Taylor & Francis
Taylor & Francis Group

Connection Science

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/ccos20

Hybrid feature learning framework for the classification of encrypted network traffic

S. Ramraj & G. Usha

To cite this article: S. Ramraj & G. Usha (2023) Hybrid feature learning framework for the classification of encrypted network traffic, Connection Science, 35:1, 2197172, DOI: 10.1080/09540091.2023.2197172

To link to this article: https://doi.org/10.1080/09540091.2023.2197172

© 2023 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group

Published online: 13 Apr 2023.

Submit your article to this journal

Article views: 1402

View related articles

View Crossmark data

Full Terms & Conditions of access and use can be found at
https://www.tandfonline.com/action/journalInformation?journalCode=ccos20

CONNECTION SCIENCE
2023, VOL. 35, NO. 1, 2197172
https://doi.org/10.1080/09540091.2023.2197172

Taylor & Francis
Taylor & Francis Group

OPEN ACCESS   Check for updates

Hybrid feature learning framework for the classification of encrypted network traffic

S. Ramraj[a] and G. Usha[b]

[a]Department of Computer Science and Engineering, SRM Institute of Science and Technology, Kattankullathur, Chennai, India; [b]Department of Computing Technologies, SRM Institute of Science and Technology, Kattankullathur, Chennai, India

ABSTRACT
The purpose of traffic classification is to allocate bandwidth to different types of data on a network. Application-level traffic classification is important for identifying the applications that are in high demand on the network. Due to the increasing complexity and volume of internet traffic, machine learning and deep learning methods are being used more frequently in traffic classification. The focus of this research is to evaluate the performance of the Support Vector Machine (SVM) in classifying network packets by application type, as well as classifying the type of data communicated within an application. The research considers encrypted network packets, including those from Virtual Private Networks (VPN) and the WhatsApp mobile application. Previous research has shown that deep learning methods are effective in the feature learning process, so this study uses a simple feed-forward Deep Neural Network (DNN) to improve the performance of the SVM algorithm. Additionally, various feature learning frameworks based on deep learning, such as DNN, Autoencoder and PCA, are compared. The study concludes that the DNN is able to improve the F1 score of the SVM classifier from 0.78 to 0.90. Furthermore, the study shows that using a hybrid framework of DNN with SVM can address the class imbalance problem often present in machine learning.

ARTICLE HISTORY
Received 8 January 2023
Accepted 24 March 2023

KEYWORDS
SVM; encrypted network traffic; WhatsApp classification

1. Introduction

The digital industry is revolutionising at a fast pace. The implementation and adaptation of new protocols and end-to-end encryption has made it a challenging task to carry out the classification of network packets. In order to face these rising challenges, measures need to be taken and novel methodologies need to be implemented in order to cope up with the new protocols. With the launch of new applications and social media platforms such as Facebook (Sudozai et al., 2017), Skype (Shim et al., 2017) the level of security and encryption is also tightening day by day. This has also contributed towards the rising complexity of the network packets.

CONTACT S. Ramraj ramrajitsrm333@gmail.com Department of Computer Science and Engineering, SRM Institute of Science and Technology, Kattankullathur, Chennai, India
This article has been corrected with minor changes. These changes do not impact the academic content of the article.

© 2023 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group
This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. The terms on which this article has been published allow the posting of the Accepted Manuscript in a repository by the author(s) or with their consent.

To have a better understanding of the ongoing network traffic, it is needed that we have a framework capable of capturing the network traces and providing a characterisation of it. Traffic classification (TC) is the medium of characterising the network data packets and labelling/segmenting them accordingly. A TC can be branched into two types: Classification based on Applications and Classification based on the file contents. Both these methods provide us with adequate knowledge of the network traces that we deal in our day-to-day life. In this paper, we have proposed a novel framework combined with a Deep Learning Neural Network that performs both types of classification mentioned above.

The end-to-end data encryption has disabled intruders to capture network traces and has made the internet a safer place to communicate. Encrypted data prevent intruders to perform any sort of malicious activities over the network and in return, the user's data is secured. At the same time, it has become a strenuous task for the Network Traffic Analyzers to gather the essential information for their research and future maintenance. Considering the strength of encryption, it is a complex task to classify a Network Packet into applications and content. A lot of research is being done in this arena to expand the scope of Network Traffic Classification with novel methodologies and with a greater accuracy. Moving forward with this approach, we have performed experiments on the dataset ISCXVPN2016 to give a clear characterisation of the end-to-end encrypted data.

Users today use VPNs (Virtual Private Networks) to protect their privacy from unwanted threats. This creates an extra tunnel of security and makes it even more challenging to carry out TC. With the use of VPN, one can conceal its identity including IP address from the main server and can access the internet more freely. The ISCXVPN2016 dataset is taken to utilise the benefit of VPN data entries from the dataset. In addition to this, an experimental setup is also laid to capture the WhatsApp network packets which are end-to-end encrypted. This combined set of VPN and Non-VPN data entries is then used to train the proposed model.

It is also worth mentioning that TC of network packets brings great importance to Traffic Analyzers and has its scope in the wide area of applications. TC is the initial step in the analysis and segmentation of the flow of data in a network. With the cognizance of the different types of applications being used by the users, Network Traffic analyzers can use this information for the purpose of advertisement and maintenance. It should also be noted that allocating resources for maintenance/updates of applications highly depends upon their usage by the user. An application having the highest network traces in a data packet should be monitored more closely for this purpose. An anomaly can also be traced (if any) in the future with an accurate TC.

Different approaches have been tested by researchers in the past few years. Some studies have also shown that a combination of ML and DL modules together can fulfil the requirement of understanding the complex features of Deep packets.

Considering the high-level encryption, in Alshammari and Zincir-Heywood (2009) the focus lies on making the use of only packet-size and time-of-arrival for TC. For this, Spiking Neural Networks (SNN) are used for observing the Time-related patterns. The simplicity of the SNN is the central point in Alshammari and Zincir-Heywood (2009). The challenges associated with the extensive list of features of network data packets are also observed in Alshammari and Zincir-Heywood (2010). However, in Alshammari and Zincir-Heywood (2010), the author has enhanced the performance of SVM by clubbing it with hybrid algorithms of feature selection and performance optimisers. Instead of fusing the framework with a DL-based model, Alshammari and Zincir-Heywood (2010) have improvised the performance of SVM based model with efficient algorithms. Cheng et al. (2011) and Wongyai and Charoenwatana (2012) have also proposed independent ML algorithms and Neural Networks for the classification of encrypted traffic. Similarly, a combination of both systems using information entropy is also present in Cheng et al. (2011). Our study aligns closely with Cheng et al. (2011) where an optimal set of features are fetched to three different ML classifiers along with a hybrid system of Deep Learning (Wang et al., 2022) also. The studies presented in Alshammari and Zincir-Heywood (2009, 2010) and Cheng et al. (2011) mainly differ in terms of the feature selection process.

In order to enhance the capabilities of an ML classifier namely SVM (Support Vector Machine), Deep Learning plays a central role in extracting the features and performing feature learning. These learned featured are then transferred to the classifier which outperforms the results of TC done using SVM only. This extra Deep Learning module thus acts as the brain behind the entire experiment as it extracts and reduces the features from the raw data to make it relevant for the classifier. Our results suggest that when Deep Neural Networks are clubbed with SVM, it enhances the performance of TC. As a result, our proposed model can enhance the performance of the SVM classifier and thus aid the Network traffic analyzers in carrying out the classification process with ease.

1.1. Contributions

The major contributions in this paper are:

(1) Introduction of WhatsApp network packets into Traffic Classification – The extensive use of WhatsApp application has made it a popular social media platform with millions of users worldwide. The large-scale use of this application along with the end-to-end encryption of data has thus attracted a lot of attention for traffic analysis. It is also important to note that such wide-scale applications stay protected from unwanted intrusions. As a result, we have laid an experimental setup under a monitored environment to capture the WhatsApp network packets in real time. These packets are captured over a secure network between the users of WhatsApp mobile version and the WhatsApp web.

(2) Integration of Deep Learning with ML classifiers – We have clubbed different Deep Learning modules with Machine Learning classifier SVM to evaluate its performance and boost its accuracy. A simple yet effective framework is presented in this paper. For the purpose of comparison and to back our aim, different models with a fusion of DL and SVM are rendered in this research.

(3) Multi – characterisation tasks of Network data packets – We have deduced and showcased the efficacy of our models with multiple characterisations of the network traces. Firstly, we have performed the identification of WhatsApp application from other network traces and secondly, we have segmented the WhatsApp application media content into image and text. We have also shown that the model fetched with the set of same features can perform different types of classification tasks as per the need.

1.2. Novelty

(1) A lot of open-source datasets are available for the purpose of traffic analysis and research. However, the unavailability of WhatsApp network data has enabled us to

capture their packets in real time traffic using an experimental setup. This WhatsApp data consists of various user activities (file & media transfers, messaging, contact & location sharing, etc.) and has an end-to-end encryption. As novelty, we have performed this experiment by capturing and using real time traffic data through the users over the network.

(2) Recent studies have used the traditional Machine Learning classifier SVM for various network classification applications. However, a very few have compared the results of SVM with the other algorithms. Therefore, this study includes a systematic comparison of SVM with other models like DNN, Autoencoder and PCA. Apart from this, a contrast is also made between SVM and hybrid systems of Deep Learning and Machine Learning models. This differentiation between models is unique in our research

The paper is organised in such a way that the related works in encrypted traffic analysis and its comparitive study are done in Section 2. Followed by the proposed methodology in Section 3. Dataset collection and preparation are discussed in Section 4. In Section 5 results from the experiments are discussed. 4 describes the acronyms used in the paper.

2. Related works

To cope up with the heightened level of encryption (Nguyen & Armitage, 2008), a lot of studies have been done on achieving true results for Traffic Classification (Park et al., 2008; Shen & Fan, 2008; Yoon et al., 2015) of Network Packets. In the past few years, new methodologies (Yuan et al., 2014) and frameworks have been implemented to meet the requirements and understand the rising complexity of Deep packets. In order to have a better understanding of the features in the network packets, many studies have also demonstrated the fusion of Deep Learning with ML classifiers (Cai et al., 2010).

In Coull and Dyer (2014) Zhanyi Wang has used Neural Networks to showcase its capability in identifying the network protocols via model training. This has elaborated the use of Neural Networks in feature extraction and feature learning. Wei Wang and others in Cuadra-Sanchez and Aracil (2017) have shown how a simplified framework of one-dimensional convolutional neural network can perform the traffic classification of end-to-end encrypted data. In Cuadra-Sanchez and Aracil (2017), they have shown that the relationship between the raw data and the output can be easily drawn and learned with their model. Tom and others in Datta et al. (2015) introduced a fusion of supervised machine learning with Bayesian-trained neural networks which had the advantage in a wider range of applications. The approach presented in Dorfinger (2010) called Seq2Img captures the static as well as the dynamic behaviours of the sequence. This approach averts the limitations associated with training the model with a handful of handcrafted features. Meanwhile Ehlert et al. (2006) and Fu et al. (2016) have elaborated upon a Deep Learning approach capable of segmenting the network into classes namely F2P and P2P. Along with this, the model proposed by Ehlert et al. (2006) and Fu et al. (2016) takes care of user application identification as well. Yet in another research presented in Goo et al. (2016) and Janani and Ramamoorthy (2022) by Manuel, Jun and others, have performed the Network Traffic Classification using novel methods. Goo et al. (2016) have enhanced the performance of an existing algorithm based on normalised thresholds by taking three simple properties of IP packets. The research presented by Janani and Ramamoorthy (2022) use an additional correlated information for enhancing the performance and overcoming the limitations of overfitting and availability of limited data set while training. Mobile traffic data is studied in Liu et al. (2019). In Rahman et al. (2022) use message statistics explored for traffic classification.

With the rising demand for Network traffic classification, a lot of studies have incorporated their models with convolutional spiking neural networks. A recent study indicated in Kumar and Sharma (2016), Lee et al. (2015), Liu et al. (2019) and Park et al. (2008) involve the use of the same model. The spiking neural networks have shown promising results in wide areas of applications which include detection, computation and recognition tasks. These models have been introduced in signal processing problems and have a wider scope in understanding the dynamic behaviour of the data packets. A comparitive study is given in table 2 and followed by the challenges in recent studies are tabulated in table 3.

Given the extensive use of Neural Networks, our main point of focus lies in distributing our overall experimental tasks into different modules of Deep Learning and Machine Learning. Moving forward with this approach, we have focused on enhancing the performance of Machine Learning classifier via the use of Neural Networks. Our results have supported our aim with promising results.

3. Proposed methodology

In this paper, we will be looking forward to three principal aims. The proposed work is shown in Figure 1. First, we will classify the data as WhatsApp Data or Other Application Data. Second, we will implement ML classifiers to predict if the dataset consists of images or text. Finally, we will fuse different deep learning modules including Neural Network, Autoencoder and PCA with the SVM classifier. This step is done to enhance the performance of the SVM classifier. In accordance with this, we will perform feature extraction through Deep Learning and pass it to suitable ML classifiers to speculate if the dataset consists of images or texts (Figure 1). These aims comprise of three main modules listed below:

(1) Data Pre-processing and Feature Scaling: We obtain the network packets by the process of Port Mirroring technique. In this process, the network packets get captured by

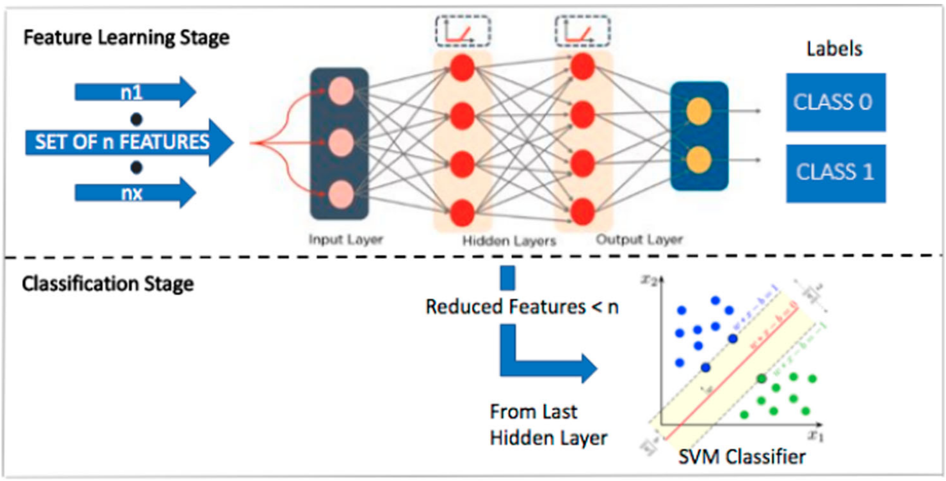

Figure 1. Feature learning using neural network followed by SVM classification.

a Port Mirroring Switch in the .pcap extension. These packets are then converted to the .csv extension using the CIC flowmeter tool. This tool performs mathematical calculations at the backend to generate new features from the raw data set. The dataset we achieved is of MXN dimensions, where M = 100 and N = 50,000. Since the size is too large, the dataset was cleaned and reduced to M = 42 and N = 35,000. In dataset cleaning, duplicate data is removed, and only essential features make up the cleaned dataset. Also, a new column, termed LABEL, is added to the combined dataset of all applications, including WhatsApp. This column defines to which application the data belongs. Next, feature scaling is applied to normalise the data using the MinMax Scaler. This crucial step helps in making the data entries lie within a suitable range of [0,1] to train the model accurately.

(2) Features Learning using Deep Learning: Previous studies and our results indicate that deep learning is suitable for feature learning. This approach of extracting features and feeding it to the classifier is termed as Transfer Learning. Following this approach, the pre-processed data along with the labels is passed to the Deep Learning module followed by its classification using SVM. With the selection of the relevant features and their extraction from the data set, the Neural Networks are compared with Autoencoders and PCA to check the classification performance by the SVM.

(3) Machine Learning Classifier: At the last stage of this transfer learning, the Machine learning module holds the extracted features from the NN and performs a classification on the network packets. The output from the Deep Learning module when fetched to the SVM classifier, boosts the performance of the classification task. As a result, a simplified architecture is presented in this research with involves a fusion of Deep Learning and Machine learning together. For the purpose of comparison, the output is first given to the SVM without feature learning. The same experiment is carried out with feature extraction through Neural Networks, Autoencoders and PCA. The accuracy and F-1 scores of each model is then compared in the result section.

The architectures for both experiments are shown in Figures 2, 3: Figure 3 renders an enhancement to the architecture shown in Figure 2. In the presence of Deep Learning, the feature extraction and feature learning are now done prior to the classification. This

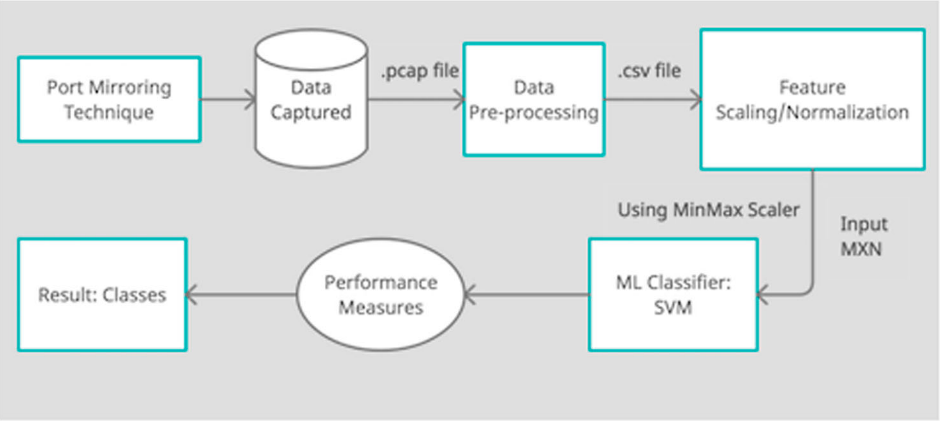

Figure 2. Classification architecture without feature extraction.

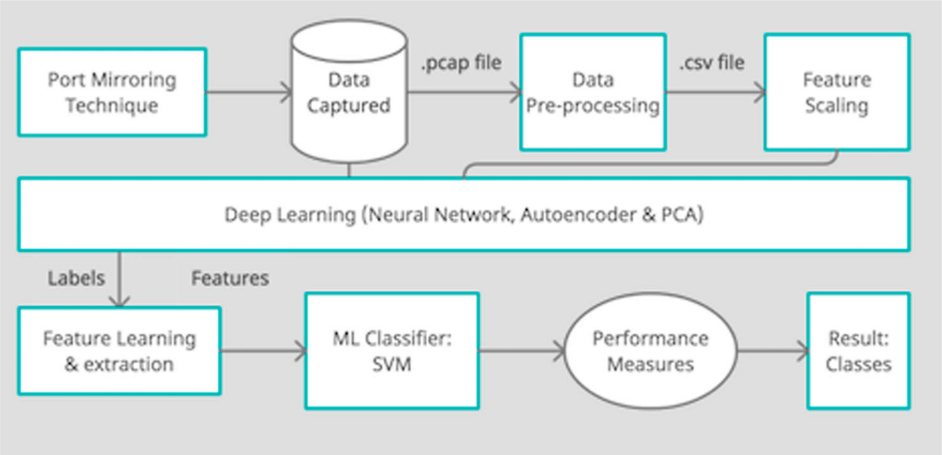

Figure 3. Classification architecture with feature extraction using deep learning.

extra module has helped in aggrandising the accuracy of the model with a considerable increase in the F-1 scores. In this architecture, the labels along with the learned features are fetched to the SVM for the classification task. The various algorithms used in the proposed framework are mentioned below:

```
Algorithm 1: Training a Deep Neural Network & extracting features
 Initialise:
 $i_n = \{input\} \rightarrow n\ features$
 Learned_Features = {output} → initiallyNULL
 $w_i = \{\} \rightarrow weights$
 $b = \{\} \rightarrow bias$
 L → layers
 Label → {0, 1}
 for each input do
     for each layer do
         predicted_label ← call forward_pass($i_n, w_i, b$)
         call calculate_loss(Predicted_label, Label)
         call  optimisation_backwardPass()
     end
 end
 LearnedModel ← {ı,$L, w_i, b$}
 for L in LearnedModel do
     if L is n-1 then
         Table 1 Learned_Features ← {i}
     else
         continue
     end
 end
```

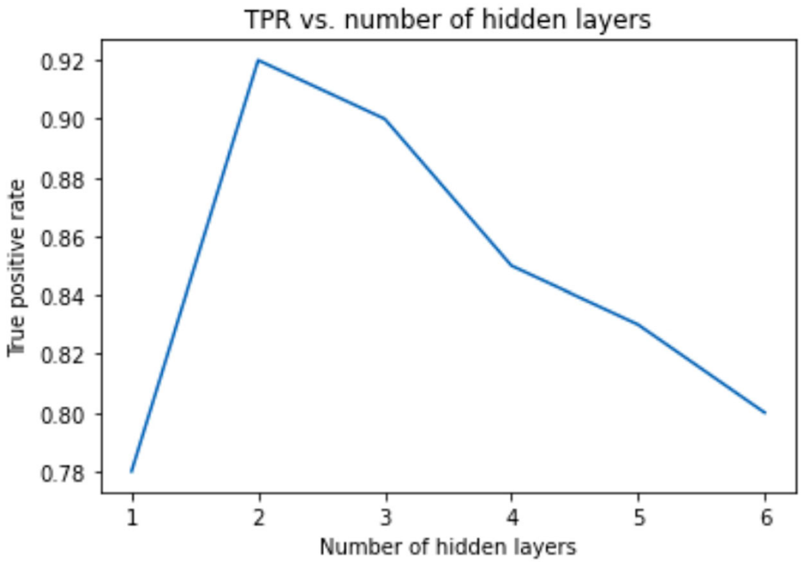

Figure 4. True positive rate vs number of hidden layers in DNN.

Algorithm 2: Forward propagation algorithm

```
for each hiddenLayers do
    for each hiddenLayer's neurons do
        set weightedSum to 0
        for each neuron's links do
            multiply links weight with associated previousLayer's neuron result
            add result to weightedSum
        end
        call relu(weightedSum)
        set neuron's value to result
    end
end
for each outputLayer's neurons do
    set weightedSum to 0
    for each neuron's links do
        multiply links weight with associated previousLayer's neuron result
        add result to weightedSum
    end
    call softMax(weightedSum)
    set neuron's value to result
end
```

Deep Neural Network (DNN) results in a decrease in the True Positive Rate (TPR) and an increase in the False Positive Rate (FPR) it is shown in Figure 4. This may be due to a phenomenon known as overfitting, where the model becomes too complex and starts to memorise the training data rather than learning to generalise to new data.

Table 1. List of acronyms.

Acronym	Description
SVM	Support vector machine
TC	Traffic classification
DNN	Deep learning neural network
VPN	Virtual private networks
PCA	Principal component analysis
NN	Neural networks
ROC	Receiver operating characteristic curve
AUC	Area under the curve

Table 2. Comparative study of related works.

Research article	Encrypted traffic	WhatsApp data	Dataset type	Methodology
Lotfollahi et al. (2020)	Yes	No	UNB ISCX VPN-nonVPN dataset	CNN
Wang et al. (2017)	Yes	No	ISCX VPN-nonVPN traffic dataset	DNN
Wang et al. (2018)	Yes	No	ISCX VPN-nonVPN	Stacked autoencoder+DNN+CNN
Wang et al. (2018)	Yes	No	Private dataset	Ensemble of deep learning methods
Aceto et al. (2019)	Yes	No	Human generated mobile traffic data set	DNN
Aceto et al. (2021)	Yes	No	ISCX VPN-nonVPN	DNN
This Paper	Yes	Yes	UNB ISCX VPN + Private data	DNN + ML

Table 3. Recent challenges and proposed solutions.

S.No	Recent challenges	Solution in the proposed system
1	The ISCXVPN2016 dataset has been commonly utilised in recent research related to classifying encrypted network traffic. However, it is important to note that this dataset does not include network packets from mobile apps such as WhatsApp.	In the proposed work the WhatsApp network packets are added with the ISCXVPN2016 dataset
2	One of the current challenges in research is the insufficient attention given to feature engineering in the context of classifying encrypted network traffic.	In the current research the feature learning process is automated with deep learning methods like deep neural network, PCA, Autoencoder

As the number of layers increases, the model becomes more complex and has more capacity to fit the training data. However, this increased capacity can also lead to the model becoming overly specialised to the training data, resulting in poorer performance on new, unseen data. This is reflected in the decrease in TPR and increase in FPR, as the model may start to classify some true positives as false positives due to overfitting (Tables 1–3).

In contrast, using a DNN with fewer layers may result in a model that is not complex enough to capture the underlying patterns in the data. This can lead to underfitting, where the model is unable to accurately capture the relationships between the input features and the output labels. The SVM classifier is trained in such a way that it accepts two parameters namely Labels and DL Extracted Features. The second parameter has been shown to significantly improve the classification performance when compared to TC using SVM alone.

The proposed module of DL consists of Deep Neural Networks (Yoon et al., 2012) comprising two hidden layers of varying features. Table 4 demonstrates the network and its

Table 4. Neural network layers.

Layer (type)	Output shape	Parameters
m1 input (Dense)	(None, 46)	141,358
m1 hidden1 (Dense)	(None, 20)	141,358
m1 hidden2 (Dense)	(None, 46)	940
m1 output (Dense)	(None, 2)	210

Table 5. Demonstration of deep neural network layers.

Number of layers	Loss function	Optimisation function
4	Sparse categorical loss	Sparse categorical entropy

parameters linked to each other. The capacity of this network is utilised to extract and learn the features of the Network Packets before it is passed onto the ML classifier. Before the features are passed onto the classifier, these are reduced and taken out from the last hidden layer.

This Deep Neural Network algorithm works by doing a forward propagation of features through the network of neurons. For each forward pass, the weights of the hidden layers are adjusted. A matrix multiplication of weights and previous layer features is calculated to provide the input to the next (hidden/output) layer. With this, the ReLu activation function in the hidden layers comes into action before the features are passed onto the last output layer. Once this loop of forward pass is completed, the result from the last hidden layer gets stored to be later passed into the SVM classifier (Table 5).

With each forward pass, the neural network also makes a backward propagation to correct the errors and adjust the weights or/and biases. With each backward propagation, the network becomes more and more accurate in terms of learning the features of the dataset. This error correction is a part of the Neural Network's learning process. Based upon the input and the layer in operation, the activation functions come into action. The algorithm for the same is mentioned below:

4. Dataset preparation

Data network packets consist of complex features which indicate the users' activities and the nature of applications used by them. Such data packets contain data entries which are end-to- end encrypted and even secured with an extra layer of VPN protection. The author in this paper has performed the Traffic Classification (TC) experiments on a similar type of dataset. ISCXVPN2016 dataset is used to conduct these experiments which include a set of VPN and Non-VPN data entries.

To test and conduct the experiment under more challenging conditions, an experimental setup consisting of the Port-Mirroring technique is implemented as shown in figure 5. With the use of a port-mirroring switch, the data packets are captured over the network which comprise of the raw data and features. A data pre-processing pipeline has been laid under which the raw data is analysed over the Wireshark software and then fetched to the CIC Flowmeter. At this point in the pipeline, the data gets cleaned and the features are extracted from the raw data packets. In addition to the network traces present in ISCXVPN2016, WhatsApp data packets are also added by following the steps in the above

Table 6. Dataset information.

Application	Number of samples
WhatsApp	38,804
Facebook messenger	13,060
Email	11,260
Hangout	900
Browsing	8000

experimental setup. Thus, in total the dataset consists of end-to-end encrypted data entries along with VPN traces (Table 5).

In addition to the WhatsApp application, VPN consists of Facebook, hangout. The dataset information is tabulated in Table 6. The relevant features obtained after the data preprocessing are limited to 45 features and include backward and forward transmission flow. For instance, backward header, bwd packet length along with forward header and fwd packet length are among the 45 features of the transmission flow. The importance of choosing the features in the Network Packets are thoroughly studied in the previous studies. Also, the Deep Learning implementation in the framework is proven to be prolific when it comes to feature extraction and feature learning.

5. Research objectives and experiments

In this research work, two objectives were pursued. The first objective was to classify encrypted network packets as belonging to either WhatsApp or not, which is a binary classification task. The second objective was to classify WhatsApp network packets according to the type of activity being performed, such as image transfer or text transfer, also a binary classification problem.

In the first level of experiments, three machine learning models – SVM, Random Forest and Logistic regression – were trained and tested, and their f1 scores were listed in Tables 8, 9. The results showed that SVM performed the best.

In the second level of experiments, DNN, PCA and Autoencoder were used for feature extraction, and SVM was trained with those features. The results are tabulated in Tables 6, 7. The experiments showed that DNN + SVM gave the best f1 score. Overall, the research demonstrated that a combination of DNN and SVM can effectively classify encrypted network packets as belonging to WhatsApp or not, as well as classify WhatsApp network packets according to the type of activity being performed.

Table 7. Comparison of various feature learning process for application identification.

Model	Precision	Recall	F1
SVM	0.87	0.85	0.83
Neural network	0.87	0.85	0.86
Neural network + SVM	0.93	0.92	0.90
Autoencoder+SVM	0.85	0.83	0.82
PCA + SVM	0.69	0.75	0.72

6. Results and discussion

The proposed model in Alshammari and Zincir-Heywood (2009) successfully performs a multi-classification process of traffic category like VoIP and the classification of encryption technique like VPN in the dataset. Different categories are made as per the classes in the dataset, and for each category, the study has indicated an accuracy of 99 and even 100 ML based classifiers and therefore supports our claim as well. Alshammari and Zincir-Heywood (2010) have claimed to increase the accuracy of original SVM by at least 9.28 other algorithms, yields a better result. Similarly, our study has also proved that when SVM is associated with a Neural Network, its performance is increased from F-1 score of 0.83 to 0.90 (for Application Identification) and from F-1 score of 0.78 to 0.90 (for Media Content segmentation) as tabulated in Table 7. In this paper, we performed a Traffic Classification (TC) task on the encrypted network data packets. Our aim is to classify the packets based on the application type followed by the classification of its media content (image or text). A dataset consisting of network traces of multiple applications including WhatsApp, Facebook, YouTube, Email, etc. was used to carry out this experiment (Figure 5).

Figures 6, 7 renders a clear indication between the different models used for the TC. This experiment aimed to classify between different data packets and identify the most widely used application WhatsApp from the others. The results are tabulated in Table 4 The models that were taken for comparison were SVM, Neural Networks, Neural Networks + SVM, Autoencoders + SVM and PCA + SVM. Upon checking and comparing the precision, recall and F-1 scores of all the 5 models used, it is found that Neural Networks + SVM performed the best among the others. With an F-1 Score of 0.94(for WhatsApp) and 0.90 (for others), NN+SVM outperforms the other models. On comparison it is indicated that when an extra

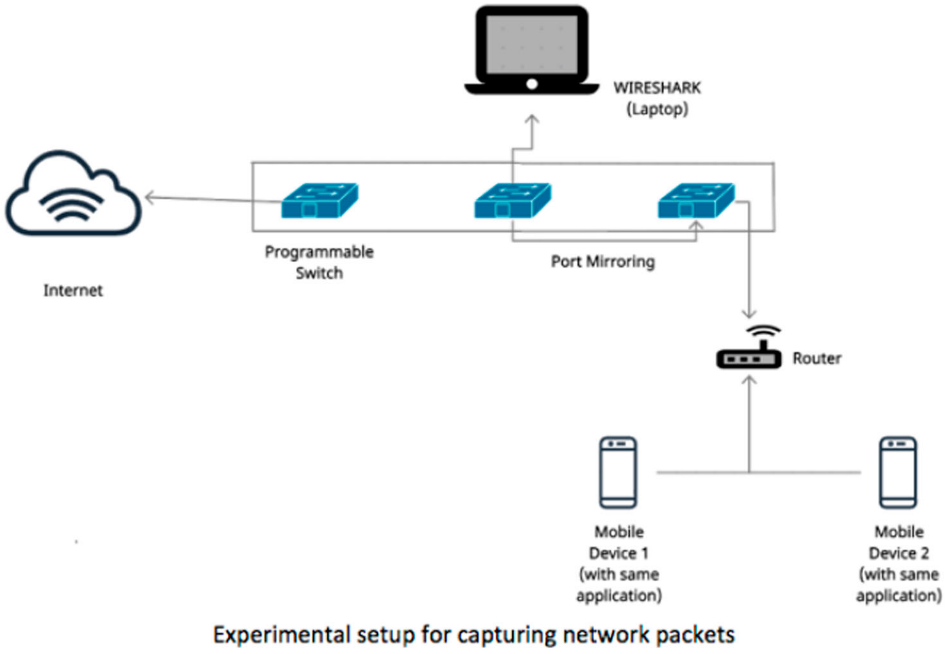

Figure 5. Experimental setup for dataset collection.

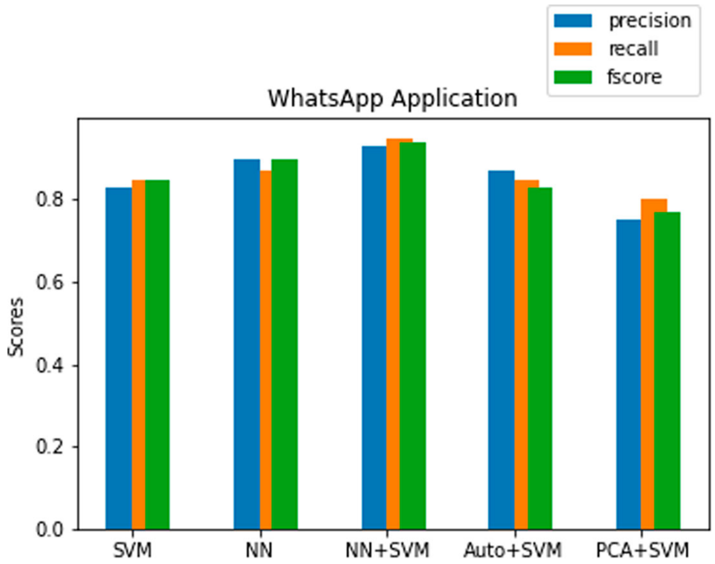

Figure 6. Precision, recall, F1 comparison between models for classifying WhatsApp application from others.

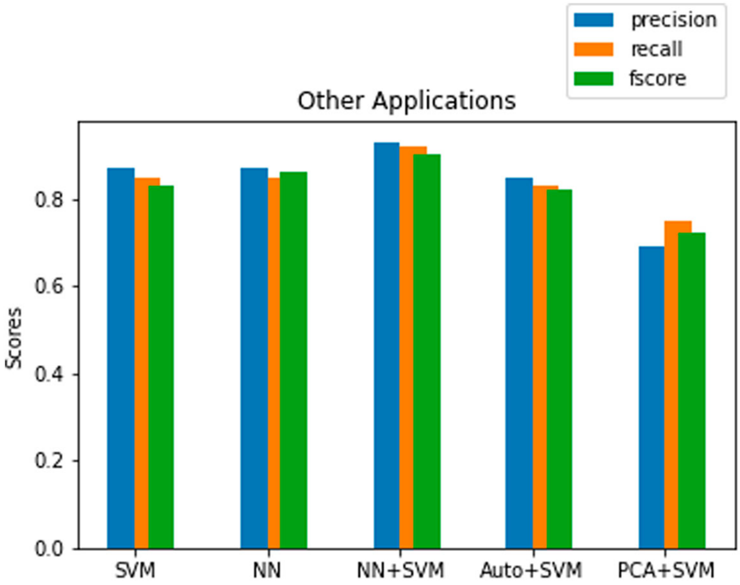

Figure 7. Precision, recall, F1 comparison between models for classifying other application from others.

Deep Learning module is induced with an SVM classifier, it enhances the performance of the Machine Learning algorithm and simultaneously takes care of the feature extraction and feature learning process. The proposed framework after being trained with a rigorous dataset of 45 features segmented 18,148 WhatsApp and 16,395 Other Applications network traces from the combined data set.

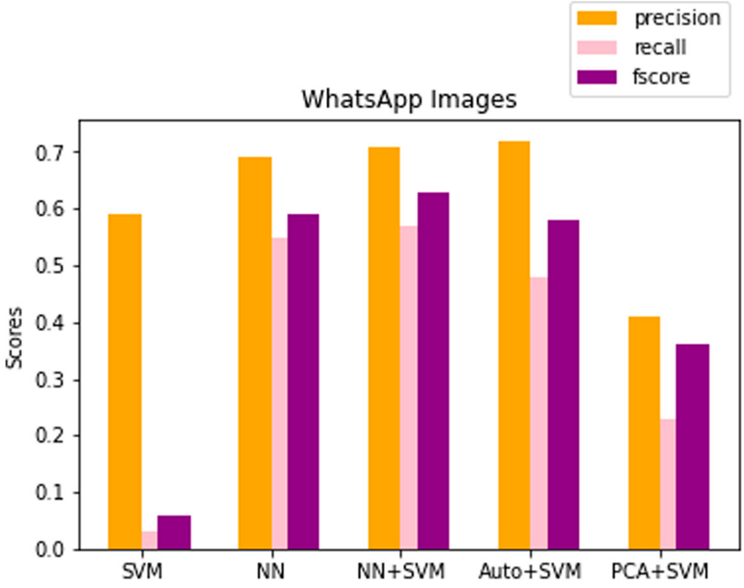

Figure 8. Precision, recall,F1 comparison between models for WhatsApp image.

Figures 7, 8 demonstrate the efficacy of the same five models tested in the previous experiment. This classification is aimed at segmenting the application's media content into Images and Text, where the application considered taken into consideration is WhatsApp. As the graph indicates, Neural Network+SVM performed the best among the other models and classified a total of 26,258 entries of WhatsApp text and 12546 entries of WhatsApp images. The findings of my research indicate that Support Vector Machine (SVM) performs well when trained with features extracted from a Deep Neural Network (DNN) for encrypted traffic classification. The DNN is able to learn high-level features from raw data, and these features are then used as input to the SVM classifier. The combination of these two methods improves the accuracy of the classification process. The SVM is particularly effective at identifying patterns in the feature space, while the DNN can learn complex and abstract features. This approach has several advantages, including the ability to handle high-dimensional data and the ability to deal with non-linear relationships between the features. Furthermore, this method can be used to classify different types of encrypted traffic, including Virtual Private Network (VPN) and WhatsApp mobile application packets, with high accuracy. These results suggest that combining SVM with DNN can be an effective approach for encrypted traffic classification. Thus, this successful classification of the data set into applications and their media content backs the Deep Learning + SVM classifier framework proposed in the paper.

In Figure 6, it is noticed that the exclusion of Deep Learning module leads to the prevailing problem of class imbalance. This problem associated with Machine learning leads to a reduced score and accuracy of the model after training. The results of various machine learning algorithm in classification of network traffic is tabulated in table 10.A class imbalance problem may arise with the limited availability of data set. As in our case, the WhatsApp application data has fewer data entries as compared to the VPN network traces in the

Table 8. Comparison of SVM and proposed work (NN + SVM) for classification WhatsApp media content (image/text).

Model	Precision	Recall	F1
SVM	0.68	0.99	0.78
Neural network	0.79	0.85	0.85
Neural network + SVM	0.81	0.93	0.90
Autoencoder+SVM	0.79	0.91	0.83
PCA + SVM	0.68	0.98	0.80

Table 9. Comparison of various machine learning models for application identification.

Model	Precision	Recall	F1
Random Forest	0.70	0.80	0.75
Logistic Regression	0.65	0.75	0.70
SVM	0.87	0.85	0.83

Table 10. Comparison of various machine learning models for classification of WhatsApp media Content (image/text).

Model	Precision	Recall	F1
Random forest	0.75	0.80	0.78
Logistic regression	0.78	0.75	0.73
SVM	0.68	0.99	0.78

combined dataset. This leads to an improper training of the model which yields reduced accuracy in classification. Whereas in Figure 4, the architecture with deep learning takes care of the class imbalances and thus enhances the performance of the model significantly.

The rise in F-1 score from 0.83 to 0.90 (WhatsApp/other) and 0.78 to 0.90 (for whatsapp image/text) in the Application classification is a clear indication that when Machine Learning Classifier SVM is clubbed with Deep Learning, the performance is improved, and the class imbalances are rectified as well.

Following the similar architecture, we have clubbed SVM with other models namely Autoencoder and PCA. The autoencoder layer details are expressed in Table 8. In both cases, the classification of media content (i.e. image or text) yields a significantly improved result when compared to SVM alone.

In Table 7, the rise in the F-1 score from 0.78 to 0.90 is a huge jump in the results in the classification of WhatsApp image and text. The comparision of the proposed work with the existing methodologies are given in Table 11.

In both the experiments of classification, the performance of SVM alone has been on the lower side as compared to the performance resulted from the fusion of Deep Learning and SVM (Tables 8–10).

Figures 9 and 10 indicate the ROC (Receiver operating characteristic curve) which highlights the performance of both models involving a Deep Learning framework. The ROC curve is a graphical medium of showcasing the accuracy of model by calculating its True Positive and False Positive Rate and drawing a relation between the two. This curve denotes the binary classification performance at each threshold point with which we can easily find

Table 11. Comparison of proposed methodology with other state of the art deep learning methods in classifying VPN traffic application.

Model	Precision
Lotfollahi et al. (2020)	0.93
Wang et al. (2017)	0.92
Aceto et al. (2021)	0.83
Proposed methodology	0.94

Table 12. Autoencoder layer details.

Layer	Activation	Dimension
Input	No	45
Layer 1 Dense	Relu	30
Layer 2 Dense	Relu	10
Layer 3 Dense	Relu	30
Output	Sigmoid	45

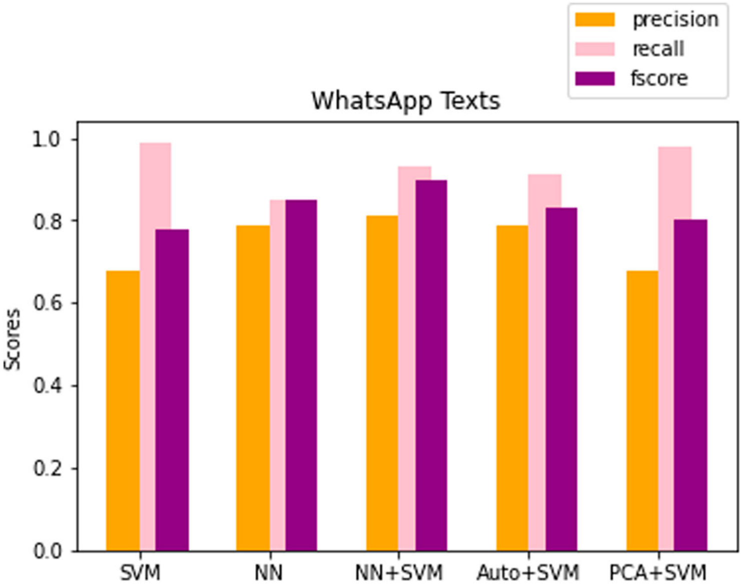

Figure 9. Precision, recall, F1 comparison between models for WhatsApp text.

which model performs better than the other. For the purpose of evaluating a binary classification model, ROC curve is a suitable representation to find AUC (area under the curve) for each model.

The ROC curves in Figures 9, 10 are generated while training our proposed models for the classification of WhatsApp image vs text. The ROC curve for Autoencoder + SVM has an area of 0.70 whereas the ROC curve for Neural Network + SVM has an area of 0.72. The result from this graphical representation indicates that feature learning with Neural Network is more fruitful than Autoencoders while segmenting the media content of WhatsApp application. In Figures 10, 11 the two graphs render different AUC (Area under the curve). As per the rule,

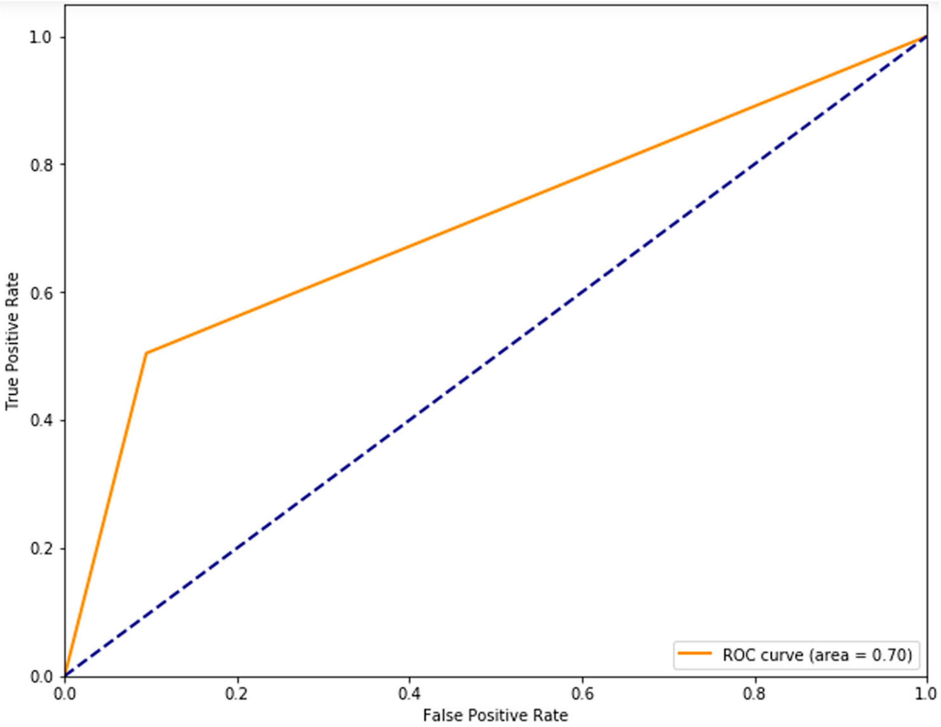

Figure 10. ROC curve for autoencoders + SVM.

the model with greater AUC tends to perform better as compared to a model with a lower AUC. In general, a higher AUC means that the model distinguishes more accurately and precisely between the positive and negative classes and thus identifies a greater number of True Positives and False Positives than True Negatives and False Negatives. As a result, the model Neural Network + SVM with an extra 0.02 AUC is considered better than the other models for the TC of WhatsApp text and images.

7. Conclusion and future work

All the models considered in the paper were analysed with their computational complexities. The model comprising of Deep Neural Networks has a total of 4 layers (i, j, k, l) with two hidden layers. The training of this model yielded a computational complexity of O(nt X (ij + jk + kl)), where n is the count of the epochs and t belongs to the training samples. For the model with the SVM classifier, the computational complexity turned out to be O(n^3),where n denotes the strength of the training data. For the PCA, it is O(min(p^3, n^3)), where p is the number of features considered and n are the data points. Whereas the models with a combination of multiple modules/algorithms have an overall complexity greater than all the previously mentioned models.

Finally, out of the five models, the union of Neural Networks with SVM classifier turned out to be the best among all the models. With the use of an extra Deep Learning module, it was possible to enhance the performance of the Machine Learning SVM classifier. However, even with the introduction of Deep Learning with Machine Learning algorithm,

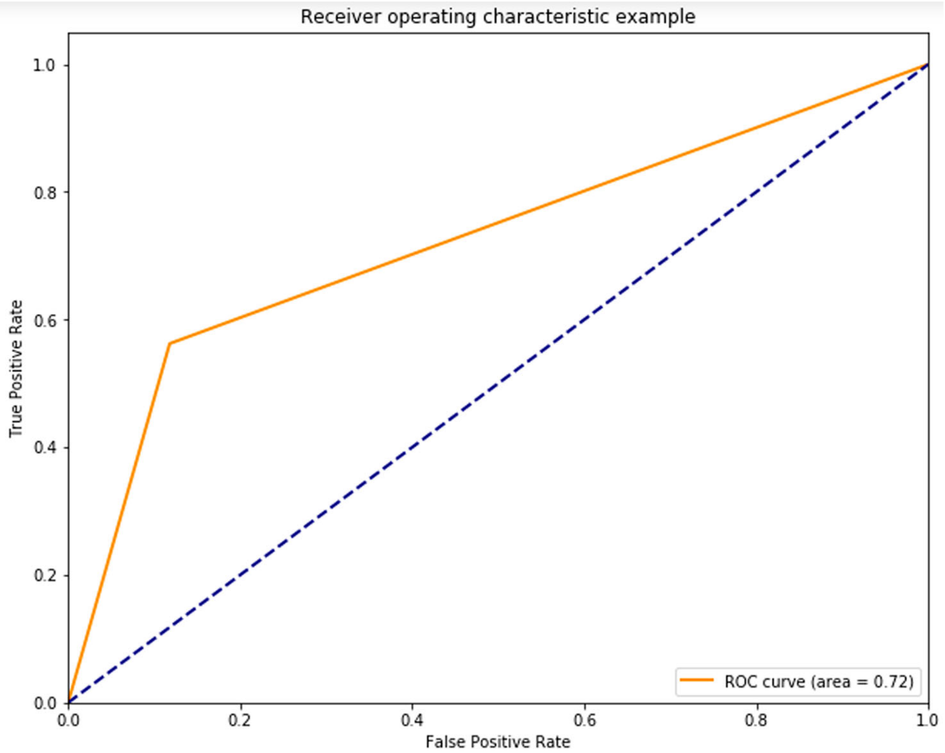

Figure 11. ROC curve for neural network + SVM.

there still exists some uncertainty in the segmentation of Network Traffic. For example, due to the limited availability of the dataset, we have not considered certain media segments of the WhatsApp application which include file sharing, location sharing and audio recording. Therefore, an extensive and a self-gathered dataset is required to check the proposed model with all the features of the application. Furthermore, it may be worthwhile to explore the use of other machine learning algorithms or ensemble methods, such as Random Forests or Gradient Boosting, to further improve the accuracy and efficiency of the classification process. Overall, the integration of XAI tools with the existing SVM and DNN models can lead to a more comprehensive and transparent approach to encrypted traffic classification.

Disclosure statement

No potential conflict of interest was reported by the author(s).

References

Aceto, G., Ciuonzo, D., Montieri, A., & Pescapé, A. (2019). Mimetic: Mobile encrypted traffic classification using multimodal deep learning. *Computer Networks*, *165*, 106944. https://doi.org/10.1016/j.comnet.2019.106944

Aceto, G., Ciuonzo, D., Montieri, A., & Pescapé, A. (2021). Distiller: Encrypted traffic classification via multimodal multitask deep learning. *Journal of Network and Computer Applications*, *183–184*, 102985. https://doi.org/10.1016/j.jnca.2021.102985

Alshammari, R., & Zincir-Heywood, A. N. (2009). Machine learning based encrypted traffic classification: Identifying SSH and Skype. In *Proceedings of the Second IEEE International Conference on Computational Intelligence for Security and Defense Applications* (pp. 289–296). IEEE Press.

Alshammari, R., & Zincir-Heywood, A. N. (2010). An investigation on the identification of VoIP traffic: Case study on Gtalk and Skype. In *2010 International Conference on Network and Service Management* (pp. 310–313).

Cai, Z., Liu, F., Xiao, N., Liu, Q., & Wang, Z. (2010). Virtual network embedding for evolving networks. In *2010 IEEE Global Telecommunications Conference Globecom 2010* (pp. 1–5).

Cheng, M., HUANG, X.-h., Xu, T., Yan, M., & Qi, J.-l. (2011). Automatic traffic signature extraction based on fixed bit offset algorithm for traffic classification. *The Journal of China Universities of Posts and Telecommunications*, *18*(1), 79–85. https://doi.org/10.1016/S1005-8885(10)60156-2

Coull, S. E., & Dyer, K. P. (2014). Traffic analysis of encrypted messaging services: Apple imessage and beyond. *ACM SIGCOMM Computer Communication Review*, *44*(5), 5–11. https://doi.org/10.1145/2677046.2677048

Cuadra-Sanchez, A., & Aracil, J. (2017). A novel blind traffic analysis technique for detection of WhatsApp VoIP calls. *International Journal of Network Management*, *27*(2), e1968. https://doi.org/10.1002/nem.v27.2

Datta, J., Kataria, N., & Hubballi, N. (2015). Network traffic classification in encrypted environment: A case study of google hangout. In *2015 Twenty First National Conference on Communications (NCC)* (pp. 1–6).

Dorfinger, P. (2010). *Real-time detection of encrypted traffic based on entropy estimation*. na.

Ehlert, S., Petgang, S., Magedanz, T., & Sisalem, D. (2006). Analysis and signature of Skype VoIP session traffic. *4th IASTED International*.

Fu, Y., Xiong, H., Lu, X., Yang, J., & Chen, C. (2016). Service usage classification with encrypted internet traffic in mobile messaging apps. *IEEE Transactions on Mobile Computing*, *15*(11), 2851–2864. https://doi.org/10.1109/TMC.2016.2516020

Goo, Y.-H., Shim, K.-S., Lee, S.-K., & Kim, M.-S. (2016). Payload signature structure for accurate application traffic classification. In *2016 18th Asia-Pacific Network Operations and Management Symposium (APNOMS)* (pp. 1–4).

Janani, K., & Ramamoorthy, S. (2022). Threat analysis model to control iot network routing attacks through deep learning approach. *Connection Science*, *34*(1), 2714–2754. https://doi.org/10.1080/09540091.2022.2149698

Kumar, N., & Sharma, S. (2016). Survey analysis on the usage and impact of WhatsApp messenger. *Global Journal of Enterprise Information System*, *8*(3), 52–57. https://doi.org/10.18311/gjeis/2016/15741

Lee, S.-H., Park, J.-S., Yoon, S.-H., & Kim, M.-S. (2015). High performance payload signature-based internet traffic classification system. In *2015 17th asia-pacific network operations and management symposium (apnoms)* (pp. 491–494).

Liu, Z., Japkowicz, N., Wang, R., & Tang, D. (2019). Adaptive learning on mobile network traffic data. *Connection Science*, *31*(2), 185–214. https://doi.org/10.1080/09540091.2018.1512557

Lotfollahi, M., Jafari Siavoshani, M., Shirali Hossein Zade, R., & Saberian, M. (2020). Deep packet: A novel approach for encrypted traffic classification using deep learning. *Soft Computing*, *24*(3), 1999–2012. https://doi.org/10.1007/s00500-019-04030-2

Nguyen, T. T., & Armitage, G. (2008). A survey of techniques for internet traffic classification using machine learning. *IEEE Communications Surveys & Tutorials*, *10*(4), 56–76. https://doi.org/10.1109/SURV.2008.080406

Park, B.-C., Won, Y. J., Kim, M.-S., & Hong, J. W. (2008). Towards automated application signature generation for traffic identification. In *Noms 2008–2008 IEEE Network Operations and Management Symposium* (pp. 160–167).

Rahman, M. H., Mofidul, R. B., & Jang, Y. M. (2022). Spectrum based wireless radio traffic classification using hybrid deep neural network. In *2022 Thirteenth International Conference on Ubiquitous and Future Networks (ICUFN)* (pp. 95–99).

Shen, G., & Fan, L. (2008). Network traffic classification based on message statistics. In *2008 4th International Conference on Wireless Communications, Networking and Mobile Computing* (pp. 1–4).

Shim, K.-S., Ham, J.-H., Sija, B. D., & Kim, M.-S. (2017). Application traffic classification using payload size sequence signature. *International Journal of Network Management*, *27*(5), e1981. https://doi.org/10.1002/nem.1981

Sudozai, M., Habib, N., Saleem, S., & Khan, A. (2017). Signatures of viber security traffic. *Journal of Digital Forensics, Security and Law*, *12*(2), 11. https://doi.org/10.15394/jdfsl.2017.1477

Wang, H., Zhou, S., Li, H., Hu, J., Du, X., Zhou, J., & Yang, H. (2022). Deep learning network intrusion detection based on network traffic. In *International Conference on Artificial Intelligence and Security* (pp. 194–207).

Wang, P., Ye, F., Chen, X., & Qian, Y. (2018). Datanet: deep learning based encrypted network traffic classification in sdn home gateway. *IEEE Access*, *6*(1), 55380–55391. doi: 10.1109/ACCESS.2018.2872430

Wang, W., Zhu, M., Wang, J., Zeng, X., & Yang, Z. (2017). End-to-end encrypted traffic classification with one-dimensional convolution neural networks. In *2017 IEEE International Conference on Intelligence and Security Informatics (ISI)* (pp. 43–48).

Wongyai, W., & Charoenwatana, L. (2012). Examining the network traffic of Facebook homepage retrieval: An end user perspective. In *2012 Ninth International Conference on Computer Science and Software Engineering (JCSSE)* (pp. 77–81).

Yoon, S.-H., Park, J.-S., & Kim, M.-S. (2012). Signature maintenance for internet application traffic identification using header signatures. In *2012 IEEE Network Operations and Management Symposium* (pp. 1151–1158).

Yoon, S.-H., Park, J.-S., & Kim, M.-S. (2015). Behavior signature for fine-grained traffic identification. *Applied Mathematics*, *9*(2L), 523–534.

Yuan, Z., Du, C., Chen, X., Wang, D., & Xue, Y. (2014). Skytracer: Towards fine-grained identification for Skype traffic via sequence signatures. In *2014 International Conference on Computing, Networking and Communications (ICNC)* (pp. 1–5).