Unsupervised Feature Learning Methodology for Tree based Classifier and SVM to Classify Encrypted Traffic

RAMRAJ S[1], Usha G[2]
Research Scholar, Department of Computer Science and Engineering
SRM Institute of Science and Technology, Kattankullathur, Chennai, India[1]
Associate Professor, Department of Computing Technologies
SRM Institute of Science and Technology, Kattankullathur, Chennai, India[2]

***Abstract*—Presently, sample social applications have emerged, and each one is trying to knock down the other. They expand their game by bringing novelty to the market, being ingenious and providing advanced level of security in the form of encryption. It has become significant to manage the network traffic and analyze it; hence we are performing a network traffic binary classification on one of the globally used application – WhatsApp. Also, this will be helpful to evaluate the sender-receiver system of the application alongside stipulate the properties of the network traces. By analyzing the behavior of network traces, we can scrutinize the type and nature of traffic for future maintenance of the network. In this study, we have carried out three different objectives. First, we have classified between the WhatsApp network packets and other applications using different ML classifiers, secondly, we have segmented the WhatsApp application files into image and text and third, we have incorporated a deep learning module with the same ML classifiers to understand and boost the performance of the previous experiments. Following the experiments, we have also highlighted the difference in the performance of both tree-based and vector-based classifiers of Machine Learning. Based on our findings, XGBoost classifier is a pre-eminent algorithm in the identification of WhatsApp network traces from the dataset. Whereas in the experiment of WhatsApp media segmentation, Random Forest has outperformed the other ML algorithms. Similarly, SVM when clubbed with a Deep Learning Auto encoder boosts the performance of this vector-based classifier in the binary classification task.**

Keywords*—Network traffic; encrypted network traffic; tree based classifiers; SVM

I. Introduction

All network applications need encryption as it provides authenticity, confidentiality and integrity to the users. In unencrypted network traffic, an intruder; whether spiteful (attacker), or not (e.g. network administrator tracking infrastructure) can read network packets and can view their contents. This leads to the intrusion of privacy and misuse of user's data. Whereas, in case of WhatsApp application, the data is end-to-end encrypted from the sender to receiver. Such applications do not leave room for any kind of violation of privacy. In widespread, encryption has a giant effect on detection and analysis of network traffic, because it conceals all payload statistics. As a result, new methodologies and frameworks are required to understand the complexity of Network traces without the need for decryption.

With an efficient and accurate Network Traffic Classification (TC), we can attain the cognizance of the nature and type of packets without the need of decryption. This is a secure way for Network traffic analysers to understand about the complex features of data packet. This could benefit the Network Traffic analysers in wide area of applications including advertising, allocating more bandwidth, understanding network patterns and its alterations etc. without the need of decryption. However, the rising strength of encryption requires efficient frameworks which can sustain the complexity of different and novel features of the data packets and can yield accurate results. In this study, we have used multiple Machine learning classifiers for performing binary TC. In accordance with that, we have also incorporated Deep Learning Auto encoder and PCA with ML classifiers to see their influence on the previous results. Over the last few studies, researchers have demonstrated how the inclusion of Deep Learning in the classification frameworks has improvised the results. This extra module of DL thus takes care of the packet's features and extracts them for the classifier. With an extracted set of features, the complexity for the classifier reduces and as a result it performs better. The results in this paper indicate the same and give a clear understanding of the performance of different models taken into consideration. The model which we proposed is to make the features learned from the deep learning algorithm such as auto encoder, PCA and those features will be feeded to train the machine learning model such as SVM, XGBoost, and Random forest. The performance analysis is done to verify whether the auto encoder or PCA helps the machine learning model to improve the classification in network traffic data.

A. Key Contributions

Our major contributions in this paper are:

- The available dataset [1] [2] does not includes WhatsApp network traffic traces. In this work we collected WhatsApp network traffic and integrate into the existing dataset.
- Implementing auto encoder, PCA with machine learning models for network traffic classification.
- Comparative study on the performance of tree based classifiers and large margin classifier for encrypted network traffic classification.

B. Introduction of WhatsApp Application Data in the Dataset

The WhatsApp Network traces are captured using the Port Mirroring Technique in a supervised environment over a secure connection. This enables us to club this captured data with open- source datasets available and use this combined dataset to train our proposed models.

C. Comparison of Tree-Based and Vector-Based ML Classifiers for Traffic Classification

A contrast is highlighted between Tree-based and Vector-based algorithms of Machine Learning and the results are thoroughly explained using precision and F-1 scores. At the end of this study, a claim is also made in favour of Tree-based algorithms for their excelling performance.

D. Feature Learning using Deep Learning and Comparison of Proposed Cross-Frameworks

An extra step is implemented to distribute the workload of Traffic Classification over different modules of the proposed framework. During the feature extraction and learning, Auto encoder and PCA come into action and pass the set of learned features to the classifier. This not only performs the TC but also boosts the performance of the model.

II. Related Work

The rising demand for Network Traffic Classification (TC) [3] [4]has led to many studies in the recent years. TC has use in wide areas of applications and holds a huge demand among the Network Analysers. A lot of studies have also demonstrated hybrid models that are known to have better accuracy in identifying large variety of applications.

In [5], T.T. Nguyen and others put forth the execution of ML techniques to IP traffic classification. They claim that the algorithms have demonstrated varied accuracy, even up to 99 percent, for a wide range of web application traffic. In [6], A. Dainotti and others have provided a wide range of worthy recommendations for traffic classification. According to one of their recommendations, the blend of traffic classification and algorithms should include a thorough analysis of efficiency and performance. Weibo Liu and others in [7] bestow the combination of Auto encoder, convolutional neural networks, deep belief network, and restricted Boltzmann machine. Using this combination, they indicate that we can now use unsupervised learning algorithms to process the unlabelled data. In [1], Hongtao Shi and others propose an approach that insists on dimensional reduction in feature space and overcomes the multi-class imbalance. Giuseppe Aceto and others in [2] put forward Deep Learning to build traffic classifiers based on auto-extracted features and reflect their traffic patterns. Finally, they have dissected existing DL algorithms in standard traffic classification. In [8], Chuan Guo and others put forth the calibration prospect of the ML algorithms. Their findings signify the effectiveness of temperature scaling on datasets. Arthur Callado and others in [9] propose techniques like signature-matching, sampling, and inherence, known in the field of IP traffic analysis, and focuses on application detection. In [10], Wei Wang and others present a new perspective of traffic classification using AI. They achieved good accuracy using a traffic classifier, which can learn features automatically (used CNN). Meanwhile, [11] and [12] are concerned with the privacy involved in network traffic analysis in applications present on the smartphone. They propose methods to secure end-to-end encryption as well as show the threats an eavesdropper can bid.

A similar approach is also investigated in [13] where the author proposes high performance multi class classification architecture capable of enhancing the classification results by up to +9.5 percent. The popularity and efficacy of DL based hybrid model is also evident in [14]. In [14], [15], [16] a pure DL framework with a series of Neural Network is proposed. The focus here lies on addressing a novel and updated experimental setup for an umbrella of TC tasks which are encrypted. In addition to this, sustainable frameworks are designed by researchers to use it for multi-classification tasks. In [17], a single architecture is proposed which can perform two tasks simultaneously. Task one being the characterization of the network traces based on F2P and P2P. The second task being the identification of applications [18]. With this single Deep Learning framework [19], the author has been able to further distinguish the packets into VPN and Non-VPN [20] traces followed by TC. The need of standard framework in network traffic analysis is discussed in [21].

Our research is greatly influenced by the concept of hybrid models and their multi-classification purpose. With that in mind, we have deduced different models and selected the most promising among them.

In this work the feature learning process is automated through the auto encoder, PCA. The proposed model trains the machine learning model [5] with the features learned using the auto encoder, PCA. A comparative study is done between the performance of the tree based classifiers and SVM. From the results it shows the performance of random forest improved much better with auto encoder.

III. Dataset Collection

Fig. 1 demonstrates the experimental setup involved in the data collection process. This setup consists of a router, a viable internet connection, a port mirroring switch, communicating devices and software for the purpose of analysis. The Wireshark software is used on the controlling unit which displays the features and network traces in a series of timeline. However, this raw data obtained from this setup is in the .pcap format. CIC Flowmeter tool is used to convert this extension in a usable format of .csv extension for our model to be trained. Once the data is converted, it passes through the pre-processing stage. Here, the CIC Flowmeter tool extracts more features from the raw data by performing mathematical calculations using statistics at the backend. These are called the derived features. Based on the previous studies, the relevant features for our model are considered and the others are omitted. These features include both the backward BWD and forward FWD transmission flow. Once the data is pre-processed, it is combined with an open-source network traffic dataset (ISCXVPN2016) to train our proposed model. As a result, this combined dataset includes WhatsApp network traces along with Other Applications. This experimental setup is carried out in a supervised environment and is one our major contributions in this paper.

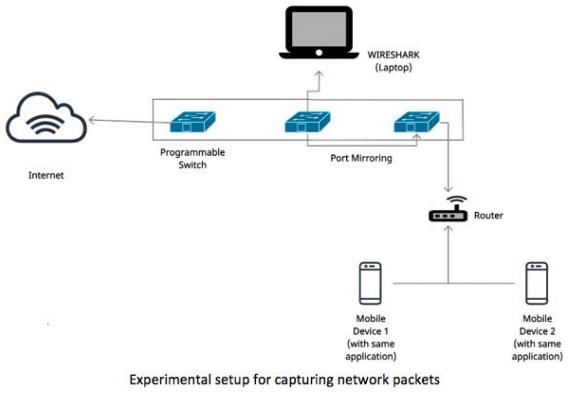

Fig. 1. Experimental setup for capturing network packets

A. Dataset

The dataset used in the experiment includes VPN [22] traffic data from the Canada Institute for Cybersecurity. Since VPN data are encrypted it is combined with WhatsApp traffic data. Total VPN data is 16395 which includes 'vpn_email', 'vpn_facebook', 'vpn_hangouts', 'vpn_spotify', 'vpn_youtube' and the WhatsApp data is about 17997. For experiments on classifying WhatsApp data as image, text the number of image data is 12546 and number of text data is 26258

IV. Proposed Methodology

Although a simple Machine Learning Classifier is capable of distinguishing between two types of data packets and segmenting them into classes, however with the rising complexity and security of social media applications, these traditional classifiers namely SVM, Random Forest, XGBoost etc. under perform.

Our proposed model in Fig. 2 is a dual stage framework with each stage performing an independent task in the TC. This not only boosts the efficiency of the model but also yields improvised results as compared to pure Machine Learning models. Stage 1 provides a pipeline where the data is captured and pre-processed before it is passed onto the classifiers. In this stage, MinMax scaler is implemented to normalize the data entries and convert them in the range of [0,1]. ML classifiers like SVM, XGBoost and Random Forest are tested on our self-gathered dataset. Also, a contrast between Tree-based (XGBoost and Random Forest) and Vector-based (SVM) algorithms is made at the end of each stage.

In order to improve the results obtained in stage 1, stage 2 is introduced with an extra module of Deep Learning. Stage 2 provides a fusion of Machine Learning and Deep Learning to form a Hybrid System. This system comes into action after the data passes through the pre-processing pipeline. For the purpose of comparison, Auto encoders and PCA are used with each classifier implemented in stage 1. Once the data is passed through the Auto encoders or PCA, it extracts the complex features from the dataset and provides a reduced set of relevant features which are then traversed back to stage 1. Here the normal flow of data is then followed by Machine Learning.

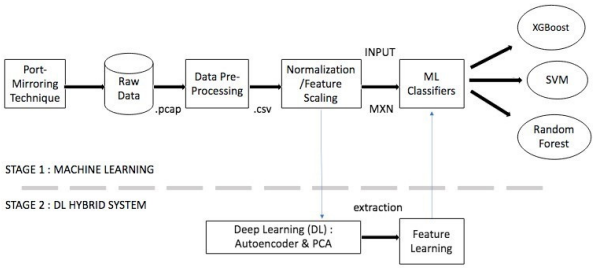

Fig. 2. A dual-stage hybrid architecture

This simple yet effective system has proved to enhance the outcome of classification obtained from stage 1.

The Random Forest algorithm is a tree-based algorithm suitable for selecting relevant features to perform classification. In the above algorithm, the tree starts at a single root node. At each node of the tree, a subset from the feature set is taken which is then split into different nodes. Each node denotes an attribute relevant for the classification of the Network Packets. f denotes the subset of features taken for each node, where f is much smaller than F. The decision to split a particular node is a computationally expensive process. By restricting each split in the tree, the rate of learning becomes faster.

Algorithm 1 Algorithm for Feature Learning with Auto Encoder

Input:[X1,X2,X3,....]

Feature in X1=X1,X2,....Xn
for each X **do**
    y=f(x)=se(w*x+bh)
    se(x) = sigmoid(x)=1 / (1+e-x)
    g(y)=sd(w*y+bf)
    sd = tanh(x)(ex-e-x) / (ex+e-x)
    Optimize θ=[W,bh,b'r]
    while Di = [x1, x2, x3, ..., xn] **do**
        JAE(θ)=$\sum$R(x,r) x$\in$Di
    end while
end for
Train_Feature =
Test_Feature=
for each Train_X in Se(X) **do**
    Train_Feature = X
end for
for each Test_X in Se(X) **do**
    Test_Feature = X
end for
se – Encoder Function
sd – Decoder Function

In Algorithm 1, where R denotes the reconstruction error, w' the weights given to the inputs of the hidden layer, and b' the biasness of the inputs given to the hidden layer.Train$_X$, Test$_X$ denotes the number of training and testing samples. The Train$_{Feature}$, Test$_{Feature}$ are used to train the machine learning models. Encoder encodes the input Xi to hidden representation hi. It does with the function h(X)=G(W*X+B). W is the set of weights, B is bias and G () is a nonlinear

function. The work of the decoder is to reconstruct the input from the hidden representation. Initially the weights and bias are randomly assigned and the values are optimized with every iteration. The loss function is used to calculate how much the hidden values are deviated from the original data. The network traffic data with dimensions of 21 features are given as input to the encoder. The encoder with sigmoid activation function produces a hidden representation of data with 10 dimensions. The loss function used for the decoder is binary cross entropy. The Adam optimizer is used for getting the right set of values for W and B. After the encoder optimizes the values, the features are transformed to train the machine learning models. The machine learning model such as SVM, XGBoost, and Random Forest are trained and tested with the features from auto encoder, PCA.

V. Results and Discussion

TABLE I. Comparison of ML Classifiers for WhatsApp and Other Applications

Model	Precision	Recall	F1 Score
Random Forest	0.84	0.99	0.91
XGBoost	0.98	0.97	0.98
SVM	1.00	0.90	0.94

In this experiment, we aim to classify the WhatsApp network packets from other applications. Evaluation of this experiment is conducted using a self-gathered dataset. The dataset consists of encrypted traffic [23] flow which is gathered using a port-mirroring switch and a network router. The raw data consists of more than 35 features which are eventually narrowed down to 21 features after doing a considerable amount of data analysis. The influence of original (non-normalized) and normalized data has been studied in this experiment. The encrypted data is then normalized using MinMax Scaler for our distance based algorithms to yield correct and accurate results.

Towards the end of this experiment, we propose a comparison between vector based and tree-based machine learning classifiers. The normalized traffic flow data is fetched into three machine learning algorithms, namely, SVM, XGBoost and Random Forest algorithm. The comparison between these three classifiers is illustrated in Table I. It demonstrates the effectiveness and performance of tree-based algorithms over the vector-based classifiers. With a F-1 score of 0.92 (for WhatsApp) and 0.98 (for others), XGBoost succeeds the other classifiers in this experiment. With the analysis of the F-1 scores derived above, we conclude that tree-based algorithms perform better than vector-based algorithms in the classification of network packets.

In the second experiment we have presented the comparison between different machine learning classifiers in the classification of image and text files of WhatsApp Network packets. The network packets obtained using the Port mirroring switch consists of a combination of different file transmissions. Due to their encrypted nature, it becomes challenging to classify them into different classes based on the nature of these files. As a result, tree- based and vector-based classifiers in machine learning are used for this purpose.To have a better understanding, the author has denoted the media files as class 1 and text files as class 0. For each class, precision, recall and F-1 score is calculated to measure and analyze the performance of these classifiers. Table II gives the performance metrics for each class and its classifier. In Table II, Random Forest and XGBoost are the tree-based classifier which follows the approach of branching for classification tasks. On the other hand, SVM follows a vector- based approach to classify the packets. Upon comparison, it is found that Random Forest overcomes the performance of XGBoost and SVM. With an F-1 score of 0.91 (class 0) and 0.80 (class 1) for image files and text files classification, Random Forest has been the best classifier among the other two algorithms. Followed by Random Forest, it is noticed that XGBoost is closer to Random Forest in terms of performance with an F-1 score of 0.86 (class 0) and 0.67 (class 1). From this experiment, we conclude that tree-based algorithms perform better than vector-based classifiers and Random Forest has achieved better results than XGBoost and SVM in the classification of image and text files of WhatsApp packets.

TABLE II. Classification of WhatsApp Files into Text and Image using ML Classifiers

Model	Precision	Recall	F1 Score
Random Forest	0.88	0.95	0.91
XGBoost	0.83	0.90	0.86
SVM	0.72	0.91	0.80

TABLE III. Comparison of Proposed Hybrid Systems for WhatsApp Content Classification

Model	Precision	Recall	F1	Accuracy
Auto Encoder + SVM	0.80	0.80	0.84	0.90
Auto Encoder +XGBoost	0.83	0.88	0.85	0.91
Auto Encoder +Random Forest	0.85	0.84	0.84	0.88
PCA + SVM	0.68	0.98	0.80	0.85
PCA + XGBoost	0.73	0.66	0.69	0.78
PCA + Random Forest	0.70	0.95	0.81	0.84

In the last experiment, we present a method and a framework for efficient and effective feature extraction followed by ML classification. This architecture consists of deep learning modules and machine learning classifiers in order to fulfill the objective of our experiment i.e. classification of WhatsApp network packets into text and image. During this experiment, we show that our implementation of the framework can extract the features of the network packets which are encrypted and unlabelled. For the purpose of comparison, Auto encoders and PCA are taken into consideration for feature extraction.

As we move forward in this architecture, tree-based and vector-based machine learning classifiers are implemented to classify the packets based on the features extracted. For each feature extraction module, three classifiers are tested, namely, SVM, XGBoost and Random Forest. Table III highlights the results obtained after testing all the algorithms. SVM when implemented alone performed the lowest among XGBoost and Random Forest as observed in experiment 2. However, in contrast to this, it is seen that auto encoders are improvising the results of SVM in experiment 3. With the use of Auto encoders, the F-1 scores of SVM have drastically improved

whereas it has been nearly same for the other two classifiers. From the chart shown in Fig. 3 and 4, it is clear that PCA has failed to show any improvement in the classification task, the results suggest that feature extraction through auto encoders has contributed towards a positive learning curve. In conclusion, the deep learning module – auto encoders has provided a better result when clubbed with SVM in comparison to when SVM is implemented alone. Fig. 5 and 6 shows how the precision and recall value varies between the different models with respect to auto encoder, PCA usage.

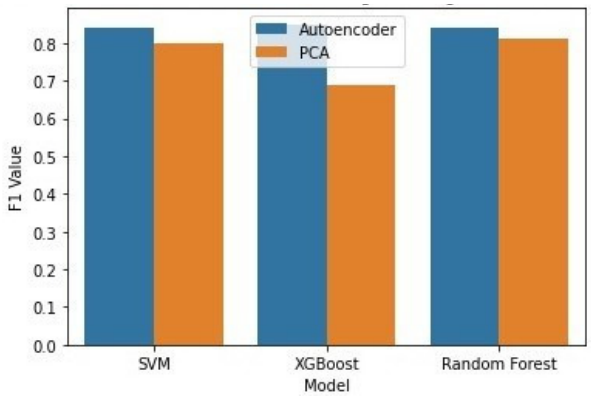

Fig. 3. Comparison of auto encoder and PCA with machine learning model in classifying WhatsApp image from text

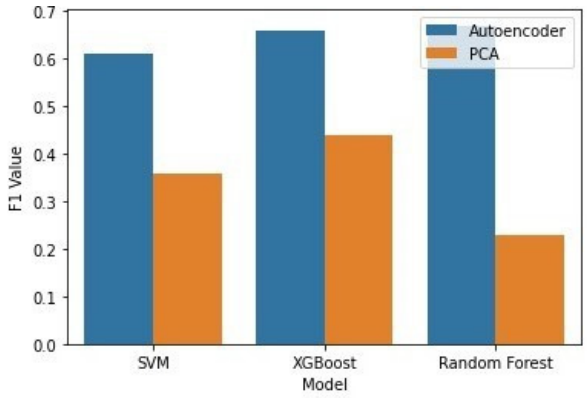

Fig. 4. Comparison of auto encoder and PCA with machine learning model in classifying WhatsApp text from image

VI. Conclusion and Future Work

Every tactical model performed differently under our experimental environment. This is also attributed to the dataset as well as the computational complexity of the algorithms. Thus, the model yielding the best results should also be efficient enough to perform TC on large datasets. It should also be noted that the efficiency of the model is directly related to its performance in real time TC.The computational complexity of each algorithm is closely scrutinized. The vector-based classifier SVM has a computational complexity of O (n^3), wherein n is the training data's strength. While the computational complexity of Tree based algorithms highly depends upon the number of attributes taken into consideration which is 45

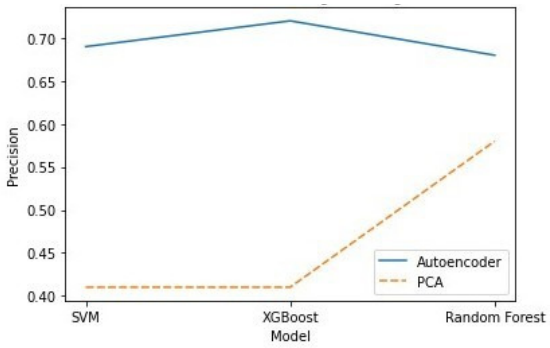

Fig. 5. Comparison of auto encoder and PCA with machine learning model in classifying WhatsApp image from text in terms of precision

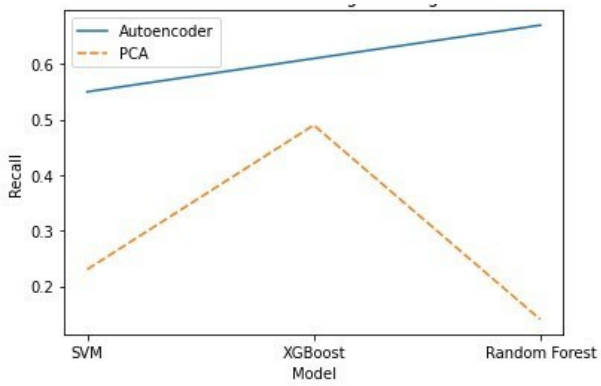

Fig. 6. Comparison of auto encoder and PCA with machine learning model in classifying WhatsApp image from text in terms of recall

features in our case. Thus, the feature count becomes directly proportional to the number of subtrees formed in the model. As a result, the computational complexity of XGBoost for learning each tree becomes O (nlogn). In the case of Random Forest algorithm, the computational complexity is O (TD), where T is the size of random forest and D is the maximum depth. In case of Random Forest, the subtree balance and D highly influence the results. Different tactical models and hybrid systems are tested in our paper and a conclusion is drawn in favour of the Tree-based classifiers. Upon testing all the classifiers upon our self-gathered dataset, it is concluded that Tree-based classifiers (XGBoost and Random Forest) outperforms the Vector-based classifier (SVM) and yields a better accuracy and F-1 score in Network TC. Therefore, from experiment 1, Performance (XGBoost) is greater than the Performance of Random Forest and SVM. Similarly, in experiment 2, the performance of Random Forest and XGBoost are interchanged whereas SVM remains the last in comparison. This clearly indicates that Tree-based classifiers are better in performance than Vector-based classifiers. Also, the use of Deep Learning for feature extraction has given a boost to the results of SVM. Thus, auto encoder reduces the complexity of the features and supports the classifiers in the classification process.

Certain cases are expected to be covered in the future work for making this proposed architecture a state-of the art system. This includes the segmentation of other media files including file transfer, voice message and location sharing.

Apart from this, other Deep Learning models like CNN, Deep Neural Networks, RNN, etc. are yet to be tested upon this dataset. Thus, a further investigation is required with other Deep Learning modules. In future the work can be extended to other WhatsApp data such as text, voice, etc. The development of an intrusion detection system for encrypted data such as WhatsApp, Telegram.

REFERENCES

[1] 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.

[2] G. Aceto, D. Ciuonzo, A. Montieri, and A. Pescapé, "Mobile encrypted traffic classification using deep learning," in *2018 Network traffic measurement and analysis conference (TMA)*. IEEE, 2018, pp. 1–8.

[3] H. Wang, S. Zhou, H. Li, J. Hu, X. Du, J. Zhou, Y. He, F. Fu, and H. Yang, "Deep learning network intrusion detection based on network traffic," in *International Conference on Artificial Intelligence and Security*. Springer, 2022, pp. 194–207.

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

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

[6] A. Dainotti, A. Pescape, and K. C. Claffy, "Issues and future directions in traffic classification," *IEEE network*, vol. 26, no. 1, pp. 35–40, 2012.

[7] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, "A survey of deep neural network architectures and their applications," *Neurocomputing*, vol. 234, pp. 11–26, 2017.

[8] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *International conference on machine learning*. PMLR, 2017, pp. 1321–1330.

[9] A. Callado, C. Kamienski, G. Szabó, B. P. Gero, J. Kelner, S. Fernandes, and D. Sadok, "A survey on internet traffic identification," *IEEE communications surveys & tutorials*, vol. 11, no. 3, pp. 37–52, 2009.

[10] W. Wang, M. Zhu, X. Zeng, X. Ye, and Y. Sheng, "Malware traffic classification using convolutional neural network for representation learning," in *2017 International conference on information networking (ICOIN)*. IEEE, 2017, pp. 712–717.

[11] V. F. Taylor, R. Spolaor, M. Conti, and I. Martinovic, "Robust smartphone app identification via encrypted network traffic analysis," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 1, pp. 63–78, 2017.

[12] B. Saltaformaggio, H. Choi, K. Johnson, Y. Kwon, Q. Zhang, X. Zhang, D. Xu, and J. Qian, "Eavesdropping on {Fine-Grained} user activities within smartphone apps over encrypted network traffic," in *10th USENIX Workshop on Offensive Technologies (WOOT 16)*, 2016.

[13] R. Schuster, V. Shmatikov, and E. Tromer, "Beauty and the burst: Remote identification of encrypted video streams," in *26th USENIX Security Symposium (USENIX Security 17)*, 2017, pp. 1357–1374.

[14] P. Velan, M. Čermák, P. Čeleda, and M. Drašar, "A survey of methods for encrypted traffic classification and analysis," *International Journal of Network Management*, vol. 25, no. 5, pp. 355–374, 2015.

[15] R. Chapaneri and S. Shah, "Enhanced detection of imbalanced malicious network traffic with regularized generative adversarial networks," *Journal of Network and Computer Applications*, vol. 202, p. 103368, 2022.

[16] M. Lotfollahi, M. Jafari Siavoshani, R. Shirali Hossein 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.

[17] A. Rago, G. Piro, G. Boggia, and P. Dini, "Multi-task learning at the mobile edge: An effective way to combine traffic classification and prediction," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 9, pp. 10 362–10 374, 2020.

[18] S. Dong, "Online encrypted skype identification based on an updating mechanism," *arXiv preprint arXiv:2203.12141*, 2022.

[19] S. Rezaei and X. Liu, "Deep learning for encrypted traffic classification: An overview," *IEEE communications magazine*, vol. 57, no. 5, pp. 76–81, 2019.

[20] L. Chen, Y. Xue, Y. Mu, L. Zeng, F. Rezaeibagha, and R. Deng, "Case-sse: Context-aware semantically extensible searchable symmetric encryption for encrypted cloud data," *IEEE Transactions on Services Computing*, 2022.

[21] J. Holland, P. Schmitt, P. Mittal, and N. Feamster, "Towards reproducible network traffic analysis," *arXiv preprint arXiv:2203.12410*, 2022.

[22] "VPN-nonVPN dataset (ISCXVPN2016)," [21] https://www.unb.ca/cic/datasets/vpn.html, accessed: 2022-08-30.

[23] D. F. Isingizwe, M. Wang, W. Liu, D. Wang, T. Wu, and J. Li, "Analyzing learning-based encrypted malware traffic classification with automl," in *2021 IEEE 21st International Conference on Communication Technology (ICCT)*. IEEE, 2021, pp. 313–322.