Learning from Imbalanced Data for Encrypted Traffic Identification Problem

Ly Vu
Le Quy Don University
Ha noi, Vietnam
lyvt@mta.edu.com

Dong Van Tra
Technical Economic College
Ho Chi Minh City, Vietnam
trvdong@gmail.com

Quang Uy Nguyen
Le Quy Don University
Ha noi, Vietnam
quanguyhn@gmail.com

ABSTRACT

Identifying encrypted application traffic is an important issue for many network tasks including quality of service, firewall enforcement and security. One of the challenging problems of classifying encrypted application traffic is the imbalanced property of network data. Usually, the amount of unencrypted traffic is much higher than the amount of encrypted traffic. To date, the machine learning based approach for identifying encrypted traffic often solely focused on examining and improving algorithms. The techniques for addressing imbalanced data are rarely investigated. In this paper, we present a thorough analysis of the impact of various techniques for handling imbalanced data when machine learning approaches are applied to identifying encrypted traffic. The experiments are conducted on a well-known network traffic dataset and the results showed that some techniques for addressing imbalanced data help machine learning algorithms to achieve better performance.

CCS Concepts

•**Computer systems organization** → **Embedded systems;** *Redundancy;* Robotics; •**Networks** → Network reliability;

Keywords

Machine learning; Encrypted Network Traffic; Imbalanced Data

1. INTRODUCTION

In network analysis, traffic classification is a crucial requirement for administrators to effectively manage their network bandwidth and resource. Moreover, traffic classification is important for security applications since it helps to assess the security threats to the network. In traffic classification, the identification of encrypted traffic represents a first step in identifying malicious behaviours since most of the time, users with malicious intentions try to hide their behaviour either in encrypted or covert tunnels.

Traditionally, three are two approaches used for classifying network traffic [20]: the first approach is to use well-known port numbers (visible in TCP or UDP headers) while the second approach is based on Deep Packet Inspection (DPI) to look for specific protocol signatures. However, these approaches present some disadvantages in the modern networks. The first approach assumes that most application always uses well-known port numbers. This assumption becomes increasingly inaccurate when applications use non-standard ports to bypass firewalls or circumvent operating system restrictions. The second approach assumes that the payload of every packet is available. However, this assumption is not always true since the access to payload maybe restricted due to the violation of the organizational privacy policies. Moreover, examining the payload of a packet at the network speed is a computationally expensive. Consequently, other techniques are required to increase the accuracy of network traffic classification.

Recently, many studies have attempted to employ machine learning techniques that use statistical flow information (features) for network traffic classification. Such features are extracted from the information on the transport layer, which does not depend on port numbers or payload inspections. The previous results of using machine learning methods for classifying encrypted applications are promising [2, 4, 1, 18, 6]. However, one of the difficulty when applying machine learning techniques to identifying encrypted traffic is the imbalanced structure of the network data. Since there are many applications running simultaneously on the network, the portion of encrypted traffic is often tiny compared to the huge amount of unencrypted traffic. Thus, the approach for handling imbalanced data plays a crucial role in the performance of machine learning techniques.

To date, the research in classifying encrypted traffic has focused on testing the efficiency and effectiveness of classifiers. Different research groups have employed various machine learning techniques such as Hidden Markov model, Naive Bayesian model and Decision Tree [1, 4, 6, 18]. However, there has not been any research on examining the effect of the methods for addressing imbalanced data. This paper is the first attempt to systematically investigate the impact of the techniques for handling imbalanced data when machine learning techniques are used for identifying encrypted traffic. The rest of this paper is organized as follows. Section 2 presents the related work in identifying encrypted network traffic. The methods for handling imbalance data are shown

SoICT '16, December 08 - 09, 2016, Ho Chi Minh City, Viet Nam

DOI: http://dx.doi.org/10.1145/3011077.3011132

in Section 3. Section 4 describes the experimental settings. The result of our experiments are presented and discussed in Section 5. Finally, Section 6 concludes the paper.

2. RELATED WORK

Three have been a number of the previous work that attempted to identify encrypted network traffic. However, the existing work usually focused only on testing various machine learning algorithms and ignored to consider the imbalanced structure of the data set.

Riyad et al. [1] assessed the performance of some machine learning algorithms including Adaboost, Support Vector Machine, Naive Bayesian, and Decision Tree (C4.5) in identifying encrypted network traffic. Among the tested algorithms, only Adaboost [19] method can handle imbalanced data. The experimental results in [1] proved that Adaboost algorithm has good generalization ability but sensitive to the stopping criterion. Moreover, the decision tree classifier (C4.5) is better than other classifiers when used for classifying encrypted traffic.

After that, Carlos et al. [4] examined the performance of unsupervised learning techniques for classifying encrypted network traffic. They compared five unsupervised clustering algorithms including basic K-Means, semi-supervised K-Means, Density-based spatial clustering of applications with noise (DBSCAN), expectation-maximization (EM) and Multi-Objective Genetic Algorithm (MOGA). The results showed that their proposed clustering algorithm (MOGA) outperforms other algorithms in classifying encrypted network traffic.

Recently, JZigang Cao et al. [6] conducted an analysis on the recent advances and challenges in encrypted network identification. More recently, Petr Velan et al. [18] described a survey of encrypted network traffic classification and analysis. They described in detailed the structure of encrypted network traffic and some tools for classifying network traffic. They compared the performance of a large number of classifiers including Markov models, Repeated Incremental Pruning to Produce Error Reduction (RIPPER), AdaBoost, SVMs, C4.5, K-mean, K-nearest neighbours and MOGA.

In machine learning, there has been a large number of researches that afforded to address imbalanced data [16, 9, 7, 12, 3, 8, 17]. Generally, there are three approaches for handling imbalanced data: Modifying the objective cost function, sampling and generating artificial data. However, to the best of our knowledge, these techniques have not been applied and investigated in encrypted network traffic classification. In the next section, we will present the techniques for handling imbalanced data that are used in our paper.

3. METHODS

This section presents three techniques for addressing imbalanced data that are used in this paper: Modifying the objective cost function, under-sample and over-sampling, and generating artificial data. These techniques will be used to prepare the training data for two machine learning techniques: Classification and Regression Trees (CART)[5] and Random forest (RF)[15]. We used CART and RF algorithms in this paper since they have been showed the good performance on classifying encrypted network traffic.

The Figure 1 presents the overview our system. The first step was to collect data including encrypted traffic (SSH traffic) and non-encrypted traffic. The feature extraction step was then applied to the collected traffic to obtain 22 features of each traffic (the extracted features are shown in the Section 4). After that,there techniques for handling imbalanced data was used to prepare the training data for the machine learning algorithms. Finally, the performance of two machine learning algorithms (CART and RF) was reported as the indicator for the goodness of the techniques for addressing imbalanced data.

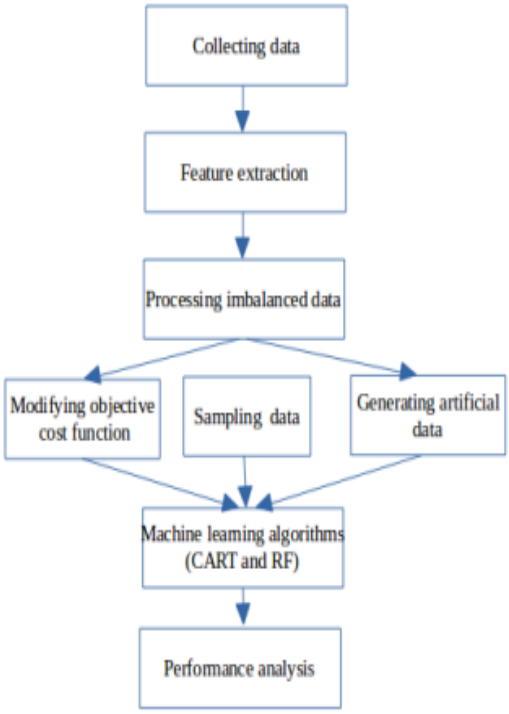

Figure 1: The overview of our system.

3.1 Modifying the Objective Cost Function

The method of modifying objective cost function is based on weighting differently the samples in minor and major classes. This method gives higher score on the minor samples to penalize more intensely on miss-classifying of the sample in the minor class. If the algorithm miss-classifies the samples in the minor class, the heavier penalty is set on that. In our experiments, the weighting value for the minor class is M_a/N and the weight on the major class is M_i/N where M_a and M_i is the number of major and minor samples and N is total samples in the dataset.

3.2 Under Sampling Methods

The objective of the under-sampling method is to reduce the size of major class by removing some major instances. Two under sampling methods used in this paper are: Random under sampling and Condensed Nearest Neighbour.

Random under sampling - (RUS): RUS works by randomly removing some instances in major class(see Algorithm 1 for the more detailed). As shown in [10], RUS is an efficient sampling method to deal with imbalanced class classification problems. Moreover, RUS reduces the data size leading

increasing the speed of classification algorithms. However, the drawback is that it may removes some potentially useful data.

Algorithm 1 Random under sampling

INPUT:
X: original training set
OUTPUT:
X_{new}: new sampling set
BEGIN:
$X_{new} = X$
while number of majority and minority samples is NOT balance **do**
  Generate a random number id
  if $X[id]$ is majority samples **then**
    $X_{new} = X_{new} \setminus X[id]$
  end if
end while
END.

Condensed Nearest Neighbour (CNN): CNN was first introduced by Hart [12] that generates a consistent subset of an original sample set. In this algorithm, the original sample set is divided into two sets as S and T. Initially, the S set has one sample that is randomly selected from the original set and the remaining samples are in the T set. CNN then scans all samples of T and adds to S if that sample is misclassified by the content of S. The algorithm scans T as many times as necessary until no sample transfer from T to S. In CNN, the misclassified data lies close to the decision boundary as shown in Algorithm 2. The disadvantage of CNN is that it is very computational expensive since the algorithm needs to repeatedly scan the T set.

Algorithm 2 Condensed Nearest Neighbour

INPUT:
X: original training set
C: Classifier
OUTPUT:
Consistent subset S of T
BEGIN:
$S = X[0]$
$T = X \setminus S$
while Having samples in T transfer to S **do**
  for all samples $x[i] \in T$ **do**
    Use classifier C in S
    if C can not classified $x[i]$ **then**
      $S = S \cup x[i]$
    end if
  end for
end while
END

3.3 Over Sampling Method

The purpose of the over sampling method is to raise the samples in the minor class. *The random over sampling (ROS)*[13] is used in this paper. ROS generates some copies of the samples of the minor class as shown in Algorithm 3. This method is easy to implement and has low computational cost. The main drawback of ROS is overfitting problem as generating same copies of the minor class. Moreover, due to the huge size of major class, ROS can make dataset extremely large.

Algorithm 3 Random oversampling

INPUT:
X: original training set
OUTPUT:
X_{new}: new sampling set
BEGIN:
while number of minority and majority samples is NOT balance **do**
  Generate a random number id
  if $X[id]$ is minority samples **then**
    $x_{new} = X[i]$
    $X_{new} = X \cup \{x_{new}\}$
  end if
end while
END.

3.4 Generating Artificial Data

The first method for generating artificial data used in this paper is *Synthetic Minority Over-sampling Technique (SMOTE)*. The SMOTE was first introduced by Nitesh et al. [8] where the authors proposed an over-sampling technique in which minority samples are generated by "synthetic" samples rather than replicating samples. Synthetic samples are samples that are generated by operating on the feature space of that samples and its k-nearest neighbours where k is chosen based on the amount of minority samples required. In detail, to create synthetic samples, let d_i vector be the different of feature vector of minority sample x_i and it's k-nearest neighbours and let $d'_i = d_i * r$ where r is random number in [0, 1]. A new sample $x'_i = x_i + d'_i$ is generated.

After SMOTE was proposed, there has been a number of improved versions. Nguyen et al. [17] proposed a new technique to improve SMOTE [8]. The idea is that sampling on the entire of the minority class is less important than generating samples along the decision boundary. In this approach, they used support vectors obtained by training a Support Vector Machines (SVMs) classifier on the original training set. New samples are generated by combining each minority class support vector with its nearest neighbours using interpolation or extrapolation technique based on the density of majority samples in its nearest neighbours. This technique is referred to as SMOTE-SVM and the its details is described as in Algorithm 4.

4. EXPERIMENTAL SETTINGS

The experiments were conducted on the Network Information Management and Security Group (NIMS) dataset [2]. The NIMS dataset consists of packets collected from the internal network of Dalhousie University Computing and Information Services Centre (UCIS) in 2007. The collected traffic is the network traffic between the university and the commercial Internet where different network models are simulated with many applications. There are six encrypted services as Shell login; X11; Local tunneling; Remote tunneling; SCP; and SFTP. Some unencrypted applications are also emulated in this network such as DNS, HTTP, FTP, P2P (limewire),and telnet. In totally, the NIMS dataset includes 14,681 encrypted flows and 699,170 unencrypted flows and the ratio of encrypted to unencrypted flows is about

Algorithm 4 SMOTE-SVM borderline oversampling

INPUT:
X: original training set
N: number of new sampling set
m: number of nearest neighbours
nn: nearest neighbour vector
β: coefficient of extrapolate/interpolate function
OUTPUT:
X_{new}: new sampling set
BEGIN:
- Run SVMs classifier on X to have support vector index id and $SVs = X[id]$ is the set of support vectors
- Set $T = \frac{N}{100} + |X|$ is the number of new samples
- Set $k = \frac{N}{100}$ is the number of nearest neighbours of minority class
- Distributing T satisfy that each $sv_i \in SVs$, generate $quantity[i]$ samples
for each $sv_i \in SVs$ **do**
  - Compute m nearest neighbours in X
  - Generate $quantity[i]$ new samples by following ways:
  if there are less $\frac{m}{2}$ is minority samples **then**
    $x_{new} = sv_i + \beta(sv_i - nn_{i,j})$ {extrapolate samples}
  else
    $x_{new} = sv_i + \beta(nn_{i,j} - sv_i)$ {interpolate samples}
  end if
end for
return $X_{new} = X \cup \{x_{new}\}$
END.

0.021. Thus, this dataset can be considered as an imbalanced dataset.

Traffic flows are defined by the sequence of packets that have same five tuples including the source IP address, the destination IP address, the source port number, the destination port number, and protocol type [14]. Each flow is described by 22 statistical features [2] as shown in Table 1.

We randomly selected 50% data samples for training and the rest for testing. The techniques of sampling and generating artificial data were then applied to the training data to reduce the number of over-class samples or to increase the number of minor-class samples. The number of unencrypted samples (major-class) after applying under-sample techniques (RUS and CNN) and the number of encrypted samples (minor-class) after applying over-sampling technique (ROS) and generating artificial data techniques (SMOTE and SMOTE-SVM) are presented in Table 2. In the table, the number of the original encrypted and unencrypted samples are shown in the first two rows. It can be seen that, after applying the sampling and generating artificial data techniques, the number of the samples in two classes are roughly equal.

Table 3: Confusion matric of encrypted traffic identification

	Positive Prediction	Negative Prediction
Positive class	True Positive (TP)	False Negative (FN)
Negative class	False Positive (FP)	True Negative (TN)

For each classification algorithm (CART and RF), we performed five set of experiments including (1) no technique for imbalanced data (No-handling); (2) the objective cost function modification (Cost-function); (3) random under sampling technique (RUS); (4) Condense Nearest Neighbours (CNN); (4) random oversampling technique (ROS); and (5) generating data technique (SMOTE and SMOTE-SVM). For all algorithms, we used the their implementation in the Scikit learn machine learning packet. Scikit learn is a popular machine learning packet in Python [11]. The default parameters of all algorithms in Scikit learn packet were selected.

To evaluate the performance of the tested methods, we used the confusion matrix described in Table 3. For the binary classifier, four possible outcomes are possible. Encrypted flow correctly detected as encrypted flow (TP), or incorrectly predicted as unencrypted (FN). The unencrypted traffic correctly predicted as unencrypted traffic (TN), or incorrectly predicted as encrypted traffic (FP). From the results in the confusion matrix, we measure the detection rate (DR) as the prediction probability of the encrypted traffic correctly and the false alarm rate (FAR) as the detection probability of the unencrypted traffic incorrectly. These measures are defined in the equations 1 and 2. The best classifier is the algorithm which achieves the highest DR value the lowest FAR value.

Table 1: Statistical features for network flow

Index	Feature name	Abbreviation
1	min forward packet length	min_fpktl
2	mean forward packet length	$mean_fpktl$
3	max forward packet length	max_fpktl
4	std dev forward packet length	std_fpktl
5	min backward packet length	min_bpktl
6	mean backward packet length	$mean_bpktl$
7	max backward packet length	max_bpktl
8	std dev backward packet length	std_bpktl
9	min forward inter arrival time	min_fiat
10	mean forward inter arrival time	$mean_fiat$
11	max forward inter arrival time	max_fiat
12	std dev forward inter arrival time	std_fiat
13	min backward inter arrival time	min_biat
14	mean backward inter arrival time	$mean_biat$
15	max backward inter arrival time	max_biat
16	std dev backward inter arrival time	std_biat
17	duration of the flow	$duration$
18	protocol (tcp, udp)	$proto$
19	total forward packets	$total_fpackets$
20	total forward volume	$total_fvolume$
21	total backward packets	$total_bpackets$
22	total backward volume	$total_bvolume$

Table 2: The number of unencrypted samples in RUS and CNN and the number of encrypted samples in ROS, SMOTE and SMOTE-SVM

Methods	Number of samples
Original-encrypted	17725
Original-unencrypted	339200
RUS	35450
CNN	18216
ROS	321425
SMOTE	321213
SMOTE-SVM	320888

$$DR = \frac{TP}{TP + FN} \quad (1)$$

$$FAR = \frac{FP}{FP + TN} \quad (2)$$

5. RESULTS AND DISCUSSION

This section presents the results of our experiments. Each experiment was performed ten times. The results were then averaged over ten runs. The table 4 represents the average values and the variance of DR and FAR of the CART classifier with and without using method for handling imbalanced data techniques. It can be seen from this table that most algorithm achieved genuinely small value of FAR. This is understandable since the value for FAR is calculated relying on the ability of false prediction of majority class. With the huge amount of unencrypted traffic and tiny portion of encrypted traffic in the dataset, the ratio of false prediction to total number of unencrypted traffic is very small.

Moreover, this table shows that using techniques for handling imbalanced data is promising in identifying encrypted traffic. Overall, when the CART algorithm combining with preprocessing imbalanced data, the DR value was increased from around 1-7%. More precisely, when no techniques for addressing imbalanced data was used, the CART classifier achieved the DR value of 89,48%. When CART was incorporated with methods for imbalanced data such as CNN and SMOTE-SVM, the DR value is 95.07% and 96.12%, respectively. Other techniques including WS, RUS, ROS and SMOTE also improve the ability to predict the minor class but the improvement is only marginal. The effective techniques for handling imbalanced traffic data for the CART classifier are CNN and SMOTE-SVM where SMOTE-SVM is the best method with the DR value exceeding more 7% comparing to when only using the CART classifier.

The results of the RF classifier when combining with imbalanced data processing techniques are showed in Table 5. Similar to the CART algorithm, the value of FAR is very tiny. Particularly, some algorithms achieved the value of FAR at zero. This did not achieve with the CART classifier. Table 5 also presents the good ability of the RF classifier in classifying the encrypted network traffic. Comparing with the results in Table 4, most method for imbalanced data processing when combined with the RF classifier performed better than the CART classifier. SMOTE-SVM is the only technique that achieved lower value of DR when combined with the RF classifier. However, the value of FAR when SMOTE-SVM is combined with RF is much less than that its value when combined with the CART algorithm.

Table 6 presents the processing of the methods for addressing imbalanced data. Although the CNN and SMOTE-SVM has high detection rate, they also need much higher processing time compared to other. The reason is that CNN needs to scan all the remaining set of samples after each time taking one sample while in SMOTE-SVM method, to generate a new sample, the SVM classifier have to run many times to find the nearest neighbours for each support vectors. Therefore, using the CNN and SMOTE-SVM can achieve better performance with the sacrifice of the preprocessing time. However, the data pre-processing can be executed offline. Therefore, these methods are the good candidates when the accuracy of the detection is the main priority.

Overall, the results in this section prove that using techniques for addressing imbalanced data is important for the encrypted network traffic classification problem where the huge number of samples are collected and the encrypted traffic is much more rarely than normal traffic. For this type of data, an under sampling method such as CNN and a method for generating artificial samples (SMOTE-SVM) is the best solutions to pre-process data before using machine learning algorithms.

6. CONCLUSIONS AND FUTURE WORK

This paper presented an analysis of the impact of the techniques for handling imbalanced data in machine learning to encrypted traffic classification problem. Three methods for processing imbalanced data were investigated. The experiments were conducted on a well-known network traffic dataset. The results showed that using the approaches for handling imbalanced data is beneficial for machine learning when they are applied to solve this problem. In the future, we would like to investigate and develop better machine learning techniques and better techniques for addressing imbalanced data to improve the effectiveness of machine learning in identifying encrypted network traffic.

7. ACKNOWLEDGMENT

This research is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.01-2014.09.

8. REFERENCES

[1] R. Alshammari and A. N. Zincir-Heywood. Machine learning based encrypted traffic classification: Identifying ssh and skype. In *CISDA*, 2009.

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

[3] F. Angiulli. Fast condensed nearest neighbor rule. In *ACML*, 2005.

[4] C. Bacquet, K. Gumus, D. Tizer, A. N. Zincir-Heywood, and M. I. Heywood. A comparison of unsupervised learning techniques for encrypted traffic identification. 2010.

[5] L. Breiman, J. Friedman, C. J. Stone, and R. Olshen. *Classification and Regression trees*. ISBN 9780412048418, Chapman and Hall/CRC, 1984.

[6] J. Cao, G. Xiong, Y. Zhao, Z. Li, and L. Guo. A survey on encrypted traffic classification. In *Springer*, volume 490, pages 73–81. Communications in Computer and Information Science, 2014.

[7] L. Cao, J. Zhong, and Y. Feng. Cost sensitive classification in data mining. In *LNCS 6440 Part I*, 2010.

[8] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: Synthetic minority oversampling technique. *Journal of Artificial Intelligence Research 16*, pages 321–357, 2002.

[9] C. Chen, A. Liaw, and L. Breiman. Using random forest to learn imbalanced data. In *Technical report*, 2013.

Table 4: DR and FAR of CART classifier with imbalanced data algorithm

Algorithm	Detection rate	False alarm rate
No-handling	0.894767 ± 0.004151	0.000127 ± 0.000071
Cost-function	0.895466 ± 0.006869	0.000077 ± 0.000070
RUS	0.895023 ± 0.004435	0.000491 ± 0.000222
CNN	**0.950704 ± 0.021405**	**0.001521 ± 0.000749**
ROS	0.900564 ± 0.006856	0.000066 ± 0.000044
SMOTE	0.907942 ± 0.007321	0.000179 ± 0.000108
SMOTE-SVM	**0.961213 ± 0.009552**	**0.000188 ± 0.000172**

Table 5: DR and FAR of RF classifier with imbalanced data algorithm

Algorithm	Detection rate	False alarm rate
No-handling	0.901835 ± 0.005393	0 ± 0
Cost-function	0.903372 ± 0.003397	0 ± 0
RUS	0.910442 ± 0.007859	0.000003 ± 0.000002
CNN	**0.956010 ± 0.005974**	**0.000231 ± 0.000084**
ROS	0.903116 ± 0.002688	0 ± 0
SMOTE	0.911545 ± 0.008033	0 ± 0
SMOTE-SVM	**0.935711 ± 0.008103**	**0.000013 ± 0.000012**

Table 6: Processing time in seconds of each imbalanced method

Methods	Processing time (second)
No-handling	0
Cost-function	1.53
RUS	0.99
CNN	**886.77**
ROS	1.30
SMOTE	2.31
SMOTE-SVM	**2446.80**

[10] C. Drummond and R. C. Holte. C4.5, class imbalance, and cost sensitivity: Why under-sampling beats over-sampling. In *Working Notes of the ICML Workshop on Learning from Imbalanced Data Sets*, 2003.

[11] G. Hackeling. *Mastering Machine Learning with scikit-learn*. Published by Packt Publishing Ltd, Birmingham B3 2PB, UK, 2014.

[12] P. Hart. The condensed nearest neighbor rule. *IEEE Trans. on Inform. Th.*, (14):515–522, 1968.

[13] N. Japkowicz. Learning from imbalanced data sets: A comparison of various strategies. pages 10–15. AAAI Press, 2000.

[14] G. Junior, J. Maia, R. Holanda, and J. de Sousa. P2p traffic identification using cluster analysis. In *In Global Information Infrastructure Symposium*, pages 128–133. GIIS, 2007.

[15] L.Breiman. Random forests. 2001.

[16] R. Longadge, S. S. Dongre, and L. Malik. Class imbalance problem in data mining: Review. 2:355–374, 2013.

[17] H. M. Nguyen, E. W. Cooper, and K. Kamei. Borderline over-sampling for imbalanced data. In *Fifth International Workshop on Computational Intelligence and Applications IEEE*, 2009.

[18] P. Velan, M. Cermak, P. Celeda, and M. Drasar. A survey of methods for encrypted traffic classification and analysis. pages 355–374, 2015.

[19] M. S. K. Yanmin Sun and Y. Wang. Boosting for learning multiple classes with imbalanced class distribution. In *The Sixth International Conference on Data Mining*. ICDM, 2006.

[20] J. Zhang, C. Chen, Y. Xiang, W. Zhou, and A. V. Vasilakos. An effective network traffic classification method with unknown flow detection. In *IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT*. IEEE, 2013.