Multi-model coupling method for imbalanced network traffic classification based on clustering

Zhengzhi Tang, Xuewen Zeng* and Jun Chen

National Network New Media Engineering Research Center,
Institute of Acoustics, Chinese Academy of Sciences (IACAS),
Beijing, China
and
University of Chinese Academy of Sciences,
Beijing – 100190, China
Email: tangzz@dsp.ac.cn
Email: zengxw@dsp.ac.cn
Email: chenj@dsp.ac.cn
*Corresponding author

Abstract: The imbalanced category of network traffic poses a challenge to the classification methods based on machine learning, because the unbalanced data structure affects the performance of machine learning algorithms. In this paper, we propose a multi-model coupling approach to address the imbalanced data problem in network traffic classification. We process the major class to some clusters by a clustering algorithm. Then, these clusters and the minor class are used to form the training dataset for training model respectively. During the test, the test dataset is input into the previously trained models respectively, and the classification results of respective models are coupled to obtain the final result. We tested our proposed method on two well-known network traffic datasets and the results showed that it could achieve better performance and less time consumption compared with recent proposed methods in the case where the ratio of minor to major classes is very small.

Keywords: machine learning; imbalanced network traffic classification; clustering algorithm; multi-model coupling.

Reference to this paper should be made as follows: Tang, Z., Zeng, X. and Chen, J. (2020) 'Multi-model coupling method for imbalanced network traffic classification based on clustering', *Int. J. High Performance Computing and Networking*, Vol. 16, No. 1, pp.26–35.

Biographical notes: Zhengzhi Tang received his PhD in Signal and Information Processing from the National Network New Media Engineering Research Center, Institute of Acoustics, Chinese Academy of Sciences (IACAS) and University of Chinese Academy of Sciences. His research interests include network security and machine learning (ML).

Xuewen Zeng received his BSc from the Shanghai Jiao Tong University, Shanghai, China, and MSc and PhD in Signal and Information Processing from the Institute of Acoustics, Chinese Academy of Sciences (IACAS), Beijing, China. He is currently working at the National Network New Media Engineering Research Center, IACAS as a Research Professor. His research interests include network new media technology, and media information security.

Jun Chen received his BSc and MSc from the Zhejiang University, Zhejiang, China, and PhD in Signal and Information Processing from the Institute of Acoustics, Chinese Academy of Sciences (IACAS), Beijing, China. She is currently working at the National Network New Media Engineering Research Center, IACAS as a Research Professor. Her research interests include industrial control safety.

1 Introduction

Information and communications technologies (ICTs) play an important role and opportunity in pursuing sustainable development goals (SDGs) (Wu et al., 2018). With the development of network technology, the identification of network traffic as one of the network analysis technologies is becoming more and more important. As the foundation of network cognition, management and optimising, network traffic classification is making a significant difference in resource scheduling, safety analysis and future tendency prediction (Shen et al., 2017). According to whether the traffic is encrypted or not, the classification of network traffic can be categorised into encrypted traffic classification and non-encrypted traffic classification. The research on non-encrypted traffic classification is earlier and the technology is mature. However, the research on encrypted traffic classification starts late, and it becomes an active and hot research area because of its challenges.

The methods of network traffic identification are becoming more mature and practical. As shown in Table 1, the classification accuracy of each method has certain differences, and the machine learning method has the characteristics of real-time and high accuracy, which has been widely studied. However, one of the challenging problems in applying machine learning techniques for network traffic classification is the imbalanced proportion of protocols or applications in network traffic. These methods shown in Table 1 all ignore the actual situation that the amount of traffic in some applications or protocols is much higher than the amount of traffic in other applications or protocols in the network during the research process. Currently, the research on network traffic classification mainly focuses on extracting distinguishing features effectively and the performance optimisation of the classifier. And the imbalanced class of the real network traffic has a great impact on the identification result, which will cause the minor class to be misidentified into other class and lead its identification accurate to be low in the actual engineering application.

In the network traffic classification, if only the problem of identifying encrypted traffic is considered, an obvious case of class-imbalanced traffic is the amount of encrypted traffic and non-encrypted traffic. A large amount of traffic generated by various applications and protocols is full of networks, but the proportion of encrypted traffic is generally much smaller than the unencrypted traffic. However, there are few researches on examining the effect of the methods for addressing imbalanced data (Vu et al., 2016, 2017). Vu et al. (2016), presented a thorough analysis of the impact of various techniques, such as random under sampling (RUS), random over sampling (ROS), synthetic minority over-sampling technique (SMOTE) and so on, for handling imbalanced data when machine learning approaches are applied to identifying encrypted traffic. The results showed that some techniques for addressing imbalanced data can help machine learning algorithms to achieve better performance. Vu et al. (2017), a recent proposed deep network for unsupervised learning called auxiliary classifier generative adversarial network (AC-GAN) to generate synthesised data samples for balancing between the minor and the major classes. The results showed that their proposed method achieved better performance compared with recent proposed methods, such as SMOTE-SVM and BalanceCascade, for handling imbalanced data problem in network traffic classification. However, AC-GAN is not easy to train and has large time consumption. Moreover, the ratio of the minor to major classes is very small, and the data generated by AC-GAN is much larger than the original data, making the original data less effective in training, which is not reasonable.

In this paper, we propose an application for coupling multiple machine learning models to address imbalanced data problem in network traffic classification. The major class is pre-processed by a clustering algorithm, and then multiple models are trained. Finally, the classification results of multiple models are coupled to obtain the final result. The experimental results show that the multi-model coupling approach can help classification algorithms to achieve better performance in class-imbalanced network traffic classification in the case where the ratio of minor to major classes is very small. The main contributions of this paper are as follows:

- We propose a multi-model coupling approach to address imbalanced data problem. The proposed method avoids changes in the amount of data brought about by the oversampling process of generating the samples and the under-sampling process of removing the samples.
- Several clusters acquired by major class through a cluster algorithm are combined with the minor class to train respectively, and finally the classification results of all the trained models are coupled to obtain the final classification result.
- We carry out many experiments to evaluate the performance of proposed multi-model coupling method. Its performance outperforms the state-of-the-art imbalanced process techniques in the case where the ratio of minor to major classes is very small.

The rest of this paper is organised as follows. Section 2 puts forward a summary of recent related work. Section 3 researches the impact of class imbalance on each classification algorithm. Section 4 displays the multi-model coupling approach. Section 5 presents the experimental results for the multi-model coupling approach on the given public datasets. Finally, Section 6 concludes the work and analyses possible future studies.

2 Related work

According to the work of Khater and Overill (2016) and Pan et al. (2016), we have summarised the methods used in current traffic classification as shown in Table 1.

It can be seen from the Table 1 that different methods are specific to the different traffic characteristics and these approaches can be categorised into six main categories as follows:

1 Port-based approach (Dainotti et al., 2012). It extracts the port number which is assumed to be associated with a particular application from the TCP/UDP headers of the packets.

Table 1 The summary of traffic classification methods

Methods	*Inspection content*	*Encryption or not*	*Classification accuracy (high > medium > low)*
Port-based	Port number	Non-encryption	Past high, but now low
Payload inspection	Payload	Both	High (encryption only for HTTPS)
Behaviours	Host behaviours	Both	Medium
Machine learning	Flow statistics features	Both	High
Payload randomisation and distribution	Partial payload	Encryption	Medium
Hybrid approach	Many features	Encryption	High

2 Payload inspection approach (Finsterbusch et al., 2014). It uses rule matching (such as regular expressions) or other methods to analyse the application layer payload of packets.

3 Behaviours-based approach (Karagiannis et al., 2005). It builds an interaction graphs model from the perspective of application level layer interaction behaviours among hosts and then analyse such interaction graphs with graph theory techniques.

4 Machine learning approach (Perera et al., 2017; Singh et al., 2016; Chen et al., 2017; Lopez-Martin et al., 2017). Machine learning approaches, including traditional machine learning and deep learning, are currently the most studied. With the widespread use of GPU and the development of specialised artificial intelligence (AI) chips, machine learning methods have become very efficient.

5 Payload randomisation and distribution (Khakpour and Liu, 2013; Zhao et al., 2013). According to the characteristics of the network application traffic is not completely encrypted, and the traffic can be identified by the randomness of same characteristic fields carried by each packet.

6 Hybrid approach (Sun et al., 2010). By combining multiple algorithms, a better identification accuracy can be achieved.

Rare events are difficult to detect because of their infrequency and casualness. Moreover, misclassifying rare events can result in heavy costs. Classifying imbalanced data significantly challenges traditional classification models, such as Gaussian Naive Bayes (GNB), support vector machines (SVMs), etc. (Guo et al., 2017). In the work of Guo et al. (2017), studied a total of 527 papers on the classification of imbalanced data from 2006 to 2016. The research areas covered mainly include chemical, biomedical engineering, financial management, information technology and so on. The authors give an overview of the state-of-the-art imbalanced learning techniques. In this section, we will discuss the classification technologies of class-imbalanced network traffic more detail from two perspectives.

2.1 Basic strategies for dealing with imbalanced learning

Two basic strategies for addressing imbalanced learning are introduced, which are, pre-processing and cost-sensitive learning. Pre-processing approaches include resampling methods conducted in the sample space and feature selection methods that optimise the feature space. Vu et al. (2016), studied the effects of commonly resampling techniques, such as RUS, ROS, SMOTE and so on, in dealing with imbalanced encrypted traffic classification. The results show that some resampling techniques are effective and stable for addressing imbalanced encrypted traffic classification. Shen et al. (2017) and Ding (2016), solve the class imbalance problem by feature selection and extraction, including the re-generation of integrated feature subset or re-extraction of feature sets combined with sampling methods. The final experimental results verify the effectiveness of these methods. Chen et al. (2017), used SVM cost-sensitive (SVMCS) and other class imbalance classification methods to study the imbalanced mobile malware traffic identification problem. After comparing the performance, advantages and disadvantages of each method, the authors proposed their own solution.

2.2 Classification algorithms for imbalanced learning

Imbalanced learning attempts to build a classification algorithm that can provide a better solution for class imbalance problems than traditional classifiers such as SVM, KNN, decision trees (DTs) and neural networks. Two methods for solving imbalanced learning problems have been reported in the literature: ensemble methods and algorithmic classifier modifications. Ding (2015), proposed a network traffic classification method based on rotation forest. In their method, PCA was used for feature reduction and C4.5 algorithm was used to train base classifier. The experimental results show that their method has higher accuracy and stronger generalisation ability compared with C4.5 and Bagging algorithm, and is more suitable for imbalanced network traffic classification.

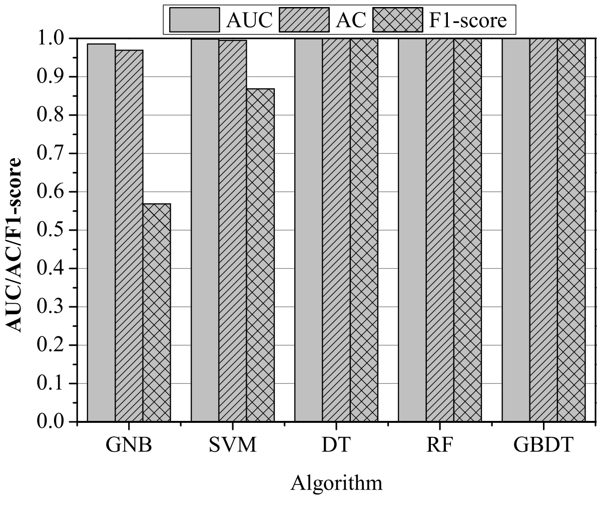
Wu et al. (2014), proposed an optimised distancebased nearest neighbour (ODNN), which has the capability of improving the classification performance of imbalanced traffic data. They analysed the proposed ODNN approach and its performance benefit from both theoretical and empirical perspectives. The results show that the performance of minor class can be improved significantly even only with small number of training data and the performance of major class remains stable.

According to the above comprehensive introduction, these improved methods have achieved good results in solving imbalanced data classification. But these methods still cannot avoid the shortcomings that exist. The resampling method has the problem of changing the original data information, and the ideal improvement effect is not obtained when the imbalance ratio is too small. The practice of changing the classification algorithm has the disadvantage that the algorithm design is complicated and may be data-oriented. In view of the shortcomings of the above methods, we propose a multi-model coupling method based on a clustering algorithm. The advantage of our method is that it is simple to implement and does not change the original data information.

3 Impact of class imbalance on classification algorithms

Because of the difference in principle of each classification algorithm, they will lead to great differences in the classification performance of imbalanced datasets. In this section, we carried out an experiment for the performances of some common machine learning classification algorithms, such as GNB, SVMs, DTs, random forests (RF), and gradient boosting decision tree (GBDT), on the imbalanced NIMS dataset (Alshammari and Zincir-Heywood, 2010).

Figure 1 The impact of class imbalance traffic on classification algorithms



The packets of NIMS dataset were collected from the internal network of Dalhousie University Computing and Information Services Centre (UCIS) in 2007. A statistical analysis of the NIMS dataset revealed that it has a total of 14,681 encrypted flows and 699,170 unencrypted flows, and the imbalanced ratio of encrypted to unencrypted flows is approximately 0.021. Therefore, the NIMS dataset can be considered as an serious imbalanced dataset. The performance metrics includes accuracy score (AC), F1 score and AUC score (Vu et al., 2017). These performance metrics are explained in detail in Subsection 5.2.

As seen from Figure 1, except for GNB and SVM classification algorithms, other classification algorithms perform well on imbalanced NIMS dataset. In view of SVM and GNB algorithms having poor adaptability to imbalanced data, so we use these two algorithms to verify that whether our proposed algorithm for imbalanced data pre-process outperforms the state-of-the-art pre-process techniques in the follow-up experiments. Herein, a briefly introduction of GNB and SVM is given (Supervised Learning, http://sklearn.apachecn.org/).

3.1 Gaussian Naive Bayes

Naive Bayes methods (Supervised Learning, http://sklearn.apachecn.org/) are a set of supervised learning algorithms based on applying Bayes' theorem with the 'naive' assumption of conditional independence between every pair of features given the value of the class variable. Bayes' theorem states the following relationship, given class variable y and dependent feature vector x_1 through x_n:

$$P(y|x_1, ..., x_n) = \frac{P(y)P(x_1, ..., x_n|y)}{P(x_1, ..., x_n)} \tag{1}$$

Since $P(x_1, ..., x_n)$ is constant given the input, we can use the following classification rule:

$$P(y|x_1, ..., x_n) \propto P(y) \prod_{i=1}^{n} P(x_i|y) \tag{2}$$

$$\Downarrow$$

$$\hat{y} = \arg\max_{y} P(y) \prod_{i=1}^{n} P(x_i|y) \tag{3}$$

and we can use maximum a posteriori (MAP) estimation to estimate $P(y)$ and $P(x_i|y)$; the former is then the relative frequency of class y in the training set. The different naive Bayes classifiers differ mainly by the assumptions they make regarding the distribution of $P(x_i|y)$. The GNB algorithm for classification that is the likelihood of the features is assumed to be Gaussian:

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right) \tag{4}$$

The parameters σ_y and μ_y are estimated using maximum likelihood.

3.2 Support vector machine

A SVM constructs a hyper-plane or set of hyper-planes in a high or infinite dimensional space, which can be used for classification, regression or other tasks. Only the binary SVM is briefly introduced here. Given training vectors $x_i \in \mathbb{R}^p, i = 1, ..., n$, in two classes, and a vector $y \in \{-1, 1\}^n$, SVM solves the following primal problem (Supervised Learning, http://sklearn.apachecn.org/):

$$\begin{aligned}&\min_{\omega,b,\xi} \frac{1}{2}\omega^T\omega + C\sum_{i=1}^{n}\xi_i\\ &\text{subject to}\\ &y_i(\omega^T\phi(x_i) + b) \geq 1 - \xi_i, \xi_i > 0, i = 1, ..., n\end{aligned} \tag{5}$$

Its dual is:

$$\begin{aligned}&\min_{\alpha} \frac{1}{2}\alpha^T Q\alpha - e^T\alpha\\ &\text{subject to}\\ &y^T\alpha = 0, 0 \leq \alpha_i \leq C, i = 1, ..., n\end{aligned} \tag{6}$$

where e is the vector of all ones, $C > 0$ is the upper bound, Q is an n by n positive semidefinite matrix, $Q_{ij} \equiv y_i y_j K(x_i, x_j)$, where $K(x_j, x_i) = \phi(x_i)^T\phi(x_j)$ is the kernel. Here training vectors are implicitly mapped into a higher (maybe infinite) dimensional space by the function ϕ. The decision function is:

$$\text{sgn}\left(\sum_{i=1}^{n} y_i\alpha_i K(x_i, x) + \rho\right) \tag{7}$$

Herein, the kernel function is radial basis function (RBF): $\exp(-\gamma\,\|x - x'\|^2)$. γ is specified by keyword gamma, must be greater than 0.

4 The multi-model coupling method

This section presents the proposed multi-model coupling method based on a clustering algorithm in details. In Algorithm 1, it can be seen that the entire multi-model coupling method can be divided into two phases. During the training phase, the input training dataset was categorised into minor class and major class. Herein, the minor class is encrypted traffic and major class is non-encrypted traffic. Then the major class was processed by a cluster algorithm to be categorised into some clusters. Finally, each cluster in $major\_clusters$ is combined with the minor class to form the training dataset for training machine learning model, and the corresponding different trained models were acquired. Since the training of multiple models can be processed in parallel, it is foreseeable that our algorithm has great advantages in training time. During the test phase, the test dataset was input into the previously trained models respectively, and the identification results of the respective models are coupled to obtain the final identification result.

Herein, the coupling method is elaborated in details. We assume that the results of test dataset on all trained models are $\{result_1, ..., result_N\}$ ($result_i$ here means accuracy). And all $result_i$ satisfy $result_1 \geq result_2 \geq, ..., \geq result_N$, if they are not satisfied, they need to be processed in descending order. We also assume that the computed probabilities of possible outcomes for samples in test dataset are $\{P_i(p_{-1}(j), p_1(j)), i = 1, ..., N, j = 1, ..., m\}$, m is the number of test samples and the order of P_i is corresponding to the above $result_i$. Herein, $p_{-1}(j)$ indicates the probability that the prediction label is –1, $p_1(j)$ indicates the probability that the prediction label is 1. If $p_{-1}(j) > p_1(j)$, it indicates that the prediction label of the j^{th} sample is –1, otherwise it is 1. Assuming that the probability of the j^{th} sample label after coupling is $P_{coupling}(j)$.

$$\begin{aligned}P_{coupling}(j) &= \alpha_1 P_1 + \alpha_2 P_2 + \cdots + \alpha_N P_N\\ &= \alpha_1 P_1(p_{-1}(j), p_1(j))\\ &+ \alpha_2 P_2(p_{-1}(j), p_1(j))\\ &+ \cdots + \alpha_N P_N(p_{-1}(j), p_1(j))\\ &= P_1(\alpha_1 p_{-1}(j), \alpha_1 p_1(j))\\ &+ P_2(\alpha_2 p_{-1}(j), \alpha_2 p_1(j))\\ &+ \cdots + P_N(\alpha_N p_{-1}(j), \alpha_N p_1(j))\\ &= P(\alpha_1 p_{-1}(j) + \cdots\\ &+ \alpha_N p_{-1}(j), \alpha_1 p_1(j))\\ &+ \cdots + \alpha_N p_1(j))\end{aligned} \tag{8}$$

where α_i is a weight coefficient whose magnitude is proportional to the accuracy $result_i$ corresponding to P_i. The greater $result_i$ is, the larger α_i is.

Algorithm 1 Multi-model coupling

```
Begin to train
Input the training dataset, initialise $i$ = 0.
Training dataset was categorised into $minor\_class$ and $major\_class$.
$major\_clusters \Leftarrow$ cluster($major\_class$).
for $cluster$ in $major\_clusters$ do
    model($i$) $\Leftarrow$ combine $cluster$ with $minor\_class$ to train
    $i = i + 1$
end for
End
Begin to test
Input the test dataset.
$N \Leftarrow$ get the number of trained model.
for $j = 1$ $to$ $N$ do
    result($j$) = $model_j(test\ dataset)$
end for
$Final\_result$ = Coupling($result$)
End
```

After coupling, the range of output values from the output layer is uncertain, it is difficult for us to visually judge the meaning of these values. On the other hand, since the actual label has discrete values, the error between these discrete values and the output values from an uncertain range is difficult to measure. We could try forcing the outputs to correspond to probabilities by softmax. The output of softmax regression is subjected to a nonlinearity which ensures that the sum over all outcomes always adds up to 1 and that none of the terms is ever negative. The nonlinear transformation works as follows:

$$\hat{P} = softmax(P) = softmax(P_{coupling}(j))$$
$$\text{where } \hat{P}_i = \frac{exp(P_i)}{\sum_j exp(P_j)} \tag{9}$$

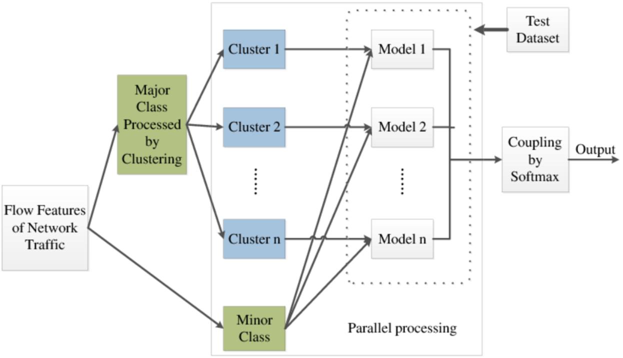
Table 2 Statistical flow features for network traffic

Index	*Abbreviation*	*Detailed description*
1	proto	The protocol (i.e., TCP = 6, UDP = 17)
2	duration	The duration of the flow
3	fb_psec	Flow bytes per second
4	fp_pesc	Flow packets per second
5	mean_flowiat	The mean time of flow inter arriving
6	std_flowiat	The standard deviation from the mean time of flow inter arriving
7	max_flowiat	The max time of flow inter arriving
8	min_flowiat	The min time of flow inter arriving
9	min_fiat	The min time between two forward packets
10	mean_fiat	The mean time between two forward packets
11	max_fiat	The max time between two forward packets
12	std_fiat	The standard deviation from the mean time between two forward packets
13	min_biat	The mini time between two backward packets
14	mean_biat	The mean time between two backward packets
15	max_biat	The max time between two backward packets
16	std_biat	The standard deviation from the mean time between two backward packets
17	mean_active	The mean time that a flow was active before going idle
18	std_active	The standard deviation from the mean time that the flow was active before going idle
19	max_active	The max time that the flow was active before going idle
20	min_active	The min time that the flow was active before going idle
21	mean_idle	The mean time a flow was idle before becoming active
22	std_idle	The standard deviation from the mean time a flow was idle before becoming active
23	max_idle	The max time a flow was idle before becoming active

After being processed by softmax, the prediction label is judged as: if $P_{-1}(j) > P_1(j)$, it indicates that the prediction label of the j^{th} sample is –1, otherwise it is 1.

Figure 2 shows the detailed process of the multi-model coupling method based on a clustering algorithm. As seen from Figure 2, the input network flow features are firstly categorised into two parts: major class and minor class. Then the major class is processed by a clustering algorithm to some clusters, and the obtained clusters are combined with minor class to form a training set to obtain multiple models respectively. This process can be performed in parallel, that is, the part of the solid line frame in the figure. Therefore, it is foreseeable that our method will have an advantage in time consumption for training and classification. The test set is input to each model in the dashed frame, and the output of each model is processed by softmax and coupling to obtain the final output.

Figure 2 Detailed principle process of multi-model coupling method based on a clustering algorithm (see online version for colours)



5 Experiments and results

To demonstrate the advantage of proposed multi-model coupling method based on a clustering algorithm. We compared it with three recent presented methods as stated in Vu et al. (2017), such as SMOTE-SVM, AC-GAN, and BalanceCascade, for handling imbalanced problem in network traffic classification. All the three methods use the idea of resampling to generate or reduce data from the training set samples.

5.1 *Description of dataset*

The datasets used in the experiments are two well-known network traffic datasets NIMS (Alshammari and Zincir-Heywood, 2010) and ISCXTor2017 [Tor-nonTor Dataset (ISCXTor2017), https://www.unb.ca/cic/datasets/tor.html]. The NIMS dataset has been introduced in Section 3. The ISCXTor2017 was generated by Canadian Institute for Cybersecurity (CIC). They created three users for the browser traffic collection and two users for the communication parts such as chat, mail, FTP, p2p, etc. In totally, the ISCXTor2017 dataset includes 14,508 TOR flows and 69,685 non-TOR flows and the ratio of TOR to non-TOR flows is about 0.208.

A flow is defined as all packets that have the same five-tuple, i.e., source IP, source port, destination IP, destination port and transport protocol. Each flow is described by 22 statistical flow features as stated in Alshammari and Zincir-Heywood (2010) for NIMS dataset. And each flow is described by 23 statistical flow features as shown in Table 2 for ISCXTor2017 dataset. Most of the statistical flow features used in ISCXTor2017 dataset are different from NIMS dataset. We randomly selected 50% data samples for training and the rest for testing.

Table 3 The classification performance after major class being processed by different clustering alogrithms for NIMS dataset

Clustering algorithm	*Parameter*	*SVM*			*GNB*		
	(α_1, α_2)	*AUC*	*AC*	*F1-score*	*AUC*	*AC*	*F1-score*
Kmeans	(1, 0)	0.998157	0.994974	0.868321	0.987244	0.971246	0.587118
	(1, 1)	*0.998224*	0.949642	0.449562	0.986617	0.675940	0.112624
	(10, 1)	*0.998224*	0.994957	0.869603	0.986637	0.971243	0.587095
	(100, 1)	0.998187	0.994951	0.869231	0.986698	0.971246	0.587118
MiniBatchKmeans	(1, 0)	0.996618	*0.994988*	0.868639	0.985012	0.968660	0.566736
	(1, 1)	0.998220	0.953864	0.471313	0.985005	0.928528	0.365265
	(10, 1)	0.997777	0.994963	*0.869842*	0.985005	0.968660	0.566736
	(100, 1)	0.996743	0.994960	0.869496	0.985005	0.968660	0.566736
Birch	(1, 0)	0.996615	*0.994988*	0.868639	*0.993506*	*0.974370*	*0.613389*
	(1, 1)	0.998221	0.956315	0.484937	0.993414	0.935107	0.387929
	(10, 1)	0.997775	0.994963	*0.869842*	0.993419	0.974336	0.613078
	(100, 1)	0.996740	0.994960	0.869496	0.993419	0.974367	0.613363

Table 4 The classification performance after major class being processed by diffierent clustering algorithms for ISCXTor2017 dataset

Clustering algorithm	*Parameter*	*SVM*			*GNB*		
	(α_1, α_2)	*AUC*	*AC*	*F1-score*	*AUC*	*AC*	*F1-score*
Kmeans	(1, 0)	0.815764	*0.728452*	*0.547792*	0.814505	0.717666	0.538736
	(1, 1)	0.919709	0.728286	0.547641	0.892974	0.174760	0.294586
	(10, 1)	0.920727	0.728096	0.547467	0.901638	0.717666	0.538736
	(100, 1)	*0.921612*	0.728096	0.547467	0.902895	0.717666	0.538736
MiniBatchKmeans	(1, 0)	0.816689	*0.728452*	*0.547792*	0.814506	0.717666	0.538736
	(1, 1)	0.919898	0.728286	0.547641	0.892932	0.174736	0.294580
	(10, 1)	0.920932	0.728048	0.547424	0.902198	0.717666	0.538736
	(100, 1)	0.921499	0.728048	0.547424	0.902852	0.717666	0.538736
Birch	(1, 0)	0.830848	0.723748	0.544036	0.819805	*0.719733*	*0.540454*
	(1, 1)	0.912994	0.723582	0.543886	0.900561	0.275872	0.321158
	(10, 1)	0.915088	0.723724	0.544014	0.907940	*0.719733*	*0.540454*
	(100, 1)	0.919906	0.723724	0.544014	*0.907974*	0.719732	*0.540454*

Table 5 The classification performance for different methods in NIMS dataset

Methods	*SVM*		
	AUC	*AC*	*F1-score*
Non-handle	0.9982	0.9950	0.8683
SMOTE-SVM	*0.9992*	0.9869	0.7585
BalanceCascade	0.9984	0.9944	0.8568
AC-GAN	0.9944	0.9879	0.5816
Multi-model	0.9966	*0.9950*	*0.8698*
Methods	*GNB*		
	AUC	*AC*	*F1-score*
Non-handle	0.9851	0.9689	0.5684
SMOTE-SVM	0.9795	0.9639	0.5121
BalanceCascade	0.9856	0.9699	0.5761
AC-GAN	0.9747	*0.9874*	0.5827
Multi-model	*0.9935*	0.9744	*0.6134*

5.2 *Experiment setup and evaluation metrics*

The experimental platform is a Dell R720 server which is equipped with CentOS release 7.3 operate system. The CPU is a 16-core XeonE5620 2.40 GHz, and the memory is 16 GB. The classification algorithms used in all experiments, such as GNB, SVM, DT, RF, and GBDT, are from Scikit-learn tool. For the convenience of comparison, all classification algorithms in the experiments use the default parameter settings. In this paper, three evaluation metrics are used: accuracy (AC), F1 value (F1), and area under curve (AUC) score.

$$AC = \frac{TP + TN}{TP + TN + FN + PN}, F1 = \frac{2PR}{P + R} \tag{10}$$

where TP is the number of instances correctly classified as X, TN is the number of instances correctly classified as Not-X, FP is the number of instances incorrectly classified as X, and FN is the number of instances incorrectly classified as Not-X (Wang et al., 2017). P indicates the precision and R indicates the recall. Accuracy is used to evaluate the overall performance of a classier. F1 value is used to evaluate performance of every class of traffic. F1 value indicates the classification performance of minor class in this paper. The AUC score is defined as the area under the ROC curve, which indicates whether the classifier is good or not.

5.3 *Impact of different clustering algorithms*

In this section, we research the classification performance of major class processed by different clustering algorithms.

We use the Kmeans, MiniBatchKmeans, and Birch clustering algorithms in Scikit-learn tool. In Scikit-learn tool, we use $calinski\_harabaz\_score$ in metrics to evaluate the cluster number. However, the cluster number of major class evaluated is not unique, making the analysis more complicated. Therefore, we only analyse the simple case where the cluster number of major class is 2 for all the clustering algorithms. When major class is clustered into two small clusters, only two models' outputs need to be coupled. So the weight coefficient in equation (8) has only α_1 and α_2.

Table 6 The classification performance for different methods in ISCXTor2017 dataset

Methods	*SVM*		
	AUC	*AC*	*F1-score*
No-handling	0.9159	0.8631	0.409
SMOTE-SVM	*0.9602*	0.8183	*0.6524*
BalanceCascade	0.9122	0.8490	0.4366
AC-GAN	0.8141	0.8332	0.3929
Multi-model	0.8170	0.7285	0.5480
Methods	*GNB*		
	AUC	*AC*	*F1-score*
No-handling	0.9122	0.8349	0.6654
SMOTE-SVM	0.9138	0.8332	0.6640
BalanceCascade	*0.9220*	*0.8481*	*0.6754*
AC-GAN	0.8377	0.7552	0.5483
Multi-model	0.9080	0.7197	0.5405

Table 7 The training time consumption (second) after imbalanced training dataset being processed by different methods

Dataset	*NIMS*		*ISCXTor2017*	
Classification algorithm	*SVM*	*GNB*	*SVM*	*GNB*
No-handling	2,610.14	0.68	296.08	0.08
SMOTE-SVM	74,062.6	0.57	1,613.03	0.06
BalanceCascade	563.69	0.33	467.6	0.05
AC-GAN	10,694.33	1.46	613.33	0.12
Multi-model	*179.19*	0.41	*123.14*	0.06

Table 8 The time consumption (second) of imbalanced data processed by different methods

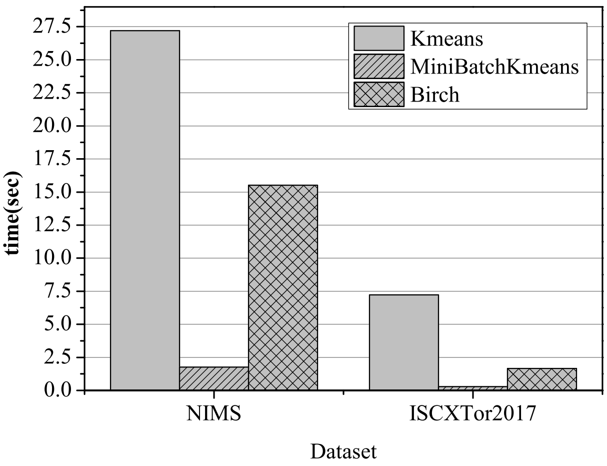
Dataset	*NIMS*	*ISCXTor2017*
SMOTE-SVM	307.95	55.57
BalanceCascade	221.63	162.03
AC-GAN	5,204.26	5,318.26
Multi-model	*15.52*	*1.72*

Tables 3 and 4 show the classification performance after major class being processed by different clustering algorithms for NIMS dataset and ISCXTor2017 dataset. As shown in Tables 3 and 4, to reach the best classification performance, the values of α_1 and α_2 are set to (1, 0), (1, 1), (10, 1), and (100, 1), and the values of α_1 and α_2 which can reach the best classification performance will be adopted. In NIMS dataset, when major class is processed by Birch clustering algorithm, the classification performance is best for GNB. For SVM, the MiniBatchKmeans and Birch clustering algorithm can both reach a better classification performance. In ISCXTor2017 dataset, the classification performance of SVM can reach better when using MiniBatchKmeans and Kmeans clustering algorithm, but it is Birch clustering algorithm for GNB. Figure 3 shows the time consumption of major class being processed by different clustering algorithms for NIMS dataset and ISCXTor2017 dataset. As shown in Figure 3, the Kmeans clustering algorithm consumes the most time, followed by the Birch clustering algorithm. The minimum time consumption is the MiniBatchKmeans clustering algorithm. Considering the classification performance and time consumption comprehensively, the MiniBatchKmeans and Birch clustering algorithms not only have better classification performance, but also consume less time.

5.4 Comparison

In Subsection 5.3, we research the best classification performance of proposed multi-mode coupling method based on a clustering algorithm in the case of different parameters. In this section, the best classification performance of multi-mode coupling method will be compared with the classification results of representative imbalanced data processing algorithms, such as SMOTE-SVM, BalanceCascade, and AC-GAN. Tables 5 and 6 show the classification performance for different data processing methods in NIMS dataset and ISCXTor2017 dataset respectively.

Figure 3 The time consumption of major class being processed by different clustering algorithms



As shown in Table 5, in the case where the ratio of minor to major classes is very small, the SMOTE-SVM method does not improve the poor classification performance of minor class for SVM and GNB. The recently proposed AC-GAN method which generates minor class data and BalanceCascade method both only improve the classification performance of GNB. Moreover, from

Tables 7 and 8, AC-GAN not only takes the most time for data pre-processing, but also consumes lots of time when training the model, and the AC-GAN model is more difficult to train. But, compared with the classification performance of No-handling, our proposed multi-model method significantly improves the classification performance of SVM and GNB as shown in Table 5. Moreover, the proposed multi-model coupling method outperforms the state-of-the-art imbalanced process techniques, such as SOMTE-SVM, BalanceCascade, and AC-GAN. Most importantly, the time of data pre-processing and model training for the proposed multi-model method are minimal except for the training time of GNB as shown in Table 7 and 8.

As shown in Table 6, in the case that the ratio of minor to major classes is not very small, the BalanceCascade method does improve the classification performance for GNB and increase the F1-score by 1%, the accuracy by 1.32%, and the AUC by 0.98%. But SMOTE-SVM, AC-GAN, and multi-model coupling methods take no effect. For the SVM, SMOTE-SVM, BalanceCascade, and multi-model coupling methods increase the F1-score, but the accuracies and AUCs decrease. The SMOTE-SVM method can increase the F1-score by 24.34% and the AUC by 4.43% for SVM, but the accuracy decreases. Although the algorithm we proposed is not outstanding, it is equivalent to the overall performance of other algorithms for the ISCXTor2017 dataset. However, our proposed method still has minimal time consumption except for the training time of GNB as shown in Tables 7 and 8.

As shown in Table 7, in the proposed multi-model training stage, since the major class is divided into small clusters by clustering algorithms, the amount of training dataset for each model is reduced, and the time consumption is reduced. More importantly, each model can be trained in parallel, and then the time consumption of entire training process is smaller. But in Table 7, the training process of GNB algorithm is not very time-consuming, mainly because the calculation process of Bayesian is simple and has a small amount of computation. The amount of training dataset has no significant impact on the time consumption of training process.

As shown in Table 8, the data pre-processing of our proposed multi-model coupling method is minimal. Since the clustering algorithm is used for data pre-processing, and the clustering algorithm can be operated in parallel, therefore, the time consumption in the data pre-processing stage is minimal.

5.5 *Analysis and discussion*

From the above experiments, the proposed multi-model coupling method based on a clustering algorithm performs better than the common resampling methods, such as SMOTE-SVM, BalanceCascade, and the recently proposed AC-GAN method, in the case that the ratio of minor to major classes is very small. In the case of a serious imbalance between minor and major classes, the resampling methods will seriously affect the original data information, which may cause the classification performance to become worse. In the case where the imbalance between minor and major classes is not serious, the resampling methods may have little impact on the data, and thus can improve the classification performance. The proposed multi-model coupling method based on a clustering algorithm has the characteristic that it neither generates data for minor class nor reduces data for major class. However, the proposed method does not performance well on the ISCXTor2017 dataset. Of course, other algorithms also do the same. Herein, we think that the clustering algorithms used in experiments are not very suitable for ISCXTor2017 dataset. From the description of the ISCXTor2017 dataset, it is also known that the extracted flow statistical features used to classification are different from NIMS dataset. It may be that the extracted flow statistical features used in the ISCXTor2017 dataset are not suitable or a clustering algorithm that is more suitable for it needs to research.

6 Conclusions and future work

Imbalanced data classification is an active and hot research area in data classification issue, which governs classification performance. Motivated by improving the classification performance of imbalanced network traffic classification, we propose a novel multi-model coupling method based on a clustering algorithm. Our goal is to process the training dataset so that we neither generate data for minor class nor reduce data for major class. The key characteristic of proposed multi-model coupling method is to use a clustering algorithm to process the major class and obtain some clusters for major class. We combine each cluster with minor class to train and obtain different models respectively. Finally, we couple the test dataset classification results on each model. Experiments on SVM and GNB to evaluate the performance of proposed multi-model coupling method verify that it has great advantages in term of time consumption. Simultaneously, it can improve the poor classification performance of minor class for a dataset with a very small imbalance ratio of minor to major classes.

In the future, we will optimise the clustering algorithm to further improve classification performance, and moreover, we will study the solution to the imbalance problem in multi-class classification (Buda et al., 2017). Finally, we will implement experiments in real-time systems, such as real-time data collection and analysis system (Dang et al., 2017).

Acknowledgements

This work was supported by The Next-Generation Broad-band Wireless Mobile Communications Network National Science and Technology of Major Projects (No. 2017ZX03001019). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers.

References

Alshammari, R. and Zincir-Heywood, A.N. (2010) 'Can encrypted traffic be identified without port numbers, IP addresses and payload inspection', *Computer Networks*, Vol. 55, No. 6, pp.1326–1350.

Buda, M., Maki, A. and Mazurowski, M.A. (2017) 'A systematic study of the class imbalance problem in convolutional neural networks', *Neural Networks*, Vol. 106, No. 10, pp.249–259.

Chen, Z.X., Liu, Z.S., Peng, L.Z., Wang, L. and Zhang, L. (2017) 'A novel semi-supervised learning method for internet application identification', *Soft Computing*, Vol. 21, No. 8, pp.1963–1975.

Dainotti, A., Pescapé, A. and Claffy, K.C. (2012) 'Issues and future directions in traffic classification', *IEEE Network*, Vol. 26, No. 1, pp.35–40.

Dang, S.J., Liu, X., Wang, X.K. and Liu, C.M. (2017) 'Design of real-time data collection and analysis system based on spark streaming (in Chinese)', *Network New Media*, Vol. 6, No. 5, pp.48–53.

Ding, Y. (2015) 'A method of imbalanced traffic classification based on ensemble learning', *IEEE International Conference on Signal Processing, Communications and Computing*, pp.1–4.

Ding, Y. (2016) 'Imbalanced network traffic classification based on ensemble feature selection', *IEEE International Conference on Signal Processing, Communications and Computting*, Hong Kong, China, pp.1–4.

Finsterbusch, M., Richter, C., Rocha, E., Müller, J.A. and Hänßgen, K. (2014) 'A survey of payload-based traffic classification approaches', *IEEE Communications Surveys and Tutorials*, Vol. 16, No. 2, pp.1135–1156.

Guo, H.X., Li, Y.J., Shang, J., Gu, M.Y., Huang, Y.Y. and Gong, B. (2017) 'Learning from class-imbalanced data: review of methods and applications', *Expert Systems with Applications*, Vol. 73, No. 7, pp.220–239.

Karagiannis, T., Papagiannaki, K. and Faloutsos, M. (2005) 'BLINC: multilevel traffic classification in the dark', *Computer Communication Review*, Vol. 35, No. 4, pp.229–240.

Khakpour, A.R. and Liu, X.A. (2013) 'An information-theoretical approach to high-speed flow nature identification', *IEEE/ACM Transactions on Networking*, Vol. 21, No. 4, pp.1076–1089.

Khater, N.A. and Overill, R.E. (2016) 'Network traffic classification techniques and challenges', *10th IEEE International Conference on Digital Information Management*, Jeju, South Korea.

Lopez-Martin, M., Carro, B., Sanchez-Esguevillas, A. and Lloret, A.J. (2017) 'Network traffic classifier with convolutional and recurrent neural networks for internet of things', *IEEE Access*, Vol. 5, pp.18042–18050.

Pan, W.B., Cheng, G., Guo, X.J. and Huang, S.X. (2016) 'Review and perspective on encrypted traffic identification research', *Journal on Communications*, Vol. 37, No. 9, pp.154–167.

Perera, P., Tian, Y.C., Fidge, C. and Kelly, W. (2017) 'A comparison of supervised machine learning algorithms for classification of communications network traffic', *Lecture Notes in Computer Science*, Vol. 10634, pp.445–454.

Shen, J., Xia, J., Shan, Y. and Wei, Z. (2017) 'Classification model for imbalanced traffic data based on secondary feature extraction', *IET Communications*, Vol. 11, No. 11, pp.1725–1731.

Singh, M.P., Srivastava, G. and Kumar, P. (2016) 'Internet traffic classification using machine learning', *International Journal of Database Theory and Application*, Vol. 9, No. 12, pp.45–54.

Sun, G.L., Xue, Y.B., Dong, Y.F., Wang, D.S. and Li, C.L. (2010) 'An novel hybrid method for effectively classifying encrypted traffic', *IEEE Global Telecommunications Conference*, pp.1–5.

Supervised Learning [online] http://sklearn.apachecn.org/ (accessed 20 October 2018).

Tor-nonTor Dataset (ISCXTor2017) [online] https://www.unb.ca/cic/datasets/tor.html (accessed 1 October 2018).

Vu, L., Van Tra, D. and Nguyen, Q.U. (2016) 'Learning from imbalanced data for encrypted traffic identification problem', *Symposium on Information and Communication Technology*, ACM, Ho Chi Minh, Vietnam, pp.147–152.

Vu, L., Bui, C.T. and Nguyen, Q.U. (2017) 'A deep learning based method for handling imbalanced problem in network traffic classification', *Eighth International Symposium on Information and Communication Technology*, ACM, Nha Trang, Viet Nam, pp.333–339.

Wang, W., Zhu, M., Zeng, X.W., Ye, X.Z. and Sheng, Y.Q. (2017) 'Malware traffic classification using convolutional neural network for representation learning', *IEEE International Conference on Information Networking*.

Wu, D., Chen, X., Chen, C., Zhang, J., Xiang, Y. and Zhou, W. (2014) 'On addressing the imbalance problem: a correlated KNN approach for network traffic classification', *International Conference on Network and System Security*, Vol. 8792, pp.138–151.

Wu, J.S., Guo, S., Huang, H.W., Liu, W. and Xiang, Y. (2018) 'Information and communications technologies for sustainable development goals: state-of-the-art, needs and perspectives', *IEEE Communications Surveys and Tutorials*, Vol. 20, No. 3, pp.2389–2406.

Zhao, B., Guo, H., Liu, Q.B. and Wu, J.X. (2013) 'Protocol independent identification of encrypted traffic based on weighted cumulative sum test', *Journal of Software*, Vol. 24, No. 6, pp.1334–1345.