A Deep Learning Based Method for Handling Imbalanced Problem in Network Traffic Classification

Ly Vu
Le Quy Don Technical University
Hanoi, Vietnam
vuthily.tin2@gmail.com

Cong Thanh Bui
Posts and Telecommunications
Institute of Technology
Hanoi, Vietnam
congthanhttmt@gmail.com

Quang Uy Nguyen
Le Quy Don Technical University
Hanoi, Vietnam
quanguyhn@gmail.com

ABSTRACT

Network traffic classification is an important problem in network traffic analysis. It plays a vital role in many network tasks including quality of service, firewall enforcement and security. One of the challenging problems of classifying network traffic is the imbalanced property of network data. Usually, the amount of traffic in some classes is much higher than the amount of traffic in other classes. In this paper, we proposed an application of a deep learning approach to address imbalanced data problem in network traffic classification. We used a recent proposed deep network for unsupervised learning called Auxiliary Classifier Generative Adversarial Network to generate synthesized data samples for balancing between the minor and the major classes. We tested our method on a well-known network traffic dataset and the results showed that our proposed method achieved better performance compared to a recent proposed method for handling imbalanced problem in network traffic classification.

CCS CONCEPTS

- **Networks** → Network reliability;

KEYWORDS

Deep learning, Network traffic classification,Auxiliary classifier GAN

ACM Reference Format:
Ly Vu, Cong Thanh Bui, and Quang Uy Nguyen. 2017. A Deep Learning Based Method for Handling Imbalanced Problem in Network Traffic Classification. In *SoICT '17: Eighth International Symposium on Information and Communication Technology, December 7–8, 2017, Nha Trang City, Viet Nam.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3155133.3155175

1 INTRODUCTION

Analysing of network traffic has an important role in many problems such as planning of resource usage, network application performance assessment, Quality of Service control, generating traffic model for researches [7]. In network analysis, traffic classification is one of the main problems. Network traffic classification has been applied to a number of applications including determining the usage and the development trend of applications, anomaly detection, accounting etc. Moreover, in most of Quality of Service (QoS) technique, traffic classification is used to prioritize applications across the limited bandwidth [22].

There are three approaches for traffic classification problem which are port-based methods, payload-based methods and flow statistics-based methods [8]. Port-based methods use port number in the packet header in order to check well-known applications [18]. This method is simple and easy to implement but it does not always provide a reliable result. Many recent applications use dynamic ports or even hide themselves by using a well-known port of other applications. Payload-based methods dig the signatures of applications in the payload of packets [21]. This method avoids the dynamic port problem. However, payload-based methods can not work well with encrypted traffic since we are unable to watch the encrypted traffic without decrypting it. Recently, researchers paid more attention to flow statistics-based methods. These methods use the statistic features of flow instead of requiring the content of packets allowing them the ability to deal with encrypted traffic.

Flow statistics-based methods often employ supervised and unsupervised machine learning algorithms to classify network traffic into predefined classes of known applications[22]. However, one of the challenges in traffic classification is the skew of network traffic data. Since, real applications on the Internet are disparity, most of the collected network traffic data is imbalanced [17]. Subsequently, handling the imbalanced problem in traffic data is essential for machine learning algorithms to achieve better performance in classifying network traffic.

In this paper, we propose an application of a deep network structure called Auxiliary Classifier GANs (AC-GAN) [16] to generate synthesized samples for network traffic classification problem. To the best of our knowledge, this is the first attempt to use deep learning in generating synthesized data in network traffic classification. The synthesized data is then combine with the original/real data to form the new training dataset

SoICT '17, December 7–8, 2017, Nha Trang City, Viet Nam
© 2017 Association for Computing Machinery.
ACM ISBN 978-1-4503-5328-1/17/12. . . $15.00
https://doi.org/10.1145/3155133.3155175

for classification algorithms. We applied three classification algorithms including support vector machine (SVM), decision tree (DT) and random forest (RF) on the new augmented training dataset. The experimental results show that using AC-GAN for generating synthesized data helps classification algorithms to achieve better performance in network traffic classification problem regarding to three popular performance metrics including accuracy score, F1 score and AUC score.

The rest of this paper is organized as follows: Section 2 briefly reviews the previous works in network traffic classification field; Section 3 presents AC-GAN algorithm; Section 4 describes the design of our proposed system. The experimental settings are presented in Section 5 . The results are provided and analysed in Section 6 ; Finally, Section 7 draws the conclusion and highlight some future works.

2 RELATED WORKS

There have been three popular methods for classifying network traffic: port-based, content-based and statistical-based. Among them, statistical-based (or flow-based) traffic classification has many advantages comparing to port-based and content-based approaches [14]. This method can avoid problems causing by port-based and content-based approaches such as encrypted applications, privacy and dynamic ports. In statistical-based methods, researchers have often applied some form of machine learning algorithms to classify network traffic into different applications [22].

Vladuto et al. [19] surveyed the application of machine learning algorithms for the Internet traffic classification based on flow statistical properties. They analysed some limitations of unsupervised approaches (K-mean, Expectation maximization-EM) and supervised approaches (Decision tree as ID3, C4.5). After that, they proposed a new technique for to improve the effectiveness of these algorithms by combining unsupervised learning and supervised learning approaches. This method groups flows into clusters then using supervised learning to train traffic classifier according to what was found in the clustering step.

Regarding to the methods for handing imbalanced problem in traffic data, Vu et al. have recently applied several methods for addressing this problem [17]. They investigated a number of techniques for addressing imbalanced data problem including Under-sampling method, Over-sampling method, Synthetic Minority Over-sampling Technique (SMOTE), Condensed Nearest Neighbour, SMOTE combining with SVM. Their work showed that using the techniques for addressing imbalanced data is essential for the performance of supervised learning algorithms in network traffic classification problem.

In machine learning, deep learning has achieved remarkable results in a large number of applications [13]. Recently, deep learning has also extended to network traffic analysis [20]. In this research [20], Wang et al. have attempted to use Stacked Auto Encoder to learn the features of packet payload. Their research proved that Artificial Neural Networks and Deep learning (in their case that is the Stacked Auto Encoder) can be used to extract meaningful features from network traffic. This work has paved for the potential research in network traffic analysis using deep learning algorithms.

In this paper, we proposed the usage of a conditional Generative Adversarial Network called AC-GAN (Auxiliary Classifier Generative Adversarial Network) [16] for generating synthesized data samples. The objective of generating synthesized data is to enrich the training dataset and handling its imbalanced property. In two recent publications [5, 9], conditional Generative Adversarial Network has been shown being able to produce convincing samples on datasets with low variability and low resolution for images. Thus, we hypothesize that the training dataset augmented by synthesized samples from AC-GAN will help supervised algorithms (SVM, Decision Tree and Random Forest) to achieve better results in classifying network traffic. The detailed description of the deep network used in this paper (AC-GAN) will be presented in the next section.

3 OVERVIEW OF GENERATIVE ADVERSARIAL NETWORK

This section describes in detailed Generative Adversarial Network and one of its extension that was used in our research: Auxiliary Classifier Generative Adversarial Network.

3.1 Generative Adversarial Network

Generative Adversarial Network (GAN) was proposed by GoodFellow et al. in [10] for unsupervised learning. A GAN has two neuron networks which are trained in an opposition way. The first neuron network is a Generator (G) and the second neuron network is a Discriminator (D). The main idea behind GAN is to have two competing neural network models. The generator takes noise as input and generates samples. The discriminator receives samples from both the generator and the training data and attempt to distinguish between the two sources. These two networks play a continuous game, where the generator is learning to produce more and more realistic samples, and the discriminator is learning to get better and better at distinguishing the generated data from the real data. These two networks are trained simultaneously, and hope that the competition will drive the generated samples to be indistinguishable from the real data.

The input of the generator (G) is a vector of random noise z and it outputs a synthesized sample $X_{fake} = G(z)$. Network D takes the input of a real data sample or a synthesized sample from the generator and the output is a probability distribution $P(S|X) = D(X)$ over possible sources. Discriminator D is trained to maximize the log-likelihood to assigns the correct label (Equation 1) while Generator G is trained to minimize the second term in this equation.

$$L = E[\log P(S = real|X_{real})] + E[\log P(S = fake|X_{fake})] \tag{1}$$

One attractive property of GAN is its ability being trained in unsupervised mode and then reuse parts of its generator and discriminator networks as feature extractors for supervised tasks. However, GAN has been known to be difficult to train and often resulting in the generator that produce

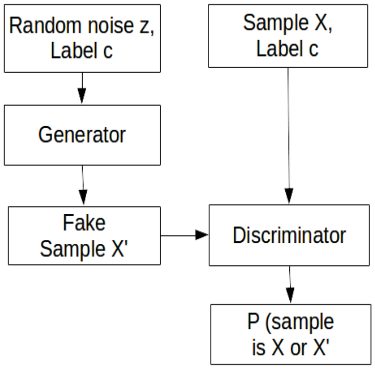

Figure 1: Training process of AC-GAN model.

nonsensical outputs. One of the extension of GAN is a model called Auxiliary Classifier GAN (AC-GAN). These models are able to generate samples taking into account external information (class label). Class label information is used to force G to generate a particular type of output.

3.2 Auxiliary Classifier Generative Adversarial Network

Odena et al. [16] proposed a variant of GAN model called auxiliary classifier GAN (AC-GAN). Figure 1 describes the process of training of AC-GAN model. The difference of AC-GAN and GAN is that the input of Generator G of AC-GAN includes a noise z and a class label c. In other words, the synthesized sample of G is $X_{fake} = G(c, z)$. Another different property is the output of Discriminator D including a probability distribution over sources L_S (Equation 2) and over class labels L_C (Equation 3).

$$L_S = E[\log P(S = real|X_{real})] + E[\log P(S = fake|X_{fake})] \tag{2}$$

$$L_C = E[\log P(C = c|X_{real})] + E[\log P(C = c|X_{fake})] \tag{3}$$

Discriminator D is trained to maximized $L_S + L_C$ and Generator G is trained to maximized $L_C - L_S$. This architecture is not exceedingly different from GAN but it is meaningful in generating new samples for a specific desired class.

In the training process described in Figure 1, the generator takes inputs as random noise z and label c and gives the output of fake samples while the discriminator has inputs as a real sample data or a fake sample data. The objective of training discriminator is making the probability of a sample data being a real sample or a fake sample closely to 0 or 1. This means that the discriminator is able to distinguish real samples and fake samples. However, the aim of training G network is generating samples closely to real samples in the dataset or the output probability of D equally to 0.5. In our experiment, G and D networks have two hidden layers and set the learning rate as $1e - 3$.

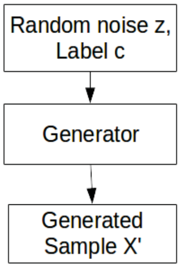

Figure 2: Process of generating samples.

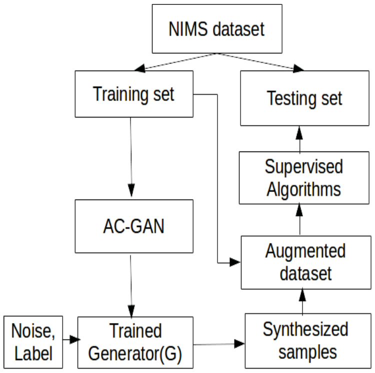

Figure 3: Process of our method.

4 METHODOLOGY

The flow of our method is described in Figure 3. It is detailed as followings:

First, the original dataset (NIMS dataset) is divided into two parts: training set and testing set. The training set is used to train AC-GAN model. After finishing the training process, we use the trained generator (G) to generate synthesized data samples with specific labels. Whenever a new synthesized sample needs to be created, a random noise is generated and a class label is selected. Thus, using the trained G network can handle the imbalance problem of network datasets by generating minor class samples. The synthesized samples are then combined with the training dataset to form the augmented dataset. After that, the augmented dataset is used to train several supervised classification algorithms (SVM, DT and RF). Finally, the supervised classification algorithms (after trained on the augmented dataset) are tested on the testing set.

5 EXPERIMENTAL SETTINGS

This section presents the dataset and the performance metrics used in this paper.

5.1 Dataset

In order to test the effectiveness of the proposed method we used a well-known network traffic dataset: Network Information Management and Security Group (NIMS) dataset [6]. This is the traffic dataset collected from the internal network with many applications of Dalhousie University Computing and Information Services Centre (UCIS) in 2007 on the campus network between the university and the Internet. In NIMS dataset, both SSH traffic and non-SSH traffic are generated from the applications. There are six SSH services as Shell login; X11; Local tunneling; Remote tunneling; SCP; and SFTP. Rest of applications are non-SSH traffic including DNS, HTTP, FTP, P2P (limewire),and telnet. In this dataset, SSH traffic is generated by SSH connections from client computers to four SSH servers outside of Dalhousie network.

In this paper, we aim to classify the SSH traffic from non-SSH traffic in NIMS. Therefore, we re-label NIMS dataset in two classes as SSH and non-SSH. Totally, NIMS dataset includes 35454 SSH flows and 678396 non-SSH flows and the ratio of SSH and non-SSH flows is about 0.052.

NIMS dataset groups packets into flows based on the statistical features. Traffic flows are defined by the sequence of packets that have same five tuples as source IP address, destination IP address, source port, destination port, and protocol type [11]. Each flow is described by 22 statistical features [6] shown in Table 1.

We divided NIMS into two parts: a half of samples for training set and rest for testing set described in Table 2. We used the training set to train AC-GAN. Then, the generator (after training) is used to generate new synthesized samples. In order to address the imbalanced problem in the training set, we generated more synthesized samples for SSH class and less synthesized sample for non-SSH class. The number of synthesized samples generated in each class, the training, testing set and the augmented data are presented in Table 2.

5.2 Evaluation Measures

We used three popular performance metrics in classification problem to measure the impact of our method. The reported performance metrics include accuracy score, F1 score and AUC score[12]. Accuracy score measures how a classifier making correct predictions and this is calculated in Equation 4.

$$Acuracy = \frac{N_{correct}}{N_{total}} \tag{4}$$

where $N_{correct}$ and N_{total} are the number of correct predictions and the total number of predicted samples. The advantage of this metric is that it is very intuitive and easy to implement. However, it makes no distinction between classes that is sometime not enough to measure a classifier especially for imbalanced dataset.

The second metric is F1 score. This metric overcomes the disadvantage of Accuracy score. F1 score is calculated based on two other metrics: Precision and Recall. Precision metric measures the ability of classifier that predicts positive samples as positive. Recall score measures how many actual positive observations are predicted correctly. F1 score is Harmonic mean [12] of Precision and Recall where Harmonic mean is an appropriate way to average ratios. Precisely, F1 score is computed in Equation 5.

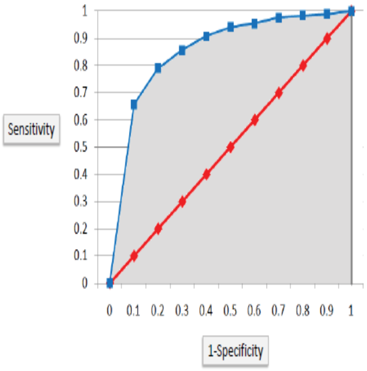

Figure 4: Demonstration of AUC score [2].

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{5}$$

The last metric is AUC score which stands for Area Under ROC Curve [12].The ROC curve is created by plotting the true positive rate (sensitivity) against the false positive rate (1-specificity) at various threshold settings (Figure 4). The ROC curve is the sensitivity as a function of 1-specificity. In general, if the probability distributions for both detection and false alarm are known, the ROC curve can be generated by plotting the cumulative distribution function of the detection probability in the y-axis versus the cumulative distribution function of the false-alarm probability on the x-axis. The space under ROC curve is represented as AUC score. This measures the average quality of classification model at different threshold. A random classifier has AUC value of 0.5 and the value of AUC score for a perfect classifier is 1.0. Therefore, almost classifiers have the value of AUC score between 0.5 and 1.0. For all three above performance metrics, larger values present better performance of an algorithm.

6 RESULTS AND DISCUSSION

We divided our experiments into two sets. In the first set, we aim to compare the performance of some popular classification algorithms (SVM, DT, and RF) when they are trained on two datasets (the original dataset and the augmented dataset that is generated by AC-GAN). In the second set, we compare the impact of the synthesized data generated by AC-GAN with the synthesized data generated by the best method in a recent research [17].

For all tested classification algorithms including SVM, DT, and RF, we used their implementation in a popular machine learning packet in python: Scikit learn [4]. In order to lessen the impact of selecting parameters to the performance of these algorithms, we used grid search technique to search the

Table 1: Description of statistical feature for network flow

Index	Feature name	Abbreviation
1	min forward packet length	$min_f pktl$
2	mean forward packet length	$mean_f pktl$
3	max forward packet length	$max_f pktl$
4	std dev forward packet length	$std_f pktl$
5	min backward packet length	$min_b pktl$
6	mean backward packet length	$mean_b pktl$
7	max backward packet length	$max_b pktl$
8	std dev backward packet length	$std_b pktl$
9	min forward inter arrival time	$min_f iat$
10	mean forward inter arrival time	$mean_f iat$
11	max forward inter arrival time	$max_f iat$
12	std dev forward inter arrival time	$std_f iat$
13	min backward inter arrival time	$min_b iat$
14	mean backward inter arrival time	$mean_b iat$
15	max backward inter arrival time	$max_b iat$
16	std dev backward inter arrival time	$std_b iat$
17	duration of the flow	$duration$
18	protocol (tcp, udp)	$proto$
19	total forward packets	$total_f packets$
20	total forward volume	$total_f volume$
21	total backward packets	$total_b packets$
22	total backward volume	$total_b volume$

Table 2: Number of data samples

Datasets	SSH	nonSSH
Original training set	17736	339189
Testing dataset	17718	339207
Synthesized samples	300000	20000
Augmented dataset	317736	359189

Table 3: Parameter range of grid search for classifiers

Classifiers	Parameters
SVM	$kernel = rbf, linear; gama = 0.001, 0.01, 0.1, 1.0$
DT	$max - depth = 5, 6, 7, 8, 9, 10$
RF	$n - estimators = 20, 40, 80, 150$

Table 4: Accuracy, F1, AUC score of our experiments

Algorithms	Acuracy score	F1 score	AUC score
SVM	0.9873	0.5909	0.7096
DT	0.9976	0.9482	0.9495
RF	0.9978	0.9485	0.9536
SVM+AC-GAN	0.9878	0.6050	0.7159
DT + AC-GAN	0.9978	0.9552	0.9501
RF+AC-GAN	**0.9989**	**0.9543**	**0.9565**

best value of the parameters for each algorithms. The range of values used in the grid is presented in Table 3.

The results of the first experimental set are presented in Table 4. This table present accuracy score, F1 score and AUC score of SVM, DT, RF on the testing dataset. It should be noted that two versions of the classification algorithms (one trained on the original training dataset and another trained on the augmented dataset (shorted with +AC-GAN at the end)) are compared in this table. It can be seen that, the performance of all three classification algorithms are improved when they are trained on the augmented dataset. Moreover, the improvement of these classifiers when training the augmented dataset is more impressive regarding to F1 score and AUC score than with accuracy score.

The characterization of network traffic datasets is that they have more categorical features (13% categorical features in NIMS dataset) and lower dimensional space (22 dimensional space of a sample for NIMS dataset) comparing to other kinds of dataset such as image datasets (no categorical features and 784 dimensional space of a sample for MNIST dataset [1]), text datasets (no categorical features and 4702 dimensional space of a sample for DBWorld e-mails dataset [3]). Thus the algorithms based on decision tree likes DT, RF perform better on classifying network traffic than SVM does. Among three classification algorithms, the table shows that RF+AC-GAN achieved the best performance with respect to all three performance metrics.

The results of the second experimental set is presented in Table 5. We examine RF on three augmented datasets to get the accuracy, F1, and AUC score with the computation time to finish balancing NIMS dataset. In this table, we compare the performance of the best classification algorithm in the first experiment (RF) when this algorithm is trained on three augmented dataset. The first augmented dataset is generated by the best technique for generating network traffic data for addressing imbalanced problem in network traffic data according a recent research [17]. This technique is called SMOTE-SVM [17]. The detailed description of SMOTE-SVM can be found in [17]. The second augmented dataset is synthesized by the ensemble BalanceCascade technique which is the best method to handle imbalanced dataset that is created from 10000 random samples with two classes [15]. The third augmented dataset is generated by the approach proposed in this paper: AC-GAN. It can be observed that the performance of RF trained on AC-GAN is better than its performance trained on SMOTE-SVM. Particularly, F1 score and AUC score of classifier on augmented dataset by using AC-GAN is clearly higher than those values of SMOTE-SVM. In fact, these values of RF on AC-GAN augmented dataset is greater than those of SMOTE-SVM from 1% to 1.5%. The augmented dataset generated by BalanceCascade technique has similar accuracy of the classifier but lower performance in computation time comparing with AC-GAN.

Overall, the results in this section show that using AC-GAN to generate synthesized data for network traffic classification problem helps improving the performance of supervised learning algorithms in solving compared to using synthesized data and some recent techniques for handling imbalanced data such as SMOTE-SVM [17],BalanceCascade [15].

7 CONCLUSIONS AND FUTURE WORK

Enriching and handling imbalanced dataset is one way to improve the performance of classification problems. In this paper, we introduced an application of a deep learning model, AC-GAN, to address the imbalanced problem in network traffic analysis. We used AC-GAN model to synthesize network traffic data samples for a well-known traffic dataset, NIMS. After that, the synthesized samples were combined with the original training dataset to form a new augmented dataset. Three supervised learning algorithms (SVM, DT and RF) were trained on the augmented dataset.

Experimental results shows that SVM, DT, RF achieved better performance with respect to three measure metrics when that were trained on the augmented dataset compared to when they were trained on the original dataset. Moreover, the synthesized data generated by AC-GAN was also better than the synthesized data generated by the best method in a recent publication, SMOTE-SVM [17].

In the future, we would like to investigate the ability of classification algorithms to predict samples from new classes when they are trained on the synthesized data generated by AC-GAN. Moreover, we also want to examine and apply recent advance in deep learning to improve the results of machine learning algorithm in network traffic analysis.

ACKNOWLEDGMENT

The work in this paper was funded by The Vietnam National Foundation for Science and Technology Development (NAFOSTED), under grant number 102.01-2014.09.

REFERENCES

[1] 1998. The MNIST dataset of handwritten digits. (1998). Retrieved October 24, 2017 from http://yann.lecun.com/exdb/mnist/

[2] 2010. Model Evaluation - Classification. (2010). Retrieved September 14 2017 from http://chem-eng.utoronto.ca/~datamining/dmc/model_evaluation_c.htm

[3] 2011. DBWorld e-mails Data Set. (2011). Retrieved October 24, 2017 from https://archive.ics.uci.edu/ml/datasets/DBWorld+e-mails#

[4] 2014. Scikit-learn tutorial. (2014). Retrieved September 14 2017 from http://scikit-learn.org/stable/

[5] Alec, Radford, Luke Metz, and Soumith Chintala. 2016. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. In *Computer Vision and Pattern Regconition (CVPR)*. CoRR. https://arxiv.org/abs/1506.05751

[6] Riyad Alshammari and A. Nur Zincir-Heywood. 2010. Can encrypted traffic be identified without port numbers, IP addresses and payload inspection? *Elsevier* 22, 2 (2010), 1326–1348.

[7] Arthur Callado, Carlos Kamienski, Stênio Fernandes, and Djamel Sadok. 2008. A Survey on Internet Traffic Identification and Classification. (2008).

[8] Cisco Systems and Inc. 2008. *Cisco WAN ans application optimization solution guide*. Tech. Rep.. (2008).

[9] Emily Denton, Soumith Chintala, Arthur Szlam, and Rob Fergus. 2015. Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks. In *Computer Vision and Pattern Regconition (CVPR)*. CoRR. https://arxiv.org/abs/1506.05751

[10] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and

Table 5: Comparing effective of AC-GAN and SMOTE-SVM

Algorithms	Acuracy score	F1 score	AUC score	Computation time (second)
SMOTE-SVM	0.9920±0.0029	0.9494±0.0032	0.9412±0.0010	2480
BalanceCascade	**0.9990**±0.0084	0.9520±0.0065	0.9510±0.0078	3876
AC-GAN	0.9980±0.0013	**0.9523**±0.0018	**0.9545**±0.0041	**1256**

Yoshua Bengio. 2014. Generative Adversarial Networks. In *Neural Information Processing Systems*. Montreal, Canada. https://arxiv.org/pdf/1406.2661.pdf

[11] G.P.S. Junior, J.E.B. Maia, R. Holanda, and J. N. de Sousa. 2007. P2P Traffic Identification using Cluster Analysis. *2007 First International Global Information Infrastructure Symposium* (2007), 128–133.

[12] Olson David L and Delen Dursun. 2008. Advanced Data Mining Techniques, 1st edition (Ed.). Springer, 138.

[13] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature* 521 (2015), 436–444. Issue 7553.

[14] Jun Li, Shunyi Zhang, Yanqing Lu, and Junrong Yan. 2008. Real time P2P traffic identification. In *Global Telecommunications Conference*. IEEE, New Orleans, LO, USA. https://doi.org/10.1109/GLOCOM.2008.ECP.475

[15] A. More. 2016. Survey of resampling techniques for improving classification performance in unbalanced datasets. *ArXiv e-prints* (Aug. 2016). arXiv:stat.AP/1608.06048

[16] Augustus Odena, Christopher Olah, and Jonathon Shlens. 2017. Conditional Image Synthesis With Auxiliary Classifier GANs. In *Neural Information Processing Systems*. Long Beach, CA, USA. https://arxiv.org/abs/1610.09585

[17] Ly Vu Thi, Dong Van Tra, and Quang Uy Nguyen. 2016. Learning from Imbalanced Data for Encrypted Traffic Identification Problem. SoICT, Ho Chi Minh, Vietnam.

[18] Joe Touch, Eliot Lear, Allison Mankin, Markku Kojo, Kumiko Ono, Martin Stiemerling, Lars Eggert, Alexey Melnikov, Wes Eddy, Alexander Zimmermann, Brian Trammell, and Jana Iyengar. 2017. Service Name and Transport Protocol Port Number Registry. In *The Internet Assigned Numbers Authority (IANA)*.

[19] Alina Vladutu, Drago Comaneci, and Ciprian Dobre. 2015. Internet Traffic Classification based on Flows Statistical Properties with Machine Learning. In *International Journal of Network Management*. Wiley InterScience. https://doi.org/10.1002/nem

[20] Zhanyi Wang. 2015. The applications of Deep Learning on Traffic Identification. In *Black Hat*. Lasvegas, USA.

[21] Young J Won, Seong Chul Hong, Byung Chul Park, and James W. Hong. 2008. Towards automated application signature generation for traffic identification. In *Network Operations and Managements Symposium (NOMS)*. IEEE, 160–167.

[22] Jun Zhang, XiaoChen, YangXiang, Wanlei Zhou, and JieWu. 2014. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. In *IEEE/ACM transaction on networking (1063-6692)*. IEEE. https://pdfs.semanticscholar.org/763d/7955354cd0fbd6b2983ef46880dcc21533f6.pdf