Computers, Materials & Continua
DOI:10.32604/cmc.2022.020727
Article

Tech Science Press

An Efficient Internet Traffic Classification System Using Deep Learning for IoT

Muhammad Basit Umair[1], Zeshan Iqbal[1], Muhammad Bilal[2], Jamel Nebhen[4] Tarik Adnan Almohamad[3] and Raja Majid Mehmood[5,*]

[1]Department of Computer Science, University of Engineering and Technology, Taxila, 47050, Pakistan
[2]Division of Computer and Electronics Systems Engineering, Hankuk University of Foreign Studies, Yongin-si, Korea
[3]Electrical-Electronics Engineering Department, Faculty of Engineering, Karabük University, 78050, Karabük, Turkey
[4]Prince Sattam bin Abdulaziz University, College of Computer Engineering and Sciences, Alkharj, 11942, Saudi Arabia
[5]Information and Communication Technology Department, School of Electrical and Computer Engineering, Xiamen University Malaysia, Sepang, 43900, Malaysia
[*]Corresponding Author: Raja Majid Mehmood. Email: rmeex07@ieee.org
Received: 04 June 2021; Accepted: 05 July 2021

Abstract: Internet of Things (IoT) defines a network of devices connected to the internet and sharing a massive amount of data between each other and a central location. These IoT devices are connected to a network therefore prone to attacks. Various management tasks and network operations such as security, intrusion detection, Quality-of-Service provisioning, performance monitoring, resource provisioning, and traffic engineering require traffic classification. Due to the ineffectiveness of traditional classification schemes, such as port-based and payload-based methods, researchers proposed machine learning-based traffic classification systems based on shallow neural networks. Furthermore, machine learning-based models incline to misclassify internet traffic due to improper feature selection. In this research, an efficient multilayer deep learning based classification system is presented to overcome these challenges that can classify internet traffic. To examine the performance of the proposed technique, Moore-dataset is used for training the classifier. The proposed scheme takes the pre-processed data and extracts the flow features using a deep neural network (DNN). In particular, the maximum entropy classifier is used to classify the internet traffic. The experimental results show that the proposed hybrid deep learning algorithm is effective and achieved high accuracy for internet traffic classification, i.e., 99.23%. Furthermore, the proposed algorithm achieved the highest accuracy compared to the support vector machine (SVM) based classification technique and k-nearest neighbours (KNNs) based classification technique.

Keywords: Deep learning; internet traffic classification; network traffic management; QoS aware application classification

1 Introduction

Nowadays, Internet of things (IoT) devices are connected, embedded, and producing a substantial amount of data [1,2]. Hence, causing various network management issues for both academia and industry and opens new research areas and directions [3,4] Internet traffic classification (ITC) plays an important role in improving network performance, intrusion detection, QoS provisioning, and dynamic access control. Therefore, various ITC techniques have been proposed in recent years: the port-based classification method, payload-based classification method, and machine-learning-based classification system [5].

In the port-based classification method, the port numbers are allocated by the Internet Assigned Numbers Authority (IANA) [6]. Earlier, fixed port numbers were used by most internet protocols. Due to security reasons, the majority of the protocols are using a dynamic port allocation system nowadays [7]. Hence, the dynamic port allocation limits the use of port-based traffic classification. Therefore, to overcome these restrictions, the payload-based technique was introduced. In this method, the application-layer information of packets is examined, and classification is performed by inspecting the packet payload and comparing the signature of the protocols stored in the database [8]. However, this technique is challenging to implement. For example, in the presence of encrypted traffic, the DPI cannot examine the packet payload. Moreover, from the users' perspective, DPI can cause issues related to privacy.

Machine learning-based traffic classification was proposed to overcome the challenges of payload-based and port-based methods [9–15]. The machine learning algorithms can be classified into three categories: supervised, unsupervised, and semi-supervised. In supervised learning, a labelled dataset is used for training. In semi-supervised, both labelled and unlabeled data is used. In unsupervised learning, unlabeled data is used for classification [16], and classes are unknown. A number of machine learning algorithms such as SVM, Backpropagation, KNNs, and C4.5 [5] have been used for traffic classification [17]. These algorithms use the port number, inter-arrival time, and packet size for the classification [18]. However, there are a few challenges while using machine learning for ITC: the rapid development of new applications and new training models are required for newly emerging topologies.

In this research, A multilayer deep neural network (DNN) with a maximum entropy classifier is proposed for ITC. First, Moore's dataset is acquired and pre-processed using the OneHotEncoding technique to encode categorical variables. The pre-processed data is split into training and testing data. Next, the common flow features are identified and selected using an extra-trees classifier from pre-processed data. Finally, a feedforward DNN is trained, and the maximum entropy classifier is used at the output layer of the DNN. The proposed multilayer DNN has an accuracy of 99.23%. Furthermore, for comparison purposes, a series of experiments are conducted using SVM and KNN algorithms. The SVM and KNN algorithms achieved an accuracy of 98.90% and 98.56%, respectively. This comparison shows that multilayer DNN can be used to accurately classify network traffic as compared to shallow networks.

In summary, the major contributions of this research are as follows.

(1) An efficient hybrid deep learning model is designed using a multilayer feedforward DNN, and a maximum entropy classifier for internet traffic classification. Moore's dataset is acquired and pre-processed using the OneHotEncoding technique to encode categorical variables. The pre-processed data is split into training and testing data. Features are extracted through a multilayer feedforward DNN, and a maximum entropy classifier is used to classify the internet traffic.

(2) The functionalities of multilayer DNN are improved using the dropout layer. The dropout layer provides the functionality to avoid overfitting.
(3) In order to eliminate the stochastic gradient descent (SGD), slow convergence, vanishing learning rate, and high variance in the parameter updates, which causes loss function to fluctuate, Adam optimizer is used for training the network.
(4) A comparison of the proposed hybrid deep learning model is presented with state-of-the-art models on a benchmark dataset for internet traffic classification. Furthermore, for comparison purposes, a number of experiments are conducted using SVM and KNN algorithms.

The remainder of this paper is organized as follows. First, related works are presented in Section 2. Section 3 describes the proposed methodology for ITC. In Section 4, we summarize an experimental setup. Next, the experimental results are given in Section 5.Finally, the conclusion of the research and future work is discussed in Section 6.

2 Related Work

In the last decade, numerous research have been conducted for ITC, such as port-based, payload-based, and machine-learning-based approaches. Recently, machine-learning-based techniques have gained more interest due to their high performance in ITC [5]. Many algorithms are applied to the network traffic classification, but these algorithms are very different in nature. Most of the researchers focused on feature and machine learning algorithm selection. Both the feature and algorithm selection have great importance in improving the classifier performance. A summary of the relevant and state of the art classification methods, the techniques used, their dataset usage, accuracy, and associated algorithms are presented in Tab. 1.

Zhang et al. [19] have proposed a technique for application classification based on deep learning and software-defined networks (SDN) architecture. A hybrid deep learning model for network application classification was proposed by merging the stacked auto-encoder and the softmax regression. A stacked auto-encoder was used for flow features extraction, and the softmax regression layer was used for classifying the network application. The author achieved an average accuracy of 91.21% using five hidden layers and ten hidden nodes.

Lopez-Martin et al. [20] suggested a deep learning model which combines a convolutional neural network (CNN) and long short-term memory (LSTM) to identify the network traffic. For every flow, different features were extracted from the packet's header, and a time series feature vector was built. More than 25000 features and 100 services were used to train a model. On the best trained model, this statistical approach achieved 0.9632 and 0.9574 of accuracy and F1-score, respectively. Sun et al. [21] introduced a TrAdaBoost system that utilized the labeled traffic data collected from different sources for the classification of network traffic. A base classifier, the maximum entropy model (Maxent), was used to implement a source of knowledge from source data to target data. An accuracy of 98.7% was obtained using the TrAdaBoost model.

To identify suspicious flows in SDN based networks, Garg et al. [22] presented a anomaly detection technique using deep learning for social media domains. Two refined algorithms were used to satisfy the QoS requirements of the SDN. However, the dataset used in this research is not flow-based SDN. A kernel-based Extreme Learning Machine (ELM) approach [23] was applied to Moore's dataset to classify the internet traffic. To select the best features, a genetic algorithm was used. There were 12 attributes in the flow features. They obtained 96.25% accuracy using the wavelet activation function.

Table 1: Brief comparison of existing techniques

Author	Technique used	Year of publication	Dataset	Classification accuracy	Summary
Shi et al. [9]	Deep learning	2018	Moore and UNIBS	98%	To remove the irrelevant features, the symmetric uncertainty was applied.
Yu et al. [10]	Semi-supervised learning	2018	Campus network	Not mentioned	A DPI based method was used to label the network traffic.
Zhang et al. [19]	Deep learning	2018	Moore-dataset	91.21%	A DNN was designed using autoencoder and softmax model for ITC.
Lopez-Martin et al. [20]	RNN and CNN	2017	RedIRIS dataset	0.96	A deep learning-based model by combining CNN and RNN was employed.
Sun et al. [21]	Transfer learning	2018	Moore-dataset	98.7%	A TrAdaBoost was to label the traffic, and maxnet was used as the base classifier.
Garg et al. [22]	Deep learning	2019	KDD99 University dataset	Not mentioned	A deep learning method to detect the abnormal activities in SDN.
Ertam et al. [23]	Extreme learning machine	2016	Moore-dataset	96.25%	To classify the network traffic using extreme learning machine.
Dias et al. [24]	Machine-learning	2019	Campus network	98.88%	A real-time video classification scheme was introduced.

(Continued)

Table 1: Continued

Author	Technique used	Year of publication	Dataset	Classification accuracy	Summary
Gómezv et al. [25]	Machine-learning	2019	Barcelona Tech net-work traffic	—	To deal with the class imbalance problem, a new approach was adopted.
Lotfollahi et al. [26]	CNN and encoder	2020	ISCX VPN-nonVPN	—	A novel approach has been provided using CNN and Stacked auto-encoder.
Cao et al. [27]	SVM	2020	Moore dataset	91.96%	A wrapper-based hybrid feature selection method to select important features.

A video classification method [24], based on the Naïve Bayes classifiers, was implemented for real-time network traffic classification. There were three classes in the proposed classification scheme: one file download service and two video services. The author stated that the accuracy of 98.88% was achieved in real-time scenarios. This method cannot be implemented in the case of encrypted traffic, and it is computationally expensive. To handle the class imbalance problems, Gómezv et al. [25] used a base estimator to form a baseline.

Lotfollahi et al. [26] introduced a method named "deep packet" using one-dimensional convolutional neural network (1-D CNN) and stacked auto-encoder. This model was trained to classify the encrypted network traffic. This model achieved 98% and 94% of recall for application identification and traffic categorization, respectively. Cao et al. [27] proposed a network traffic classification technique based on SVM. To prevent overfitting, a wrapper-based feature selection algorithm is used to select important features. The authors have achieved good classification accuracy of 91.96% for multi-class classification, which was 4.2% higher than the original SVM. Kordestani et al. [28] surveyed a list of methods used for fault diagnosis and prognosis. For encrypted traffic classification, Aceto et al. [29] designed a method using a deep learning algorithm. In this method, first, they discussed deep learning architecture for encrypted traffic classification. They used 1-D CNN, 2-D CNN, LSTM, stacked autoencoder (SAE), and multilayer perceptron algorithms that can identify up to 49 mobile apps.

In Aceto et al. [30], proposed a novel, sophisticated multimodal deep learning framework for multi-class mobile traffic classification based on network packets. They take two different time series and payload-based features. In the multimodal deep learning architectures, the obtained accuracy 79.6% for FB/FBM class, and 89.49%, 89.14% for android and IOS, respectively. For feature engineering, Swarna et al. [31] used a hybrid technique using Principal Component Analysis (PCA) and Grey Wolf Optimization (GWO) for intrusion detection in Internet of Medical

Things and achieved good results. An Efficient Feature Optimization Approach (EFOA) [9] was proposed for optimal feature selection and optimization. A series of experiments were performed on the University of Cambridge dataset [32]. First, the correlation of original flow statistics was evaluated, and irrelevant features were removed using symmetric uncertainty (SU). Next, to get the robust features, related features were passed to the feature generation model. The feature generation model was based on a deep belief network, implemented using unsupervised learning. Finally, redundant features were removed using weighted symmetric uncertainty (WSU). However, the proposed model is computationally expensive.

On the other hand, in the semi-supervised based classification methods, Yu et al. [10] combined the DPI and semi-supervised learning for multi-classifier in SDN. The combined arrangement was used to classify applications into different categories. A DPI-based technique was used to maintain a traffic database, and a partially labeled dataset was formed and stored. However, this method cannot classify traffic if the data is encrypted. Reddy et al. [33] discussed two methods PCA and GWO, for dimensionality reduction using different machine learning algorithms.

Although numerous research efforts have been invested for ITC, there are still prominent issues in the existing literature that need to be addressed, including incapability of classifying encrypted traffic, less accuracy, and computationally expensive algorithms. The other problem with the aforementioned techniques is that the detailed number of classes is missing. An efficient method is proposed to overcome these issues that consist of a deep neural network and a softmax classifier. Compared with other networks, the proposed method not only efficiently classifies the internet traffic but also it is computationally less expensive. The performance and obtained results of the proposed system are compared with recent and state-of-the-art ITC schemes. Furthermore, the proposed study provides a complete roadmap for ITC.

3 Proposed Methodology

An abstract view of the methodology used throughout this work is shown in Fig. 1. The proposed methodology consisting of four key steps: data acquisition, pre-processing, feature engineering, and classification. The acquired data is processed using LabelEncoder and OneHotEncoder to make the classification of traffic useful. Thus, the proposed scheme is aware of classifying internet traffic into different classes. In the case of real-time traffic, the proposed technique constantly acquires data flows from the internet in the form of flow statistics.

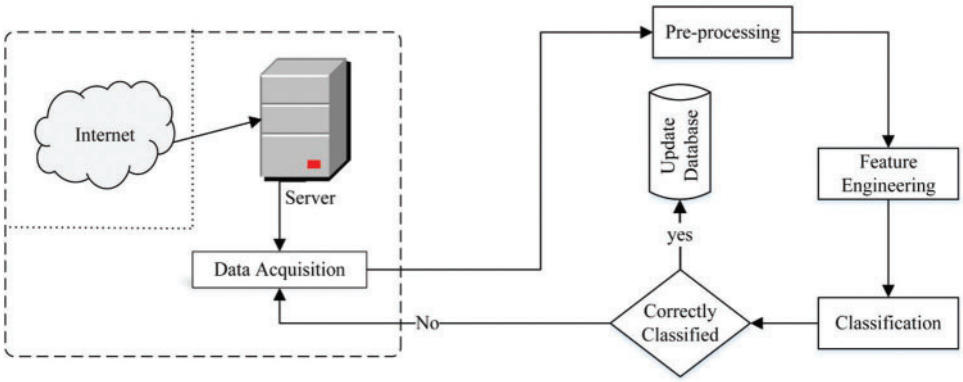

Figure 1: Overview of the proposed system methodology

3.1 Data Acquisition (Dataset)

This work employed the Moore-dataset, which contains a real-world traffic traces dataset with 12 continuous features. These traffic traces have been collected from the University of Cambridge, publicly available for research purposes [32]. The Cambridge University traffic traces are released by the computer lab on Genome Campus. The major advantage of these traffic traces is that it provides a base to get more accurate results and is extensively acceptable for assessment and comparison of traffic classification approaches. Therefore, it can be used as a benchmark to assess the performance of the ITC algorithm. These Cambridge traces are being used by most researchers for ITC research [19,21,34–36].

There are 324, 277 data flows samples in the dataset. The imperative advantage of selected Moore-dataset is different flows for different classes. The acquired dataset consists of TCP traffic flows and hand-verified class labels (ground truth). Li et al. [34] used these traffic traces for research purposes. All the flows are classified into seven application classes, i.e., Mail, P2P, WWW, Chat, Bulk, Database, and Interactive, and other classes are excluded due to the minority number of samples in the dataset.

3.2 Data Pre-Processing

Afterwards, acquisition of the dataset, pre-processing of the data is an essential step. In supervised machine learning, it plays a key role in improving the performance of classifiers. Data pre-processing helps in reducing the computational cost of any system. In the Moore-dataset [32], LabelEncoder converts categorical values into numeric values using the scikit-learn library. In this proposed methodology, there are seven distinct classes, and LabelEncoder encodes the labels with unique numeric values among 0 and (n − 1), where n represents the number of distinct classes. OneHotEncoder takes each column in categorical data, which are label encoded and converted into binary columns. These columns are replaced by 0s and 1s, for each category correspondence to which column has been replaced. Min-max normalization is performed to scale the dataset values between the range 0 and 1 by the given Eq. (1).

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{1}$$

where x' represents the normalized value, $x_{\max}$ is the maximum value, and $x_{\min}$ is the minimum value in the dataset.

3.3 Feature Engineering

Machine learning models may face many problems due to sparse features. This feature selection technique is used to overcome the sparse features and select the most essential features to improve the performance of machine learning models. For the identification of important features from the pre-processed dataset, the extra-trees classifier is used. The feature importance function calculates the information gain of all features and returns the high information gain features. This feature extraction system shows better performance as compared to previous approaches related to ITC [21,23]. To make the proposed methodology computationally efficient, flow-based features are selected. Feature sets extracted for network traffic classification are shown in Tab. 2. There are 12 features in Feature-set F and correspondence class.

$$F = \{f1, f2, f3, \ldots, f12\} \tag{2}$$

where $f1$ represents the server port number and $f2$ represents the client port number, and detail of other features is given in Tab. 2. Finally, all classes and their correspondence applications are given in Tab. 3.

Table 2: The information of given extracted features

Feature	Description
f1	Server-port
f2	Client-port
f3	Actual-data-packets (c to s)
f4	Pushed-data-packets (c to s)
f5	Pushed-data-packets (s to c)
f6	Minimum-segment-size (c to s)
f7	Average-segment-size (c to s)
f8	Initial-window-bytes (c to s)
f9	Initial-window-bytes (s to c)
f10	RTT-samples (c to s)
f11	Median-data-packets (c to s)
f12	Variance-bytes-packet (s to c)
CLASS	WWW, P2P, MAIL, INTERACTIVE, BULK, SERVICES, DATABASE

Table 3: List of classes and correspondence applications

Class	Applications	Percentage (%)
WWW	www	84.077
P2P	BitTorrent, GnuTella	8.571
BULK	ftp-control, ftp-pasv, ftp-attack	2.058
INTERACTIVE	telnet	0.124
MAIL	IMAP, pop2, SMTP	1.530
DATABASE	Sqlnet, oracle, ingress	3.531
CHAT	Yahoo IM, Jabber, MSN Messenger	0.025

3.4 Classification

In this work, a hybrid deep learning system that consists of a DNN and maximum entropy classifier, also known as the softmax regression model, is employed and trained to classify network traffic. A DNN is a feedforward artificial neural network with backpropagation using multiple hidden layers. A DNN comprises an input layer, multiple hidden layers, and an output layer. An overview of the proposed DNN is depicted in Fig. 2. The input layer consists of twelve nodes. The DNN is given by Eq. (3), where x_i is the input, w_i is the weight, and w_0 is the bias of any neuron. The applied activation function is given in Eq. (4), where y_{out} is the observed output. The objective of the implemented DNN is to reduce the error between the input and output layer. The loss function can be calculated by the Eq. (5). A nonlinear activation rectified linear unit

(ReLU) is applied on the input layer. The ReLU is given by Eq. (6). ReLU helps in performance-boosting and accelerating the training method. The main advantage of using ReLU is the removal of exploding gradient problem.

$$y_{net} = w_i x_i + w_0 \tag{3}$$

$$y_{out} = f\,(y_{net}) = \frac{1}{1 + e^{-y_{net}}} \tag{4}$$

$$L = \frac{1}{2}\sum_{i=1}^{N} ||y_{net} - y_{out}||^2 \tag{5}$$

$$\mathrm{Re}LU\,(x) = \max\,(0, z) \tag{6}$$

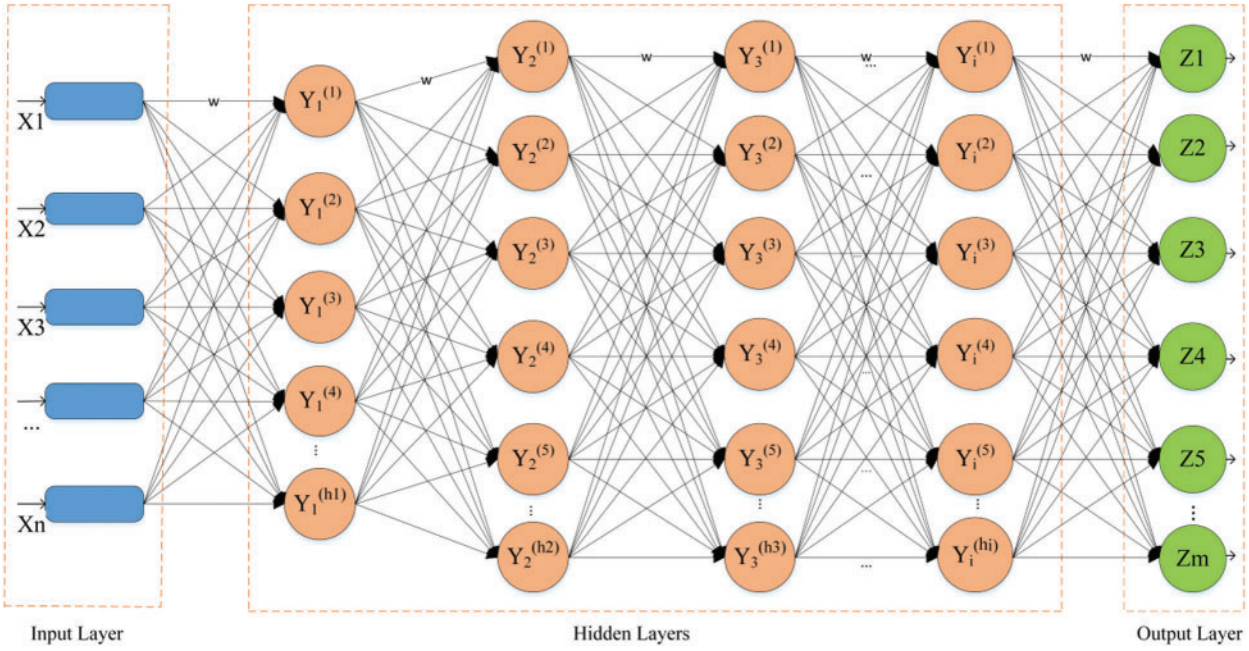

Figure 2: Proposed deep learning model

There are several hidden layers, each consisting of sixteen neurons. Each hidden layer is connected with a nonlinear sigmoid activation function, given by Eq. (7). The ITC problem is a multi-class classification problem, so a maximum entropy classifier is used at the output layer. The maximum entropy classifier belongs to supervised learning, and is usually used with other classifiers. Mathematically, the maximum entropy classifier is given by Eq. (8).

$$sigmoid\,(x) = \frac{1}{1 + e^{z}} \tag{7}$$

$$\sigma\,(u)_j = \frac{e^{u_j}}{\sum_{m=1}^{M} e^{u_m}} \tag{8}$$

where u is the M-dimensional vector, u_m and u_j are the elements of the M-dimensional vector. The $\sigma(u)$ represents M-dimensional after mapping m and j. The j, m are subscripts and values of $j, m = 1, 2, 3 \ldots M$.

4 Parameters Optimization

The performance of a deep learning model varies by the selection of hyperparameters. The grid search parameter technique is used to find out the best deep learning model in this method. To find out the best hyperparameters, the possible combination of every outcome is evaluated. The list of hyperparameters used in this study is given in Tab. 4. In this method, we verify the performance of the proposed method by performing k-fold cross-validation and the train-test split, to find out the optimal hyperparameters. In this proposed method, the batch size is 1000.

Table 4: Hyperparameters used for the training of DNN

Parameter	Value
Activation function	ReLU, sigmoid, softmax
Loss function	Categorical cross entropy
Optimizer	Adam
Learning rate	0.01
Batch size	1000
Cross-validation	10
Number of epochs	500
Dropout rate	0.2

The categorical cross-entropy is used as a loss function. Cross-entropy achieved the best performance in multi-class classification. Moreover, the cross-entropy function has faster convergence and low complexity during the iterative optimization process. The simulation environment of the proposed methodology consists of 500 epochs. To avoid overfitting, we used dropout rate (0.2) after every two hidden layers. The dropout layer removes neurons randomly. In order to eliminate the stochastic gradient descent (SGD), vanishing learning rate, slow convergence, and high variance in the parameter updates, which causes loss function to fluctuate, Adam optimizer is used. The learning rate of 0.01 was chosen that performs best in the case of both training and testing sets. Adam is computationally efficient and an adaptive learning rate optimization algorithm [37]. Further, we used a batch normalization layer that speeds up the deep neural network training and thus relieved the gradient dissipation.

5 Results and Discussion

The simulation experiments are performed using an Intel Core i5 CPU @ 1.80 GHz, 64-bit OS, ×64-based processor on Windows 10 platform. We used Python (version 3.6.5) as a programming language, with scikit-learn library, to perform the experiments [38]. The scikit-learn is an open-source machine learning library. Scikit-learn module is used with NumPy for scientific operations [39]. The dataset is divided as a 7:3 train-test split. The division of the dataset into 70% of the training set and 30% of the test set is randomized.

5.1 Evaluation Metrics

There are four evaluation metrics in the proposed methodology. The explanation of each metric is given in Tab. 5.

Table 5: The evaluation metrics in our proposed methodology

Metric	Definition	Mathematical expression
Accuracy	A number of correctly classified packets in the overall dataset	$Acc = \frac{TP+TN}{TP+FN+FP+FN}$
Precision	The ratio of true positive to entirely positive results	$Percision = \frac{TP}{TP+FP}$
Recall	True positive rate	$Recall = \frac{TP}{TP+FN}$
F1-score	The measure of the test accuracy	$F1 - score = 2 * \frac{Percision * Recall}{Percision + Recall}$

5.2 Experimental Results

The experiments are performed to assess the performance of the DNN. From the data mining perspective, ITC is a multi-class classification problem. To find the best deep learning model, it is necessary to identify the number of hidden layers in a DNN. The DNN deals with a high number of hidden layers, and it has a stronger classification capability to achieve higher classification results. Also, the learning rate plays an important role in maintaining convergence. The experiments are performed to determine the number of hidden layers of the network.

Tab. 6 presents the results of the relationship between the number of hidden layers and DNN performance for all four evaluation metrics. It is observed from Tab. 6 that the highest accuracy of 99.23% is achieved by using the seven hidden layers. However, by increasing the number of hidden layers, the performance of the network doesn't improve; therefore, seven hidden layers are chosen for DNN used in this study.

Table 6: The performance of the proposed deep learning framework

No	Hidden layers	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
1	3	94.66	92.49	94.66	93.01
2	4	99.08	99.00	99.08	99.03
3	5	99.13	98.90	99.13	99.02
4	6	99.11	98.89	99.11	99.00
5	7	99.23	99.15	99.23	99.18
6	8	99.15	99.08	99.15	99.10
7	9	99.13	99.11	99.13	99.12
8	10	99.10	99.11	99.10	99.10

In Fig. 3, the computational cost for the proposed algorithm is depicted. It is evident from the graph that by increasing the number of hidden layers, the proposed algorithm is computationally less expensive as compared to EFOA.

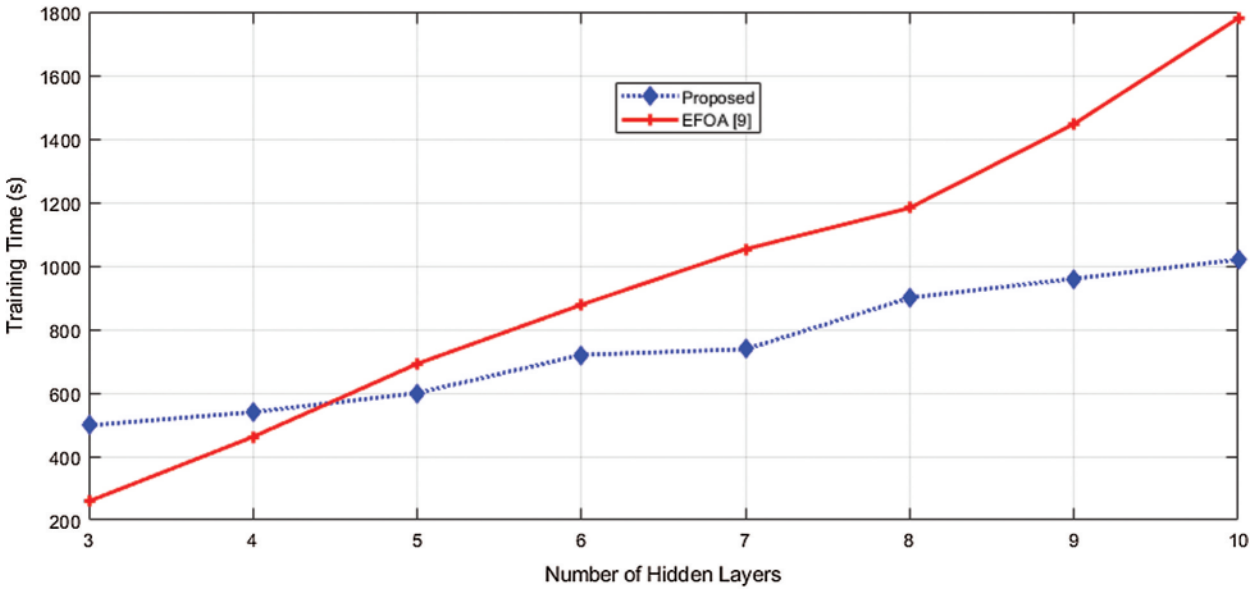

Figure 3: Comparison of computational time with state-of-the-art

Tab. 7 shows the results achieved using the proposed DNN for each class. Four evaluation parameters are calculated to verify the performance of the network. The detail of each class is given in Tab. 7, and corresponding parameters are given in Tab. 5. For Database, WWW, and P2P classes, the accuracy, precision, recall, and F1-score are high because there is a greater number of samples in the dataset. On the other hand, each metric has a low score for the Chat class due to the lowest number of samples. The graphical representation of accuracy, precision, recall, and F1-score for each class is depicted in Fig. 4.

Table 7: Results achieved using DNN for each class

Class	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
WWW	99.00	100.0	99.00	100.0
P2P	99.00	100.0	99.00	99.00
MAIL	84.00	85.00	84.00	84.00
INTERACTIVE	89.00	57.00	89.00	69.00
DATABASE	100.0	100.0	100.0	100.0
CHAT	69.00	50.00	69.00	56.00
BULK	92.00	84.00	92.00	88.00

5.3 Performance Evaluation

In k fold cross-validation, the dataset is divided into k-disjoint equal subset in which $k - 1$ samples are used for training, and the remaining samples are used as a testing set.

Tab. 8 presents the accuracy, precision, recall, and F1-score for different classes using $k = 10$ fold and the sample size = 2000. Finally, Tab. 9 summarizes the average accuracy, precision, recall, and F1-score for each class. It is evident from Tab. 9 that the performance of the proposed DNN

is exceptional for each evaluation metric accuracy (99.15%), precision (99.29%), recall (99.15%), and F1-score (99.12%) using k-folds cross-validation for the classification of network traffic.

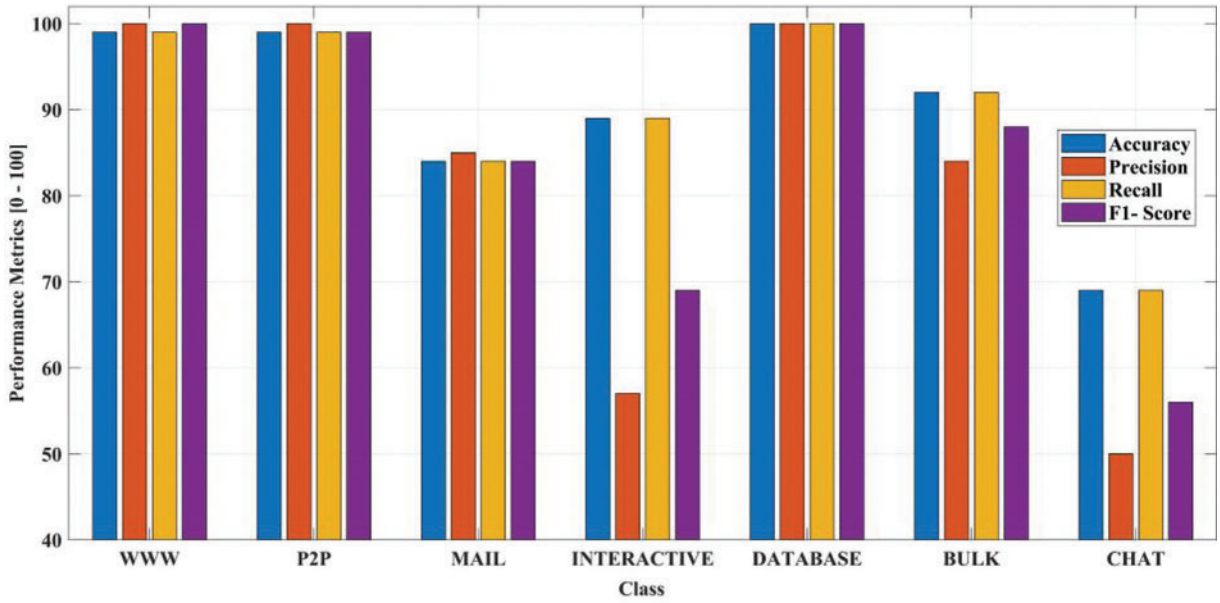

Figure 4: Performance metrics on different applications using DNN

Table 8: Results achieved using DNN for each class using k = 10 folds

Class	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
BULK	68.00	99.00	68.00	81.00
DATABASE	100.0	100.0	100.0	100.0
INTERACTIVE	100.0	98.00	100.0	99.00
MAIL	99.00	59.00	99.00	74.00
P2P	100.0	100.0	100.0	100.0
WWW	100.0	100.0	100.0	100.0

Table 9: Results achieved using DNN for using k = 10 folds

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
DNN	99.15	99.29	99.15	99.12

Further experiments are conducted to evaluate the performance of the proposed DNN with two shallow networks: SVM and KNN. Tab. 10 summarizes the comparison results. For each evaluation metric, the proposed network achieved the highest accuracy (99.23%) in comparison to SVM (98.56%) and KNN (98.90%).

Table 10: Accuracy using DNN, KNN, and SVM

Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
SVM	98.56	98.06	98.56	98.25
KNN	98.90	98.41	98.90	98.45
DNN	99.23	99.15	99.23	99.18

Tab. 11 compares the results of the proposed traffic classification approach with the different state-of-the-art research studies [9,19,21,23,27,36,40,41] for ITC. It can be observed that using DNN architecture, the highest accuracy of 99.23% is achieved using the Moore dataset due to its strong classification capability.

Table 11: Performance comparison of proposed traffic classification scheme results with state-of-the-art

Study	Year of publication	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
[9]	2018	98.0	–	–	–
[19]	2018	91.21			90
[21]	2018	98.7	–	–	–
[23]	2017	96.2	–	–	–
[27]	2020	91.96	–	–	–
[36]	2020	96.2	–	–	–
[40]	2021	94.2	–	–	–
[41]	2021	90	–	–	–
Proposed DNN		99.23	99.15	99.23	99.18

6 Conclusion

In the research, a novel multilayer DNN is proposed for ITC. The real-world traffic, Moore-dataset, is pre-processed, and important features are selected using an extra-trees classifier. A DNN network is developed with seven hidden layers, and a maximum entropy classifier is used at the output layer to classify traffic into different classes. Four performance evaluation metrics: accuracy, precision, recall, and F1-score are calculated to evaluate the performance of the proposed methodology. Experimental results show that the proposed network has achieved the highest accuracy of 99.23% as compared to 98.56% using the SVM classifier and 98.90% using the KNN classifier. Thus, the proposed DNN can be used to accurately classify real-world internet traffic.

In machine learning algorithms, a dataset is required for the training and testing of the classifier. To achieve more accurate results in ITC, there is a need to start activities for the deployment of training datasets for artificial intelligence in computer networks.

Funding Statement: This work has supported by the Xiamen University Malaysia Research Fund (XMUMRF) (Grant No: XMUMRF/2019-C3/IECE/0007).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] H. Tahaei, F. Afifi, A. Asemi, F. Zaki and N. B. Anuar, "The rise of traffic classification in IoT networks: A survey," *Journal of Network and Computer Applications*, vol. 154, pp. 102538, 2020.

[2] Y. Yue, S. Li, P. Legg and F. Li, "Deep learning-based security behaviour analysis in IoT environments: A survey," *Security and Communication Networks*, vol. 2021, no. 1, pp. 1–13, 2021.

[3] M. Abbasi, A. Shahraki and A. Taherkordi, "Deep learning for network traffic monitoring and analysis (NTMA): A survey," *Computer Communications*, vol. 170, pp. 19–41, 2021.

[4] I. Cvitić, D. Peraković, M. Periša and M. Botica, "Novel approach for detection of IoT generated DDoS traffic," *Wireless Networks*, vol. 27, no. 3, pp. 1573–1586, 2021.

[5] T. T. T. Nguyen and G. J. Armitage, "A survey of techniques for internet traffic classification using machine learning," *IEEE Communications Surveys & Tutorials*, vol. 10, no. 1–4, pp. 56–76, 2008.

[6] IANA, Internet Assigned Numbers Authority (Accessed on 04 June 2021), 2021. [Online]. Available: https://www.iana.org.

[7] A. Madhukar and C. Williamson, "A longitudinal study of P2P traffic classification," in *14th IEEE Int. Symp. on Modeling, Analysis, and Simulation*, Monterey, CA, USA, pp. 179–188, 2006.

[8] A. W. Moore and K. Papagiannaki, "Toward the accurate identification of network applications," in *Int. Workshop on Passive and Active Network Measurement*, Springer, Berlin, Heidelberg, pp. 41–54, 2005.

[9] H. Shi, H. Li, D. Zhang, C. Cheng and X. Cao, "An efficient feature generation approach based on deep learning and feature selection techniques for traffic classification," *Computer Networks*, vol. 132, pp. 81–98, 2018.

[10] C. Yu, J. Lan, J. Xie and Y. Hu, "QoS-aware UTraffic classification architecture sing machine learning and deep packet inspection in SDNs," *Procedia Computer Science*, vol. 131, no. 1, pp. 1209–1216, 2018.

[11] A. Este, F. Gringoli and L. Salgarelli, "Support vector machines for TCP traffic classification," *Computer Networks*, vol. 53, no. 14, pp. 2476–2490, 2009.

[12] G. Sun, T. Chen, Y. Su and C. Li, "Internet traffic classification based on incremental support vector machines," *Mobile Networks and Applications*, vol. 23, no. 4, pp. 1–8, 2018.

[13] M. Shafiq, X. Yu, A. K. Bashir, H. N. Chaudhry and D. Wang, "A machine learning approach for feature selection traffic classification using security analysis," *The Journal of Supercomputing*, vol. 74, no. 10, pp. 1–26, 2018.

[14] L. Zhen and L. Qiong, "A new feature selection method for internet traffic classification using ml," *Physics Procedia*, vol. 33, pp. 1338–1345, 2012.

[15] P. Amaral, J. Dinis, P. Pinto, L. Bernardo, J. Tavares *et al.,* "Machine learning in software defined networks: Data collection and traffic classification," in *2016 IEEE 24th Int. Conf. on Network Protocols*, Singapore, pp. 1–5, 2016.

[16] J. Erman, M. Arlitt and A. Mahanti, "Traffic classification using clustering algorithms," in *Proc. of the 2006 SIGCOMM Workshop on Mining Network Data*, Pisa, Italy, pp. 281–286, 2006.

[17] D. Kreutz, F. M. V. Ramos, P. Verissimo, C. E. Rothenberg, S. Azodolmolky *et al.,* "Software-defined networking: A comprehensive survey," *Proc. IEEE*, vol. 103, no. 1, pp. 14–76, 2015.

[18] N. Namdev, S. Agrawal and S. Silkari, "Recent advancement in machine learning based internet traffic classification," *Procedia Computer Science*, vol. 60, pp. 784–791, 2015.

[19] C. Zhang, X. Wang, F. Li, Q. He and M. Huang, "Deep learning-based network application classification for SDN," *Transactions on Emerging Telecommunications Technologies*, vol. 29, no. 5, pp. e3302, 2018.

[20] M. Lopez-Martin, B. Carro, A. Sanchez-Esguevillas and J. Lloret, "Network traffic classifier with convolutional and recurrent neural networks for internet of things," *IEEE Access*, vol. 5, pp. 18042–18050, 2017.
[21] G. Sun, L. Liang, T. Chen, F. Xiao and F. Lang, "Network traffic classification based on transfer learning," *Computers & Electrical Engineering*, vol. 69, no. 4, pp. 920–927, 2018.
[22] S. Garg, K. Kaur, N. Kumar and J. J. P. C. Rodrigues, "Hybrid deep-learning-based anomaly detection scheme for suspicious flow detection in SDN: A social multimedia perspective," *IEEE Transactions on Multimedia*, vol. 21, no. 3, pp. 566–578, 2019.
[23] F. Ertam and E. Avcı, "A new approach for internet traffic classification: GA-WK-ELM," *Measurement*, vol. 95, no. 4, pp. 135–142, 2017.
[24] K. L. Dias, M. A. Pongelupe, W. M. Caminhas and L. de Errico, "An innovative approach for real-time network traffic classification," *Computer Networks*, vol. 158, no. 11, pp. 143–157, 2019.
[25] S. E. Gómez, L. Hernández-Callejo, B. C. Martínez and A. J. Sánchez-Esguevillas, "Exploratory study on class imbalance and solutions for network traffic classification," *Neurocomputing*, vol. 343, no. 3, pp. 100–119, 2019.
[26] M. Lotfollahi, M. J. Siavoshani, R. S. H. Zade and M. Saberian, "Deep packet: A novel approach for encrypted traffic classification using deep learning," *Soft Computing*, vol. 24, no. 3, pp. 1999–2012, 2020.
[27] J. Cao, D. Wang, Z. Qu, H. Sun, B. Li *et al.,* "An improved network traffic classification model based on a support vector machine," *Symmetry (Basel)*, vol. 12, no. 2, pp. 301, 2020.
[28] M. Kordestani, M. Saif, M. E. Orchard, R. Razavi-Far and K. Khorasani, "Failure prognosis and applications—A survey of recent literature," *IEEE Transactions on Reliability*, vol. 70, no. 2, pp. 1–21, 2019.
[29] G. Aceto, D. Ciuonzo, A. Montieri and A. Pescapé, "Mobile encrypted traffic classification using deep learning: Experimental evaluation, lessons learned, and challenges," *IEEE Transactions on Network and Service Management*, vol. 16, no. 2, pp. 445–458, 2019.
[30] G. Aceto, D. Ciuonzo, A. Montieri and A. Pescapè, "MIMETIC: Mobile encrypted traffic classification using multimodal deep learning," *Computer Networks*, vol. 165, no. 1, pp. 106944, 2019.
[31] S. P. RM., P. K. R. Maddikunta, M. Parimala, S. Koppu, T. R. Gadekallu *et al.,* "An effective feature engineering for DNN using hybrid PCA-GWO for intrusion detection in IoMT architecture," *Computer Communications*, vol. 160, no. 6, pp. 139–149, 2020.
[32] A. Moore, D. Zuev and M. Crogan, "Discriminators for use in flow-based classification," *Tech. Rep. RR-0513*, Dept. Comput. Sci., Univ. London, London, U.K., 2013.
[33] G. T. Reddy, M. P. K. Reddy, K. Lakshmanna, R. Kaluri, D. S. Rajput *et al.,* "Analysis of dimensionality reduction techniques on big data," *IEEE Access*, vol. 8, pp. 54776–54788, 2020.
[34] W. Li, M. Canini, A. W. Moore and R. Bolla, "Efficient application identification and the temporal and spatial stability of classification schema," *Computer Networks*, vol. 53, no. 6, pp. 790–809, 2009.
[35] H. Shi, H. Li, D. Zhang, C. Cheng and X. Cao, "An efficient feature generation approaiques for traffic classification," *Computer Networks*, vol. 132, no. 1, pp. 81–98, 2018.
[36] S. Pirmoradi, M. Teshnehlab, N. Zarghami and A. Sharifi, "The self-organizing restricted boltzmann machine for deep representation with the application on classification problems," *Expert Systems with Applications*, vol. 149, no. 2, pp. 113286, 2020.
[37] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Int. Conf. on Learning Representations*, San Diego, CA, pp. 1–13, 2015.
[38] Scikit-learn, (Accessed on 04 June 2021), 2021. [Online]. Available: http://scikit-learn.org.
[39] Numpy, (Accessed on 04 June 2021), 2021. [Online]. Available: https://www.numpy.org.
[40] S. Dong, "Multi class SVM algorithm with active learning for network traffic classification," *Expert Systems with Applications*, vol. 176, pp. 114885, 2021.
[41] S. Zhao, Y. Xiao, Y. Ning, Y. Zhou and D. Zhang, "An optimized k-means clustering for improving accuracy in traffic classification," in *Wireless Personal Communications*. Berlin, Germany: Springer, pp. 1–13, 2021.