			Normal	14116		14116
Packet-Flow	31	normal	Normal	1056230	22	964834
			Mislabel	1		0
		mqtt_bruteforce	Attack	10013142		10010556
			Normal	32164		2434
			Mislabel	10		0
		scan_A	Attack	40624		40488
			Normal	70768		64531
		scan_sU	Attack	22436		22436
			Normal	210819		210818
		sparta	Attack	19728943		19728943
			Normal	947177		865694
			Mislabel	20		0

an intrusion detection model. One of the main challenges of using off-the-shelf transformer models for intrusion detection is the difference in input data format. In NLP tasks, the input is typically a sequence of words or tokens, whereas, in intrusion detection, the input is a sequence of network events or packets. As a result, the neural network architecture needs to be specifically designed to accommodate this input format. To overcome these challenges, this study adopted the TNN model to process the input data and identify malicious activities in the network.

The TNN model has a self-attention mechanism that enables it to capture dependencies between different segments of a sequence. This mechanism allows the TNN model to prioritize relevant segments of the input and assign varying levels of significance to each segment. It is particularly useful for analyzing sequential data with long-term dependencies, which is often the case in IDS. By utilizing the attention mechanism of the TNN, an IDS can learn to identify suspicious or anomalous patterns in network traffic. While there are other types of neural network architectures available for anomaly detection, the TNN's ability to capture complex relationships, model long-term dependencies, and handle sequential data makes it a suitable choice for IDS tasks.

The proposed TNN architecture consists of an encoder block, a global average pooling layer, and two dense layers. The encoder block contains a multi-head attention layer, two normalization layers, and two one-dimensional convolutional layers, as shown in Fig. 2.

The herein proposed IDS processes the input shape and describes it in terms of three characteristics, which are denoted as 'none', sequence length, and '1', where 'none' is the batch size, sequence length is the number of attributes, and '1' is the individual input time-series. Moreover, multi-head attention (also called self-attention) is used to compute the significance of each head, where significance is a one-dimensional vector. Meanwhile, the input sequence in self-attention is defined by eight heads [52]. The head size H_s is expressed in Eq. (2), where I_s and N_h denote the input sequence length, and the number of heads, respectively. Self-attention depends on queries (Q), keys (k), and values (V), and can be calculated by using Eq. (3), Eq. (4), and Eq. (5), respectively, where X is input vector and W is weight.

$$H_s = \lceil \frac{I_s}{N_h} \rceil \tag{2}$$

$$Q = XW_Q \tag{3}$$

$$K = XW_K \tag{4}$$

$$V = XW_V \tag{5}$$

The proposed TNN-IDS uses Eq. (6) to calculate S_i. Where S_i represents the calculated attention value for each head. To determine the output of the self-attention layer, the heads are all concatenated into a vector as illustrated in Eq. (7). The internal mechanism of the multi-head attention layer is shown in Fig. 3.

$$S_i = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_q}}\right)V \tag{6}$$

$$S = S_i\,(1, \ldots, n) \tag{7}$$

The output of the self-attention layer is passed to the feed-forward element of the TNN. In this step, two one-dimensional convolutional layers are used to extract high-level features as explained in Eqs. (8) and (9).

$$x_k = b_k + \sum_{i=1}^{N} (s_i, w_{ik}) \tag{8}$$

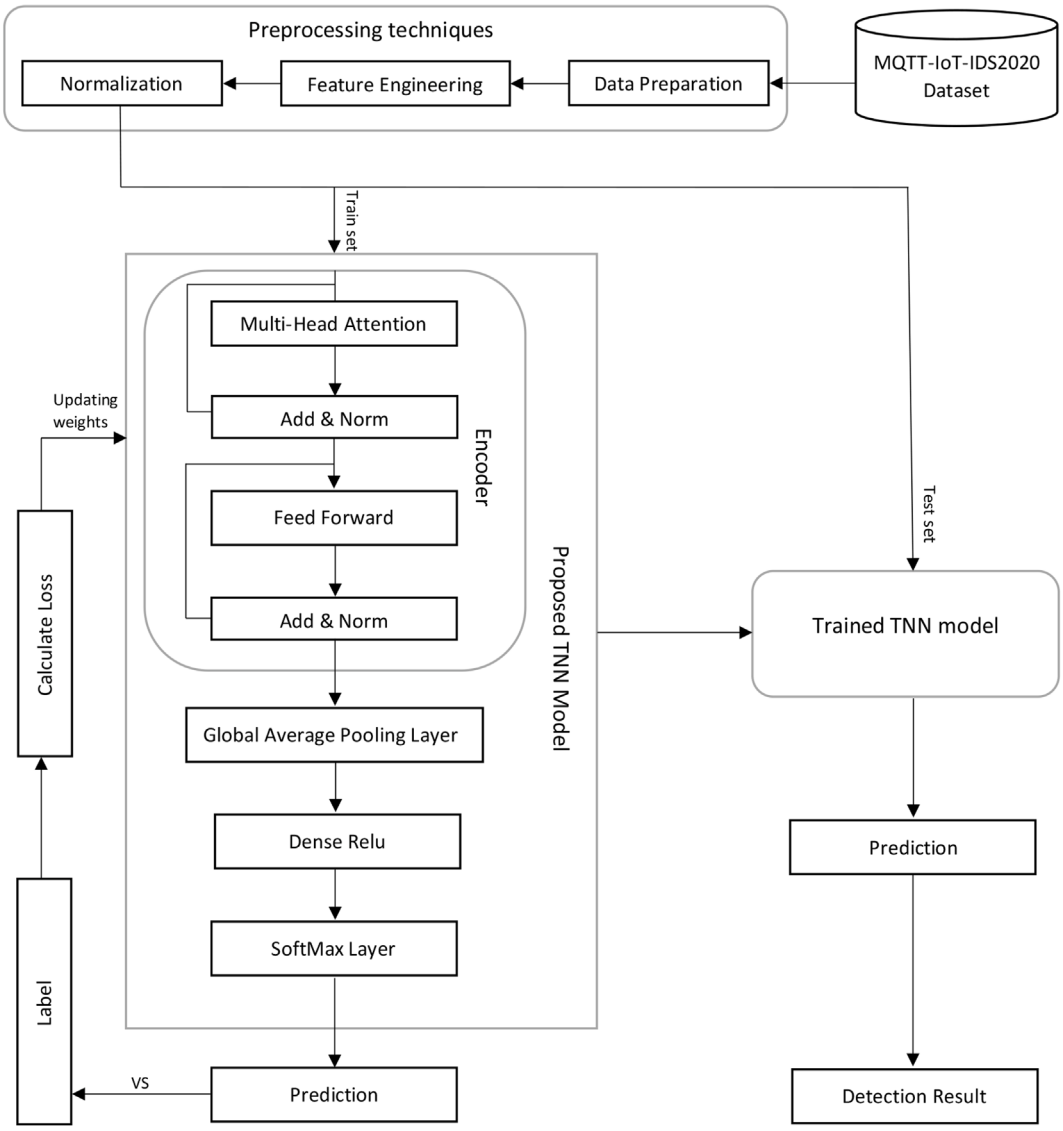


Fig. 2. Block diagram of the proposed approach.

$$y_k = \max(0, x_k) \tag{9}$$

Eight filters and a kernel size of '1' were used in the convolutional layers in the present study. Add and norm layers were used to encounter the vanishing gradient problem [54]. The output of the encoder block was fed into the global average pooling layer to compute the average value of each single map feature. This layer converts data into a single vector instead of adding extra layers to the network. A Relu-dense layer was added to the output of the previous layer. In Eq. (8) the parameter x_k denotes the samples input to the convolutional layer, s_i represents the neurons of the preceding layer, w_{ik} points to the kernel size, and b_k expresses the bias. In Eq. (9), however, the parameter y_k indicates the output of the convolutional layer in which the Relu function was used. Eventually, the softmax layer is used to generate the output. How the softmax is computed is explicated in Eq. (10).

$$\text{softmax}(x)_i = \frac{e^{x_i}}{\sum_{j=1}^{K} e^{x_j}} \tag{10}$$

3.4. Performance evaluation metrics

The effectiveness of the proposed TNN architecture was evaluated using the performance metrics of accuracy (ACC), precision (P), recall (R), and the F1 score. The proportion of right assessments is referred to as ACC. In the meantime, precision (P) refers to the ratio of accurately-detected positive instances relative to all the instances that have been identified as positive instances. Recall (R), on the other hand, is the ratio of properly-recognized positive instances to all positive instances. However, it should be highlighted that there are occasions when P and R contradict each other. They must be thoroughly examined in this case. In other respects, the F1 score is the most commonly used performance measure. It is the harmonic mean of the P and R. In this experiment, we employed the macro P, R, and F1 scores, which represent the average results across all classes. All four metrics (ACC, P, R, and the F1 scores) are calculated using the true positive (TP), false negative (FN), false positive (FP), and true negative (TN) parameters as illustrated in Eqs. (11)–(14).

$$\text{ACC} = \frac{TP + TN}{TP + TN + FP + FN} \tag{11}$$

$$\text{P} = \frac{TP}{TP + FP} \tag{12}$$

$$\text{R} = \frac{TP}{TP + FN} \tag{13}$$

$$\text{F1 Score} = \frac{2 \times (\text{P} \times \text{R})}{\text{P} + \text{R}} \tag{14}$$

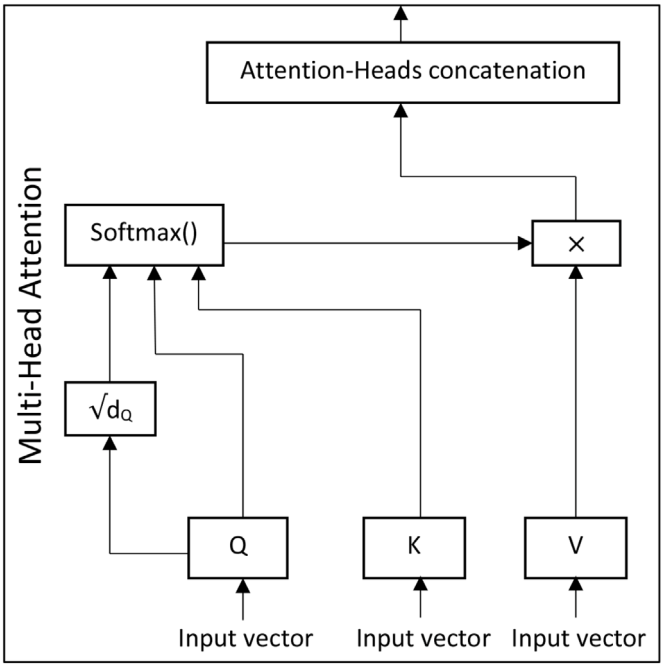


Fig. 3. Internal mechanism of self-attention layer.

3.5. Experimental settings

Experiments were conducted by running the proposed TNN-IDS on the MQTT-IoT-IDS2020 dataset using an HP ProBook G5 laptop with an 8th-generation Intel i5 processor and 24 GB RAM. The operating system was Windows 11, and the employed programming language was Python 3.8.5. Moreover, the experiments utilized the Jupyter notebook integrated development environment (IDE) and the Tensorflow and Keras libraries.

4. Performance evaluation

This section analyzes the outcomes of the testing performance of the proposed TNN-IDS. The MQTT-IoT-IDS2020 dataset was employed in the experiments. As was explicated earlier, this dataset consists of three abstract-level features (Uni-Flow, Bi-Flow, and Packet-Flow features). Four optimizers were used with the TNN-IDS, and performance was analyzed in each case based on multi-class classification. In addition, the performance of the proposed IDS was compared with levels of performance of ML and DL-based IDS under the same experimental settings.

4.1. Results analysis of proposed approach

Experiments were performed using the MQTT-IoT-IDS2020 dataset. Since this dataset has three abstract-level features, separate experiments were performed on each of these three data levels to accurately test the performance of the various IDS under consideration. In the experiments, the three well-known DL modifiers (stochastic gradient descent (SGD), root mean square propagation (RMSProp), and adaptive moment estimation (Adam)) were employed for 32, 64, and 128 batch sizes [55]. The proposed IDS was trained for six epochs, and the loss was computed by using the sparse categorical cross-entropy method. The k-fold cross-validation method was used to validate the proposed TNN IDS. In this regard, the k value was set at 5 in all experiments. With this approach, the dataset was divided into five equal parts or folds. The models were then trained on four of the folds, chosen at random, and tested on the remaining fold. This process was repeated five times, and the final result was the average of these experiments.

Table 2
Performance of the proposed IDS when run on the Uni-Flow features in combination with different optimizers.

Optimizer	Batch size	Accuracy	Precision	Recall	F1-Score
SGD	32	0.9994	0.9987	0.9990	0.9988
	64	0.9971	0.9967	0.9951	0.9959
	128	0.9914	0.9931	0.9862	0.9895
RMSprop	32	0.9997	0.9997	0.9996	0.9996
	64	0.9994	0.9987	0.9989	0.9988
	128	0.9998	0.9996	0.9996	0.9996
Adam	32	0.9999	0.9999	0.9999	0.9999
	64	0.9999	0.9998	0.9998	0.9998
	128	0.9994	0.9987	0.9989	0.9988

Table 3
Performance of the proposed IDS when run on the Bi-Flow features in combination with different optimizers.

Optimizer	Batch size	Accuracy	Precision	Recall	F1-Score
SGD	32	0.9974	0.9965	0.9946	0.9955
	64	0.9974	0.9965	0.9945	0.9954
	128	0.9932	0.9881	0.9878	0.9879
RMSprop	32	0.9996	0.9997	0.9994	0.9995
	64	0.9999	0.9999	0.9999	0.9999
	128	0.9989	0.9985	0.9978	0.9981
Adam	32	0.9998	0.9997	0.9996	0.9997
	64	0.9997	0.9987	0.9995	0.9996
	128	0.9997	0.9997	0.9994	0.9996

4.1.1. Performance evaluation of the proposed IDS for the uni-flow features

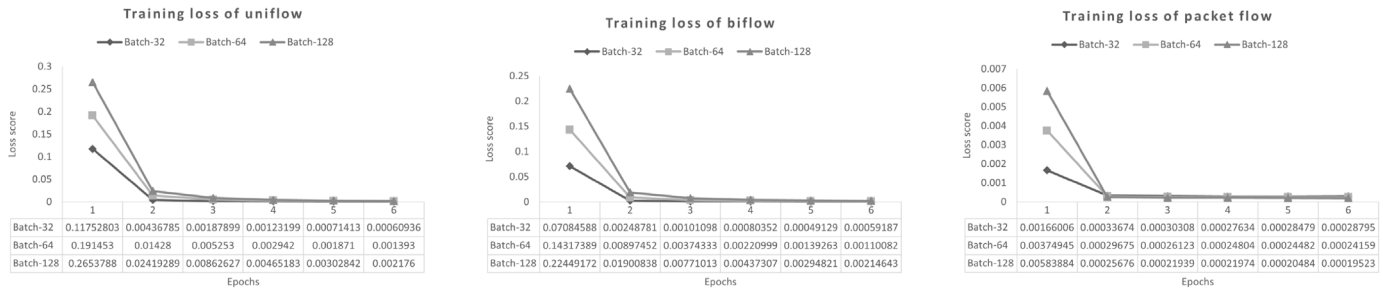
The proposed TNN-IDS was evaluated for the Uni-Flow features of the MQTT-IoT-IDS2020 dataset. The performance was assessed when it was run in combination with each of the SGD, RMSprop, and Adam optimizers. Simulation results are compiled in Table 2 for comparison purposes. These results indicate that the proposed IDS has the highest malicious activity detection performance of 99.99% for the Uni-Flow features and the batch size of 32 when this system is integrated with Adam optimizer. Values of all performance metrics are the same and optimum for Adam optimizer with batch size 32 (Table 2).

4.1.2. Performance evaluation of the proposed IDS for the bi-flow features

The performance of the proposed IDS was evaluated on the Bi-Flow features. As mentioned in the foregoing sub-section, the experiments were conducted on the proposed IDS when it was run in combination with each of the SGD, RMSprop, and Adam optimizers. For comparison purposes, the outputs of performance assessment are provided by Table 3. The experimentation results point out that the proposed TNN-IDS produced the best malicious activity detection results (99.99%) in combination with the RMSprop optimizer for the batch size of 64. Values of all the evaluation metrics were the same under these circumstances, i.e., 99.99% (Table 3).

4.1.3. Performance evaluation of the proposed IDS for the packet flow features

The performance of the proposed IDS was also evaluated using the Packet-Flow features in the cases when this system employed the SGD, RMSprop, and Adam optimizers. The performance assessment outcomes are compared in Table 4. The performance assessment results pinpoint that the proposed IDS had the greatest accuracy and precision values (99.99%, each). This value was obtained when this IDS was run in integration with Adam optimizer and applied to the three batch sizes under study, namely, 32, 64, and 128. Under these conditions, the values of the remaining performance evaluation metrics (Recall and the F1 score) were also somewhat high.



	1	2	3	4	5	6
Batch-32	0.11752803	0.00436785	0.00187899	0.00123199	0.00071413	0.00060936
Batch-64	0.191453	0.01428	0.005253	0.002942	0.001871	0.001393
Batch-128	0.2653788	0.02419289	0.00862627	0.00465183	0.00302842	0.002176

(a) Uniflow

	1	2	3	4	5	6
Batch-32	0.07084588	0.00248781	0.00101098	0.00080352	0.00049129	0.00059187
Batch-64	0.14317389	0.00897452	0.00374333	0.00220999	0.00139263	0.00110082
Batch-128	0.22449172	0.01900838	0.00771013	0.00437307	0.00294821	0.00214643

(b) Biflow

	1	2	3	4	5	6
Batch-32	0.00166006	0.00033674	0.00030308	0.00027634	0.00028479	0.00028795
Batch-64	0.00374945	0.00029675	0.00026123	0.00024804	0.00024482	0.00024159
Batch-128	0.00583884	0.00025676	0.00021939	0.00021974	0.00020484	0.00019523

(c) Packet flow

Fig. 4. Training loss of adam optimizer.

Table 4
Performance of the proposed IDS when run on the Packet-Flow features in combination with different optimizers.

Optimizer	Batch size	Accuracy	Precision	Recall	F1-Score
SGD	32	0.9998	0.9999	0.9868	0.9931
	64	0.9998	0.9992	0.9796	0.9889
	128	0.9997	0.9998	0.9772	0.9879
RMSprop	32	0.9937	0.9617	0.682	0.7434
	64	0.9998	0.9967	0.9859	0.9912
	128	0.9998	0.9997	0.9894	0.9944
Adam	32	0.9999	0.9999	0.9974	0.9987
	64	0.9999	0.9999	0.9968	0.9984
	128	0.9999	0.9999	0.9971	0.9985

Table 5
Average values of the metrics of the performance of the proposed IDS.

Data flow type	Optimizer	Avg accuracy	Avg precision	Avg recall	Avg F1-Score
Uni-Flow	SGD	0.9984	0.9983	0.9974	0.9978
	RMSprop	0.9984	0.9983	0.9974	0.9978
	Adam	0.9997	0.9994	0.9995	0.9995
Bi-Flow	SGD	0.9960	0.9937	0.9923	0.9929
	RMSprop	0.9994	0.9993	0.9990	0.9991
	Adam	0.9997	0.9993	0.9995	0.9996
Packet-Flow	SGD	0.9997	0.9996	0.9812	0.9899
	RMSprop	0.9977	0.9860	0.8857	0.9096
	Adam	0.9999	0.9999	0.9971	0.9985

4.1.4. Average results

The performance of the proposed TNN-IDS was analyzed for the Uni-Flow, Bi-Flow, and Packet-Flow features of the MQTT-IoT-IDS2020 dataset. Different levels of performance were associated with each of the three optimizers under study, i.e., the SGD algorithm, RMSprop, and Adam optimizers (Tables 2 through 5). The average values of the four considered performance evaluation measures are compared in Table 5 to specify the best optimizer for the proposed IDS. Based on the results presented in Table 5, the Adam optimizer produces better intrusion detection results than the other optimizers, almost equally for the Uni-Flow, Bi-Flow, and Packet-Flow features. Adam optimizer's performance levels in the training phase with the Uni-Flow, Bi-Flow, and Packet-Flow features are shown in Fig. 4 for the loss metric and in Fig. 5 for the accuracy metric. The training and testing times of the proposed model on the utilized dataset with various batch sizes are summarized in Table 6. This analysis reveals that increasing the batch size reduces both training and testing times for the model. Notably, when compared to other batch sizes, a batch size of 32 consumes more time. The performance evaluation indicates that the model performs optimally with a batch size of 32. However, increasing the batch size slightly degrades performance.

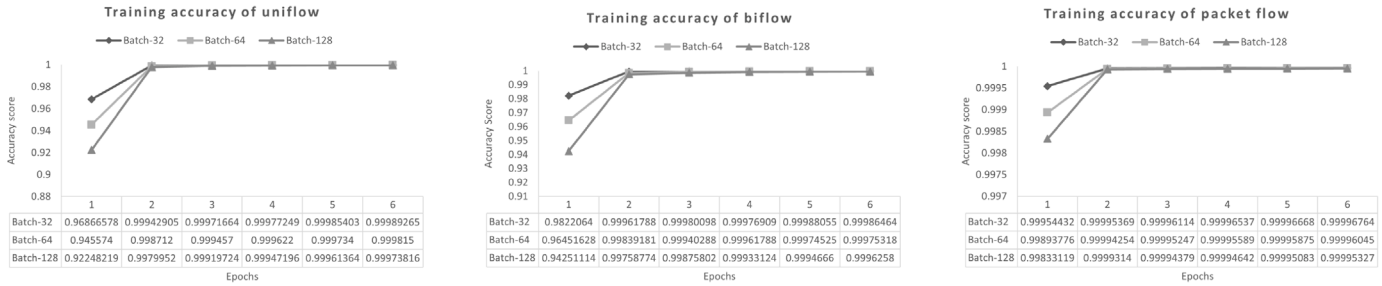
4.2. Performance comparison of the proposed TNN-IDS with other ML and DL-based systems

Experiments were also performed on the MQTT-IoTIDS2020 dataset using ML and DL-based IDS. The convolutional neural network (CNN), gated recurrent units (GRU), LSTM, multi-layer perceptron (MLP), deep neural network (DNN), and deep belief network (DBN) are well-known algorithms of DL. On the other hand, the Decision Tree (DT), Logistic Regression (LR), and Naive Bayes (NB) can be used as ML algorithms. In this study, the same preprocessing steps were applied to all the considered DL and ML-based IDS. The same preprocessing steps are important for a fair comparison with other ML and DL techniques. If different preprocessing techniques are used for different models, it becomes difficult to know whether any differences in performance are due to the model architecture or due to preprocessing. In the case of the DL-based IDS considered, the Adam optimizer and a batch size of 32 were adopted. DL IDS such as the CNN, GRU, LSTM, MLP, DNN, and DBN usually use 4, 2, 2, 5, 3, and 3 hidden layers, respectively. To assess the relative efficiency of the proposed TNN-IDS, its performance was compared with other ML-and DL-based IDS in Table 7 and with the other articles in Table 8. The comparison revealed that the proposed TNN-IDS demonstrates superior performance in detecting malicious activity compared to the investigated ML- and DL-based IDS, as well as other articles. All of these IDS perform better on the Uni-Flow and Bi-Flow features than on the Packet-Flow features, presumably because the Uni-Flow and Bi-Flow feature sets have almost balanced data, while the Packet-Flow feature set has imbalanced data. Balanced data refers to a dataset in which the number of instances for each class is relatively similar or only slightly different. On the other hand, imbalanced data refers to a dataset in which a substantial difference between the number of instances for each class. The proposed TNN-IDS produces almost identical intrusion detection results when applied to the Uni-Flow, Bi-Flow, and Packet-Flow features. This finding provides evidence that this proposed IDS is the optimum choice, both for balanced and imbalanced data.

Based on the analysis of resource utilization and time consumption, both the Uni-Flow and Bi-Flow versions required less memory and processing power for training and testing, resulting in shorter time requirements. In contrast, the Packet-Flow version consumed up to 98% of the system memory and processing power, leading to significantly longer training and testing times for the model. Overall, a larger dataset will necessitate more memory and processing resources, resulting in increased time requirements for both training and testing. Moreover, the results demonstrate the scalability of the proposed TNN model, which consistently performed well across all three data flow versions of the dataset. Notably, the Uni-Flow and Bi-Flow versions are more lightweight, while the Packet-Flow version is larger. The results for these versions are roughly the same, affirming the model's consistency with large datasets.

Table 6
Training and testing time analysis of the proposed model.

Data flow type	*Training time (in s)*			*Test time (in s)*		
	Batch size-32	Batch size-64	Batch size-128	Batch size-32	Batch size-64	Batch size-128
Uni-Flow	26	14	8	3	2	1
Bi-Flow	27	16	11	3	2	1
Packet-Flow	1914	1637	1225	305	252	166



(a) Uniflow (b) Biflow (c) Packet flow

Fig. 5. Training accuracy of adam optimizer.

Table 7
Performance comparison of the Proposed IDS with other ML- and DL-based systems.

Data flow type	Model	Accuracy	Precision	Recall	F1-Score
Uniflow	CNN	0.9933	0.9925	0.9899	0.9912
	GRU	0.9972	0.9948	0.9956	0.9952
	LSTM	0.9534	0.9502	0.9471	0.9468
	MLP	0.9771	0.9543	0.988	0.9682
	DNN	0.9828	0.9698	0.8865	0.9282
	DBN	0.9996	0.9994	0.9991	0.9993
	NB	0.9963	0.9973	0.9946	0.9959
	DT	0.9829	0.9826	0.7793	0.8692
	LR	0.9922	0.989	0.9966	0.9926
	Proposed TNN	**0.9999**	**0.9999**	**0.9999**	**0.9999**
Biflow	CNN	0.9955	0.9982	0.9918	0.9949
	GRU	0.9956	0.9928	0.9917	0.9922
	LSTM	0.9657	0.9517	0.9631	0.9571
	MLP	0.9997	0.9993	0.9993	0.9993
	DNN	0.9937	0.9916	0.9897	0.9906
	DBN	0.9997	0.9996	0.9996	0.9996
	NB	0.9968	0.9988	0.9933	0.996
	DT	0.9848	0.9681	0.8206	0.8883
	LR	0.9996	0.9995	0.9993	0.9994
	Proposed TNN	**0.9998**	**0.9997**	**0.9996**	**0.9997**
Packet flow	CNN	0.9913	0.9943	0.9191	0.9517
	GRU	0.9776	0.9117	0.8196	0.8132
	LSTM	0.9908	0.9718	0.9893	0.9801
	MLP	0.9878	0.7866	0.7510	0.7670
	DNN	0.9909	0.9946	0.9501	0.9712
	DBN	0.9928	0.9968	0.9525	0.9717
	NB	0.6697	0.6306	0.8522	0.658
	DT	0.9527	0.9139	0.9626	0.9376
	LR	0.9889	0.9866	0.9185	0.9484
	Proposed TNN	**0.9999**	**0.9999**	**0.9996**	**0.9997**

5. Conclusion and future work

This paper proposed TNN-IDS for the MQTT-enabled IoT networks, ultimately intended to improve malicious activity detection in these networks. The experimental results disclosed that the proposed IDS could enhance malicious activity detection profoundly in this kind of network. The performance of this proposed system was evaluated on the MQTT-IoT-IDS2020 dataset, which consists of three types of abstract-level data: Uni-Flow, Bi-Flow, and Packet-Flow feature data. Experiments were carried out on each type of data. The optimum accuracies of malicious activity detection that were obtained from the proposed IDS for the Uni-Flow, Bi-Flow, and Packet-Flow features were 99.97%, 99.97%, and 99.99%, respectively. These accuracies were associated with the Adam optimizer. In addition, the intrusion detection results of the TNN-IDS were compared with those of ML and DL-based IDS to appraise the proposed system's comparative efficiency. The comparison results support that the proposed IDS performance in malicious activity detection in the IoT networks is optimum. Furthermore, the experimental results indicate that this TNN-IDS can successfully handle the problem of imbalanced data. In the future, the proposed method can potentially be applied to other protocols as well, with appropriate modifications made to the features and parameters of the method. This flexibility enables the method to be adapted to the unique requirements of different protocols, thereby increasing its overall applicability and effectiveness.

CRediT authorship contribution statement

Safi Ullah: Conceptualization, Methodology, Writing – original draft. **Jawad Ahmad:** Conceptualization, Software, Validation, Resources, Writing – review & editing, Supervision, Project administration. **Muazzam A. Khan:** Conceptualization, Writing – original draft. **Mohammed S. Alshehri:** Validation, Investigation, Visualization, Funding acquisition. **Wadii Boulila:** Conceptualization, Formal analysis, Investigation, Writing – review & editing, Supervision, Project administration. **Anis Koubaa:** Software, Resources, Writing – review & editing. **Sana Ullah Jan:** Writing – original draft, Visualization. **M Munawwar Iqbal Ch:** Formal analysis, Visualization.

Declaration of competing interest

The authors have no conflicts of interest to declare. All coauthors have seen and agree with the contents of the manuscript and there is no conflict of interest to report. We certify that the submission is original work and is not under review at any other publication.

Data availability

The publicly available dataset can be found at: https://ieee-dataport.org/open-access/mqtt-iot-ids2020-mqtt-internet-things-intrusion-detection-dataset.

Table 8
Performance comparison of the Proposed IDS with other articles.

Data flow type	Article	Model	Accuracy	Precision	Recall	F1-Score
Uniflow	Khan et al. [12]	NB	0.7076	0.6918	0.7098	0.7007
		DT	0.9617	0.9614	0.7589	0.8482
		DNN	0.9708	0.9476	0.8643	0.906
	Mosaiyebzadeh et al. [44]	LSTM	0.9968	**1**	0.9900	**1**
		DNN	0.9952	**1**	0.9900	**1**
		CNN-RNN-LSTM	0.9959	**1**	0.9900	**1**
	This study	**Proposed TNN**	**0.9999**	0.9999	**0.9999**	0.9999
Biflow	Khan et al. [12]	NB	0.9118	0.9199	0.7937	0.8522
		DT	0.9703	0.9547	0.8099	0.8764
		DNN	0.9812	0.9510	0.8671	0.9071
	Mosaiyebzadeh et al. [44]	LSTM	0.9964	**1**	0.9900	0.9900
		DNN	0.9959	**1**	0.9900	0.9900
		CNN-RNN-LSTM	0.9964	**1**	0.9900	0.9900
	This study	**Proposed TNN**	**0.9998**	0.9997	**0.9996**	**0.9997**
Packet flow	Khan et al. [12]	NB	0.4876	0.7033	0.4915	0.5786
		DT	0.9053	0.8983	0.7919	0.8417
		DNN	0.9079	0.8941	0.8164	0.8534
	Mosaiyebzadeh et al. [44]	LSTM	0.9192	0.9200	**1**	0.9600
		DNN	0.9193	0.9200	**1**	0.9600
		CNN-RNN-LSTM	0.9204	0.9200	**1**	0.9600
	This study	**Proposed TNN**	**0.9999**	**0.9999**	0.9996	**0.9997**

Acknowledgments

The authors are thankful to the Deanship of Scientific Research at Najran University, Saudi Arabia for funding this work under the Research Groups Funding Program grant code (NU/RG/SERC/12/3). The authors would like to thank Prince Sultan University for their support.

References

[1] L. Rashid, S. Rubab, M. Alhaisoni, A. Alqahtani, S. Alsubai, A. Binbusayyis, S.A.C. Bukhari, Analysis of dimensionality reduction techniques on internet of things data using machine learning, Sustain. Energy Technol. Assess. 52 (2022) 102304.

[2] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, W. Zhao, A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications, IEEE Internet Things J. 4 (5) (2017) 1125–1142.

[3] B. Tahir, M. Tariq, Vulnerability assessment and federated intrusion detection of air taxi enabled smart cities, Sustain. Energy Technol. Assess. 53 (2022) 102686.

[4] A. Manocha, M. Bhatia, Iot-inspired monitoring framework for real-time stereotypic movement analysis, IEEE Syst. J. (2021).

[5] A. Basati, M.M. Faghih, Dfe: efficient iot network intrusion detection using deep feature extraction, Neural Comput. Appl. (2022) 1–21.

[6] A.A. Malibari, S.S. Alotaibi, R. Alshahrani, S. Dhahbi, R. Alabdan, F.N. Al-wesabi, A.M. Hilal, A novel metaheuristics with deep learning enabled intrusion detection system for secured smart environment, Sustain. Energy Technol. Assess. 52 (2022) 102312.

[7] M.A. Khan, K. Salah, Iot security: Review, blockchain solutions, and open challenges, Future Gener. Comput. Syst. 82 (2018) 395–411, [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X17315765.

[8] D. Swessi, H. Idoudi, A survey on internet-of-things security: Threats and emerging countermeasures, Wirel. Pers. Commun. (2022) 1–36.

[9] P. Gupta, et al., A survey of application layer protocols for internet of things, in: 2021 International Conference on Communication Information and Computing Technology, ICCICT, IEEE, 2021, pp. 1–6.

[10] A.M. González-Zapata, E. Tlelo-Cuautle, I. Cruz-Vega, W.D. León-Salas, Synchronization of chaotic artificial neurons and its application to secure image transmission under mqtt for iot protocol, Nonlinear Dynam. 104 (4) (2021) 4581–4600.

[11] M.O. Al Enany, H.M. Harb, G. Attiya, A comparative analysis of mqtt and iot application protocols, in: 2021 International Conference on Electronic Engineering, ICEEM, IEEE, 2021, pp. 1–6.

[12] M.A. Khan, M.A. Khan, S.U. Jan, J. Ahmad, S.S. Jamal, A.A. Shah, N. Pitropakis, W.J. Buchanan, A deep learning-based intrusion detection system for mqtt enabled iot, Sensors 21 (21) (2021) 7016.

[13] H.C. Hwang, J. Park, J.G. Shon, Design and implementation of a reliable message transmission system based on mqtt protocol in iot, Wirel. Pers. Commun. 91 (4) (2016) 1765–1777.

[14] U. Hunkeler, H.L. Truong, A. Stanford-Clark, Mqtt-s—a publish/subscribe protocol for wireless sensor networks, in: 2008 3rd International Conference on Communication Systems Software and Middleware and Workshops, COMSWARE'08, IEEE, 2008, pp. 791–798.

[15] P.K. Deb, S. Misra, A. Mukherjee, Latency-aware horizontal computation offloading for parallel processing in fog-enabled iot, IEEE Syst. J. (2021).

[16] M. Saharkhizan, A. Azmoodeh, A. Dehghantanha, K.-K.R. Choo, R.M. Parizi, An ensemble of deep recurrent neural networks for detecting iot cyber attacks using network traffic, IEEE Internet Things J. 7 (9) (2020) 8852–8859.

[17] S. Ullah, J. Ahmad, M.A. Khan, E.H. Alkhammash, M. Hadjouni, Y.Y. Ghadi, F. Saeed, N. Pitropakis, A new intrusion detection system for the internet of things via deep convolutional neural network and feature engineering, Sensors 22 (10) (2022) 3607.

[18] Y. Zhang, P. Li, X. Wang, Intrusion detection for iot based on improved genetic algorithm and deep belief network, IEEE Access 7 (2019) 31 711–31 722.

[19] Venafi, 2021, Cyber attacks on iot devices are growing at alarming rates [encryption digest 64].

[20] V. Shakhov, S.U. Jan, S. Ahmed, I. Koo, On lightweight method for intrusions detection in the internet of things, in: 2019 IEEE International Black Sea Conference on Communications and Networking, BlackSeaCom, 2019, pp. 1–5.

[21] M. Driss, D. Hasan, W. Boulila, J. Ahmad, Microservices in iot security: current solutions, research challenges, and future directions, Procedia Comput. Sci. 192 (2021) 2385–2395.

[22] K.A. da Costa, J.P. Papa, C.O. Lisboa, R. Munoz, V.H.C. de Albuquerque, Internet of things: A survey on machine learning-based intrusion detection approaches, Comput. Netw. 151 (2019) 147–157.

[23] N. Moustafa, B. Turnbull, K.-K.R. Choo, An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of internet of things, IEEE Internet Things J. 6 (3) (2018) 4815–4830.

[24] M.S. Alkatheiri, M.A. Alqarni, S.H. Chauhdary, Cyber security framework for smart home energy management systems, Sustain. Energy Technol. Assess. 46 (2021) 101232, [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2213138821002423.

[25] Y. Zhang, C. Yang, K. Huang, Y. Li, Intrusion detection of industrial internet-of-things based on reconstructed graph neural networks, IEEE Trans. Netw. Sci. Eng. (2022).

[26] E.M. Dovom, A. Azmoodeh, A. Dehghantanha, D.E. Newton, R.M. Parizi, H. Karimipour, Fuzzy pattern tree for edge malware detection and categorization in iot, J. Syst. Archit. 97 (2019) 1–7.

[27] M. Ragab, M.F.S. Sabir, Outlier detection with optimal hybrid deep learning enabled intrusion detection system for ubiquitous and smart environment, Sustain. Energy Technol. Assess. 52 (2022) 102311.

[28] A. Aleesa, B. Zaidan, A. Zaidan, N.M. Sahar, Review of intrusion detection systems based on deep learning techniques: coherent taxonomy, challenges, motivations, recommendations, substantial analysis and future directions, Neural Comput. Appl. 32 (14) (2020) 9827–9858.

[29] M.A. Abbas, Z. Iqbal, F.Z. Khan, S. Alsubai, A. Binbusayyis, A. Alqahtani, An ann based bidding strategy for resource allocation in cloud computing using iot double auction algorithm, Sustain. Energy Technol. Assess. 52 (2022) 102358.

[30] S. Ben Atitallah, M. Driss, W. Boulila, H. Ben Ghezala, Randomly initialized convolutional neural network for the recognition of covid-19 using x-ray images, Int. J. Imaging Syst. Technol. 32 (1) (2022) 55–73.

[31] S. Ben Atitallah, M. Driss, W. Boulila, A. Koubaa, H. Ben Ghézala, Fusion of convolutional neural networks based on dempster–shafer theory for automatic pneumonia detection from chest x-ray images, Int. J. Imaging Syst. Technol. 32 (2) (2022) 658–672.

[32] N. Naz, M.A. Khan, S.A. Alsuhibany, M. Diyan, Z. Tan, M.A. Khan, J. Ahmad, Ensemble learning-based ids for sensors telemetry data in iot networks, Math. Biosci. Eng. 19 (10) (2022).

[33] A. Alzahem, W. Boulila, M. Driss, A. Koubaa, I. Almomani, Towards optimizing malware detection: An approach based on generative adversarial networks and transformers, in: Computational Collective Intelligence: 14th International Conference, ICCCI 2022, Hammamet, Tunisia, September (2022) 28–30, Proceedings, Springer, 2022, pp. 598–610.

[34] T.-T.-H. Le, Y.E. Oktian, H. Kim, Xgboost for imbalanced multiclass classification-based industrial internet of things intrusion detection systems, Sustainability 14 (14) (2022) 8707.

[35] D. Grechishnikova, Transformer neural network for protein-specific de novo drug generation as a machine translation problem, Sci. Rep. 11 (1) (2021) 1–13.

[36] Y.-G. Yang, H.-M. Fu, S. Gao, Y.-H. Zhou, W.-M. Shi, Intrusion detection: A model based on the improved vision transformer, Trans. Emerg. Telecommun. Technol. (2022) e4522.

[37] H. Hindy, C. Tachtatzis, R. Atkinson, E. Bayne, X. Bellekens, Mqtt-iot-ids2020: Mqtt internet of things intrusion detection dataset, 2020, [Online]. Available: http://dx.doi.org/10.21227/bhxy-ep04.

[38] S.T. Mehedi, A. Anwar, Z. Rahman, K. Ahmed, I. Rafiqul, Dependable intrusion detection system for iot: A deep transfer learning-based approach, IEEE Trans. Ind. Inform. (2022).

[39] O.A. Wahab, Intrusion detection in the iot under data and concept drifts: Online deep learning approach, IEEE Internet Things J. (2022).

[40] M. Rashid, J. Kamruzzaman, T. Imam, S. Wibowo, S. Gordon, A tree-based stacking ensemble technique with feature selection for network intrusion detection, Appl. Intell. (2022) 1–14.

[41] E. Gyamfi, A.D. Jurcut, Novel online network intrusion detection system for industrial iot based on oi-svdd and as-elm, IEEE Internet Things J. (2022).

[42] J. Liu, D. Yang, M. Lian, M. Li, Research on intrusion detection based on particle swarm optimization in iot, IEEE Access 9 (2021) 38 254–38 268.

[43] D.C. Attota, V. Mothukuri, R.M. Parizi, S. Pouriyeh, An ensemble multi-view federated learning intrusion detection for iot, IEEE Access 9 (2021) 117 734–117 745.

[44] F. Mosaiyebzadeh, L.G.A. Rodriguez, D.M. Batista, R. Hirata, A network intrusion detection system using deep learning against mqtt attacks in iot, in: 2021 IEEE Latin-American Conference on Communications, LATINCOM, IEEE, 2021, pp. 1–6.

[45] H. Hindy, E. Bayne, M. Bures, R. Atkinson, C. Tachtatzis, X. Bellekens, Machine learning based iot intrusion detection system: an mqtt case study (mqtt-iot-ids2020 dataset), in: International Networking Conference, Springer, 2020, pp. 73–84.

[46] Z. Che, S. Purushotham, K. Cho, D. Sontag, Y. Liu, Recurrent neural networks for multivariate time series with missing values, Sci. Rep. 8 (1) (2018) 6085.

[47] B.J. Wells, K.M. Chagin, A.S. Nowacki, M.W. Kattan, Strategies for handling missing data in electronic health record derived data, Egems 1 (3) (2013).

[48] Z. Cui, R. Ke, Z. Pu, Y. Wang, Stacked bidirectional and unidirectional lstm recurrent neural network for forecasting network-wide traffic state with missing values, Transp. Res. C 118 (2020) 102674.

[49] M.F. Uddin, J. Lee, S. Rizvi, S. Hamada, Proposing enhanced feature engineering and a selection model for machine learning processes, Appl. Sci. 8 (4) (2018) 646.

[50] A. Globerson, N. Tishby, Sufficient dimensionality reduction, J. Mach. Learn. Res. 3 (Mar) (2003) 1307–1331.

[51] D. Singh, B. Singh, Investigating the impact of data normalization on classification performance, Appl. Soft Comput. 97 (2020) 105524.

[52] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Adv. Neural Inf. Process. Syst. 30 (2017) 00.

[53] R. Kozik, M. Pawlicki, M. Choraś, A new method of hybrid time window embedding with transformer-based traffic data classification in iot-networked environment, Pattern Anal. Appl. 24 (4) (2021) 1441–1449.

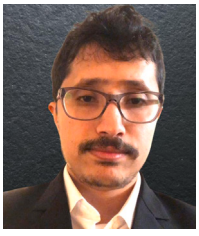
[54] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[55] S. Vani, T.M. Rao, An experimental approach towards the performance assessment of various optimizers on convolutional neural network, in: 2019 3rd International Conference on Trends in Electronics and Informatics, ICOEI, IEEE, 2019, pp. 331–336.

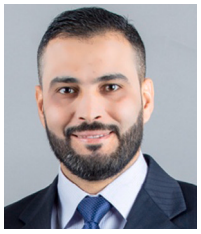
Safi Ullah received the M.Phil. degree in Computer Science from Quaid-i-Azam University, Islamabad, Pakistan, in 2022. He is currently pursuing the Ph.D. degree with the Computer Science department, Quaid-i-Azam University, Islamabad, Pakistan. He is currently working in the research area of cybersecurity of the Internet of Vehicles (IoV) and the Internet of Things (IoT).

Jawad Ahmad (Senior Member, IEEE) is an experienced researcher with more than ten years of cutting-edge research and teaching experience in prestigious institutes, including Edinburgh Napier University (U.K.), Glasgow Caledonian University (U.K.), Hongik University (South Korea), and HITEC University Taxila (Pakistan). He has coauthored more than 100 research papers, in international journals and peer-reviewed international conference proceedings. He has taught various courses both at Undergraduate (UG) and Postgraduate (PG) levels during his career. He regularly organizes timely special sessions and workshops for several flagship IEEE conferences. He is an invited reviewer for numerous world-leading high-impact journals (reviewed more than 100 journal papers to date). His research interests include cybersecurity, multimedia encryption, and machine learning.

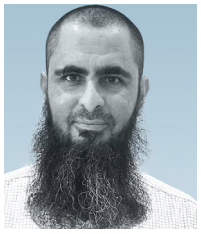
Prof. Muazzam Ali Khan Khattak (Senior Member, IEEE and Pakistan Academy of Sciences) is working as Tenured Professor and Director ICESCO Chair for Data Analytics and Edge Computing, Quaid-i-Azam University, Islamabad Pakistan. He received his Ph.D. and Postdoc from IIUI and University of Missouri, KC, USA in 2011 and 2016 respectively. Dr. Khattak joined the National University of Sciences & Technology (NUST), Islamabad in 2013 and was promoted to Postgraduate Program Head & Associate Dean in 2013 and 2017 respectively. He has been at the School of Computer Science, University of Ulm, Germany, and at the Networking and Multimedia Lab, School of Computer and Electrical Engineering, University of Missouri (UMKC), USA as a research fellow. He is also adjunct Professor at University of Missouri, KC, USA since 2016. His research interests include the Smart cities, IoTs, Next Generation Intelligent Networks, Blockchain, Information and Network Security, Ad-hoc and Acoustic Networks. He has published more than 170 publications and book chapters with total impact factor of 370+ and 2500 citations.Based on his performance he has been awarded best researcher award in 2016 and 2021.



Mohammed S. Alshehri received the B.S. degree in Computer Science from the King Khalid University, Abha, KSA, in 2010, and received M.S degree in Computer Science from the University of Colorado, Denver, USA, in 2014. Mohammed received the Ph.D. degree in Computer Science from the University of Arkansas, Fayetteville, USA, in 2021. Mohammed also received a graduate certificate in Cybersecurity from the University of Arkansas, Fayetteville, USA, in 2021. Mohammed joined, moderator volunteer, IEEE CS DVP-SYP Virtual Conference in 2021. Mohammed has gained multiple professional certificates during his graduate life, such as: Security+, Network+, and CISM. Mohammed's areas of interest contains Cyber Security, Computer Networks, Cloud-Fog-Edge Computing, IoT, Blockchain, Machine Learning, and Deep Learning. Mohammed is currently joining Najran University, Najran, KSA as assistant professor in the department of Computer Science.



Wadii Boulila received the B.Eng. degree (1st Class Honours with distinction) in computer science from the Aviation School of Borj El Amri, in 2005, the MSc. degree in computer science from the National School of Computer Science (ENSI), University of Manouba, Tunisia, in 2007, and the Ph.D. degree in computer science conjointly from the ENSI and Telecom-Bretagne, University of Rennes 1, France, in 2012. He is currently an associate professor of computer science with Prince Sultan University, Saudi Arabia. He is also a senior researcher with the RIOTU Laboratory, Prince Sultan University, Saudi Arabia, a senior researcher with RIADI Laboratory, University of Manouba, and previously a Senior Research Fellow with the ITI Department, University of Rennes 1, France. Wadii received the award of the young researcher in computer science in Tunisia for the year 2021 from Beit El-Hikma and the award of best researcher from the University of Manouba in Tunisia for the year 2021. He participated in many research and industrial-funded projects. His primary research interests include data science, big data analytics, deep learning, cybersecurity, artificial intelligence, uncertainty modeling, and image analysis and interpretation. He has served as the chair, a reviewer, and a TPC member for many leading international conferences and journals. He is an IEEE Senior member, an ACM member and a Senior Fellow of the Higher Education Academy (SFHEA), U.K.



Anis Koubâa is currently a Professor in computer science and the Leader of the Robotics and Internet of Things Research Laboratory, Prince Sultan University. He is also a Senior Researcher with CISTER and ISEP-IPP, Porto, Portugal, and a Research and Development Consultant with Gaitech Robotics, China. His current research interests include providing solutions toward the integration of robots and drones into the Internet of Things (IoT) and clouds, in the context of cloud robotics, robot operating systems (ROSs), robotic software engineering, wireless communication for the IoT, real-time communication, safety and security for cloud robotics, intelligent algorithm's design for mobile robots, and multi-robot task allocation. He is also a Senior Fellow of the Higher Education Academy (HEA), U.K. He has been the Chair of the ACM Chapter in Saudi Arabia, since 2014.

Sana Ullah Jan is an experienced researcher with more than 8 years of cutting-edge research and teaching experience in prestigious institutes including the University of the West of Scotland, the University of Ulsan (South Korea) and the University of Lahore (Pakistan). He is currently enrolled as Lecturer/Assistant Professor in Edinburgh Napier University, UK since September 2021. He was previously enrolled as Post-doctoral Research Fellow at the Center of Affective and Human Computing for Smart Environment at the school of computing, engineering and physical sciences, University of the West of Scotland since September 2020 to August 2021. He has (co)authored more than 20 papers in international journals and peer-reviewed international conference proceedings. His research area is closely related to the Artificial Intelligence or Machine Learning-based cyber security and privacy in the Internet-of-Things, Cyber Physical Systems and eHealth. He has taught various courses both at Undergraduate (UG) and Postgraduate (PG) levels during his career. He is invited reviewer for several leading high-impact journals and conferences. He has been endorsed as Global Talent by the Royal Academy of Engineering of the UK.

M. Munawwar Iqbal Ch is currently working as assistant professor in Institute of Information Technology, Quaid-i-Azam University, Islamabad, Pakistan. He received his Ph.D. degree in Computer Software Engineering from National University of Sciences and Technology (NUST), Islamabad, Pakistan in 2020. He is actively involved in information systems, image processing, waves mechanics and machine learning related research activities.