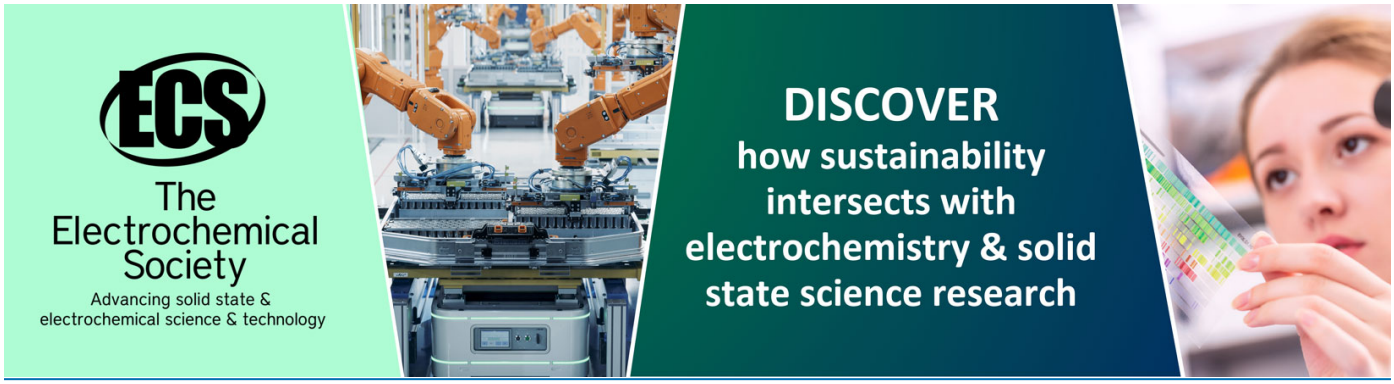
Improved application of transfer learning in network traffic classification



Improved application of transfer learning in network traffic classification

Fengjun Shang [1], Saisai Li [1], Jinlong He [2]

[1] Chongqing University of Posts and Telecommunications, Chongqing 400000, China

[2] Changchun University of Science and Technology, Changchun 130000, China

shangfj@cqupt.edu.cn

li_saisai@foxmail.com

mini92250@foxmail.com

Abstract. When using machine learning for traffic classification, there is such an assumption: the training data and the test data are independently and identically distributed. However, in reality, the assumption that the flow characteristics obey the same distribution may no longer hold because of conceptual drift or regional changes. Existing models will not be able to effectively classify new traffic. The transfer learning method TrAdaBoost has achieved great success in traffic classification and other aspects, but there are some problems, such as too much attention to the difficult-to-classify instances in the target domain, and failure to consider the wrong-classified instances in the source domain. In this study, the method of introducing weight correction factors in TrAdaBoost is used to make the iteration of weights more reasonable, and the effectiveness of this method is proved through theoretical analysis and experiments.

1. Introduction

With the increasing number of network users, the increasing variety of services, the gradual popularization of 5G networks, and the increasingly complex network behavior of users. It brings more challenges to the control and management of network traffic, anomaly detection, real-time situation analysis, safe operation and efficient use of network resources.

Through terminal application identification, the application type corresponding to the traffic in the network can be identified, and the type of the current main bandwidth traffic is known. Network managers of enterprises or campuses can timely adjust and intervene in key network traffic according to different situations, thereby ensuring the normal operation and smoothness of the network, and improving the quality of service (QoS) and quality of experience(QoE) of the network [1].

Because traditional machine learning can have a good performance, there is such a premise: the training data and the test data are from the same feature space, and the features follow the same probability distribution. When this assumption is not true, most trained statistical models need to be re-built using the collected training data again. Correspondingly, a new labeled data set is used, and the new data set is labeled to retrain the model It is a very laborious thing, and the price is often high. Transfer learning can greatly reduce the workload and training time in re-collecting training data.

2. Related Work and Problem Statement

2.1. Related work

Traffic classification has been studied for nearly 20 years, ranging from QoS and billing settings in ISP applications to firewall-related intrusion detection related to security [3]. The heavy use of mobile devices has greatly changed access to various network services and has led to the explosive growth of mobile service traffic [4][5]. Due to the emergence of new applications and restrictions on privacy by regulators, applications cannot infer their types [6]. For the identification of network application traffic, according to the different technologies and methods used, the traditional classification methods include port-based detection and identification [2], deep packet load detection and identification [7], and behavioral pattern-based identification and classification [3] The current mainstream method is machine learning recognition classification based on statistical characteristics of the traffic, including supervised learning, unsupervised learning, and semi-supervised learning. In terms of deep learning, Tal et al. Converted basic stream data into pictures, and then used a convolutional neural network (CNN) to classify the images to identify the category of the stream [8], and achieved good results.

2.2. Problem statement

By analyzing the current research status of machine learning recognition classification based on traffic statistical characteristics and instance-based transfer learning, it can be concluded that there are the following problems:

(1) The real world does not obey the assumption that training data and test data obey the same distribution. For traffic classification, the current machine learning model predicts the network traffic with a very high accuracy rate, but this has the illusion of self-satisfaction because the distribution of training data and the distribution of test data are the same. However, in the real world, the most prominent feature of network traffic data is its rapid evolution over time, so there is a phenomenon of concept drift, and the distribution of protocol types in different regions and different network environments is also inconsistent [9]. The previously available labeled data may become unavailable, resulting in a gap in semantics and distribution from the original test sample distribution [10]. This assumption is usually not true.

However, transfer learning applies the knowledge of the source domain to the target domain. Without making the above assumptions, it is suitable for classification problems where the target domain changes frequently. The comparison between the transfer learning method and the traditional machine learning method is shown in Fig. 1

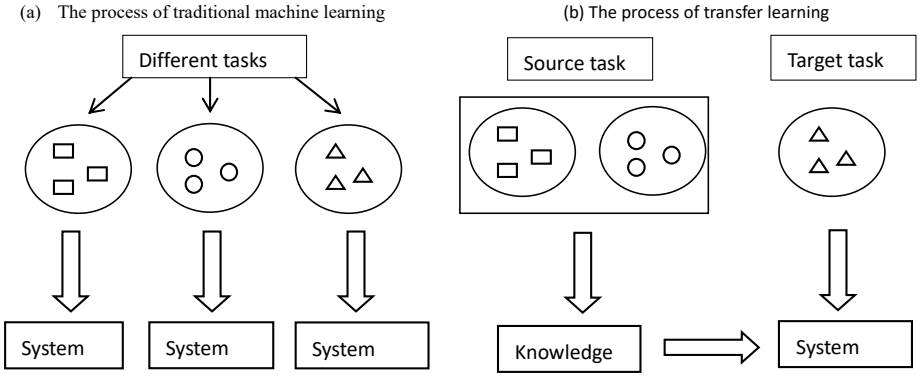


Fig. 1 The difference between traditional machine learning and transfer learning

(2) Transfer learning is prone to negative transfer, the weight of the source and target fields is seriously unbalanced, and complex guesses are easy to overfit. For instance-based transfer learning, the classic instance-based transfer learning algorithm TrAdaBoost has the problems of slow convergence, easy negative migration, easy over-fitting, source domain weight drop too fast, and small differences between base classifiers [11] .

The core idea of the TrAdaBoost algorithm is to use Boosting ideas to automatically filter out the samples in the source domain training data set that do not meet the target field training data set distribution. By assigning sample weights to the training data set, the source field training samples can be effectively migrated. measure. See Algorithm 1 for details of the TrAdaBoost algorithm.

3. Model Approach

The TrAdaBoost algorithm assigns weights to each sample in the source domain data set and the target domain data set and continuously iteratively updates the sample weights to train the base classifier. During each iteration, the weight of data samples in the misclassified target domain increases. At the same time, the weight of correctly classified data samples in the target domain is reduced. For the update of the weight of data samples in the source domain, the TrAdaBoost algorithm is based on the WMA algorithm. Contrary to the update strategy of the target domain sample weight value, if the source domain data sample is misclassified, then it can be considered that this type of sample does not conform to the target domain data distribution, and its sample weight needs to be reduced to reduce their impact on the target prediction classifier Degree, conversely, it is necessary to increase their sample weights to enhance their impact on the target domain prediction classifier.

But TrAdaBoost also has shortcomings, mainly manifested as the boosting iteration deepens, due to the algorithm sample weight update strategy, the difference between the source domain sample weight and the target domain sample weight is huge, which affects the effectiveness of knowledge transfer. TrAdaBoost convergence speed is not particularly fast, only $O(\sqrt{\ln\left(\frac{n}{N}\right)})$. In addition, the final integrated classifier of the TrAdaBoost algorithm ignores the base classifier trained in the first half of the algorithm iteration. This approach completely violates the fact that most data samples in machine learning applications are easily separable, and the data samples that are difficult to distinguish are only A few cases. The integration strategy of the TrAdaBoost algorithm classifier is easy to cause the problem of paying too much attention to difficult samples. So this time, Dynamic-TrAdaBoost, which introduces weight correction factors, is used for traffic analysis.

3.1. Dynamic-TrAdaBoost

Given a labeled source domain $\{x_{src_i}, y_{src_i}\}_{i=1}^{n}$, an target domain $\{x_{tar_i}, y_{tar_i}\}_{i=n+1}^{n+m}$, and assuming the feature space and label space is the same. But the marginal of the source and target domains are distributed differently. Transfer learning aims to learn the labels y_{tar} of D_{tar} using the source domain D_{src}.

Table.1 Dynamic-TrAdaBoost

Algorithm Dynamic-TrAdaBoost

Require:

Source domain instances $\mathbf{D}_{\mathbf{src}} = \left\{\left(\boldsymbol{x}_{src_i}, \boldsymbol{y}_{src_i}\right)\right\}$

Target domain instances $\mathbf{D}_{\mathbf{tar}} = \left\{\left(\boldsymbol{x}_{tar_i}, \boldsymbol{y}_{tar_i}\right)\right\}$

Maximum number of iterations:N

Base learner:f

Ensure: Target Classifier Output: $\left\{\mathbf{f} : \mathbf{X} \to \mathbf{Y}\right\}$

$$\boldsymbol{f} = \operatorname{sign}\left[\prod_{t=\frac{1}{2}}^{N}\left(\beta_{tar}^{t\ -\mathrm{f^t}}\right) - \prod_{t=\frac{1}{2}}^{N}\left(\beta_{tar}^{t\ -\frac{1}{2}}\right)\right]$$

Procedure:

$$w_{src} = \left\{ w_{src}^1, \ldots, w_{src}^n \right\}$$

1.Initialize the weight vector $\mathrm{D} = \left\{ \boldsymbol{D}_{\mathrm{src}} \cup \boldsymbol{D}_{tar} \right\}$ **,where:** $w_{tar} = \left\{ w_{tar}^1, \ldots, w_{tar}^m \right\}$

$$w = \left\{ w_{src} \cup w_{tar} \right\}$$

2.Set $\beta_{\mathrm{src}} = \dfrac{1}{1 + \sqrt{\dfrac{2\ln(n)}{N}}}$

3.for $\mathbf{t} = 1$ **to N do:**

Normalize Weights: $\mathbf{w} = \dfrac{w}{\sum_i^n w_{src\,i} + \sum_j^m w_{tar_j}}$

Find the candidate weak learner $\mathbf{f}^{\mathbf{t}} : X \to Y$ **that minimizes error for D weighted according to w**

Calculate the error of $\mathbf{f}^{\mathbf{t}}$ **on** $\mathbf{D}_{\mathbf{tar}}$ **:**

$$\varepsilon_{tar}^{\mathbf{t}} = \sum_{j=1}^{m} \frac{\left[w_{tar}^j \right] II \left[y_{tar_j} \neq f_j^t \right]}{\sum_{i=1}^{m} \left[w_{tar}^i \right]}$$

Set $\boldsymbol{\beta}_{\mathbf{tar}} = \dfrac{\varepsilon_{tar}^t}{1 - \varepsilon_{tar}^t}$

$$C^t = 2\left(1 - \varepsilon_{tar}^t\right)$$

$$w_{src_i}^{t+1} = C^t w_{src_i}^t \beta_{sr}^{t\ \mathrm{II}\left[y_{\mathrm{src}i} \neq f_i^t \right]}$$ **;where** $i \in D_{src}$

$$w_{tar_i}^{t+1} = w_{tar_i}^t \beta_{tar}^{t\ \mathrm{II}\left[y_{\mathrm{tar}i} \neq f_i^t \right]}$$ **;where** $i \in D_{tar}$

End for

The Dynamic-TrAdaBoost algorithm improves on the TrAdaBoost algorithm, adding an adaptive compensation parameter for sample weights in the source domain. When the integration degree is $\mathrm{N} \to \infty$ and the classification error of each base classifier on the source domain data set is ignored, all the source domain samples are correctly classified. At the t + 1 iteration, the sum of the weights of all samples in the source domain satisfies $\mathrm{S_s} \to n w_s^t$, and the weights of all samples in the target domain and $\mathrm{S_T}$ can be expressed as follow:

$$\mathrm{S_T} = m w_{tar}^t \left(1 - \varepsilon_{tar}^t\right) \beta_t^{\mathrm{I}\left[h_i(x_i) = y_{x_i} \right]} + m w_{tar}^t \left(\varepsilon_{tar}^t\right) \beta_t^{\mathrm{I}\left[h_i(x_i) \neq y_{x_i} \right]} = 2\mathrm{m}w_{\mathrm{tar}}^t \left(1 - \varepsilon_{tar}^t\right) \tag{1}$$

n and m represent the number of data sets in the source and target domains respectively, ε is the classification error rate, and t is the number of iterations. When t + 1 iterations, the distribution of sample weights in the source domain is as follow:

$$\mathrm{w}_{src}^{\mathrm{t+1}} = \frac{w_{src}^t}{S_s + S_T} = \frac{w_{src}^t}{n w_s^t + 2m w_{tar}^t \left(1 - \varepsilon_{tar}^t\right)} \tag{2}$$

The adaptive backfill parameter is $\mathrm{C^t}$, then after t iterations, the weight of the data samples in the source domain tends to be stable,

$$w_{src}^{t}=\frac{C^{t}w_{src}^{t}}{C^{t}nw_{src}^{t}+2mw_{tar}^{t}\left(1-\varepsilon_{tar}^{t}\right)} \tag{3}$$

It can be concluded that the replenishment parameters are:

$$C^{t}=2\left(1-\varepsilon_{tar}^{t}\right) \tag{4}$$

4. Experiment and Analysis

4.1. Data set introduction

Moore [13] and other researchers analyzed the TCP bidirectional flow with a complete three-way handshake and defined the data in the network without considering other circumstances. It defines a total of 248 attribute features and a special category feature (indicating the stream type of the data), such as the server port number, client port number, and various time intervals. As shown in Table 2, some definitions are introduced Meaning. By using the Nprobe network data collection tool, Moore et al. Divided into 10 periods of time after 24 hours of the whole day, each segment took 28 minutes to collect data, and then randomly collected the data collected in each period Organize to form a data set. 10 data sets were sorted out, namely entry01, entry02, entry03, entry04, entry05, entry06, entry07, entry08, entry09 and entry010. The Moore data set is a very practical data set in terms of network traffic classification, mainly because its data is relatively comprehensive. However, this also brings some problems. The huge amount of data causes processing troubles, especially its 248-dimensional features. There must be redundancy and features that are not very helpful for classification. Therefore, the feature selection algorithm based on feature weighted clustering based on information gain weight correlation coefficient feature is used for feature selection, and the redundant features and irrelevant features are screened to the maximum. 89, 46, 82, 43, 157, 155, 84, 184, 45, a total of 15 features.

Table.2 Explanation of data set definition

Number	***Short name***	***Meaning***
1	server port	Server port number
2	client port	Customer port number
3	min_IAT	The smallest packet arrival time of all sub-streams
4	q1_IAT	The time between the first quartile of packets in the stream
5	med_IAT	Median value of arrival time interval
6	mean_IAT	Average time between arrivals
7	q3_IAT	Interval time of the third quartile of packets in the stream
8	max_IAT	Maximum interval between packets in the stream
—	—	—
—	—	—
249	classes	Stream type

Table.3 Classification in the data set

	Stream type
C1	WWW
C2	MAIL
C3	FTP-CONTROL
C4	FTP-PASV
C5	ATTACK
C6	P2P
C7	DATABASE
C8	FTP-DATA
C9	MULTIMEDIA
C10	SERVICE

Through the analysis of 10 data sets, it is found that the Games application and Interactive application in 10 data sets account for a small proportion in each data set, some are zero, and some are close to zero. Therefore, this article will process the data when using the data set for the experiment. Integrate 10 data sets as shown in Table 3, then delete the Games application and Interactive application, extract 10% of the data from each category in the remaining data, and combine to form the experimental data set in this article, showing the number of traffic of the 10 applications in the new data set. Delete the Games application and Interactive application for datasets 1-10 as well. However, by observing the internal data of the data set, we know that some data streams are incomplete and have default values. This is because the flow collection tool does not collect the data, which causes a default situation. To avoid the influence of the default situation on the experiment in this paper, this article adopts the strategy of deleting this stream.

Aiming at the problems that the real world does not comply with the assumption that training data and test data are subject to the same distribution, that transfer learning is prone to negative transfer, and that the weights of the source and target fields are seriously unbalanced, transfer learning is used as a strategy, machine learning as the basis, and traffic-based statistics Research on feature classification of transfer learning recognition. Among them, for the problem that the real world does not obey the hypothesis that training data and test data obey the same distribution, a method of combining traffic classification and transfer learning is proposed; for the problem that transfer learning is prone to negative transfer and the weight of the source and target fields is seriously unbalanced, It is proposed to introduce a new weight update mechanism and introduce weight compensation factors

4.2. Evaluation indicators for results

The role of machine learning evaluation indicators is to evaluate the effectiveness of classification. Common evaluation indicators include precision, recall, accuracy, and f-score value.

Suppose the sample composition at this time is category A and category B, and A_right represents the number of A correctly classified, A_wrong represents the number of A incorrectly classified, B_right represents the number of correctly classified, B_wrong represents the number of B incorrectly classified.

There are:

$$precision\_A = \frac{A\_right}{A\_right + B\_wrong} \tag{5}$$

For this example, we know that B_wrong is wrongly classified into class A, and it can be used to measure the number of true correct classifications determined by the classifier.

$$recall\_A = \frac{A\_right}{A\_right + A\_wrong} \tag{6}$$

According to this example, it can be seen that A_wrong is erroneously classified into category B, and it can reflect the number of correct classifications that are correctly judged and occupy the proportion of total correct data.

$$accuracy = \frac{A\_right + B\_right}{A\_right + B\_right + A\_wrong + B\_wrong} \tag{7}$$

According to the analysis of this example, it can be used to reflect the judgment ability of the classifier on the entire sample, and correctly classify each category. The more the correct number, the higher the value, the better the judgment ability of the classifier on the sample.

$$f-\text{score} = \frac{2 * precision * recall}{precision + recall} \tag{8}$$

f-score is the harmonic mean of accuracy and recall. Recall is an evaluation standard of the classification model's ability to recognize correctly classified samples. When the value of recall is higher, it means that the trained model has stronger ability to recognize positive samples. Precision is an evaluation index of the model's ability to distinguish negative samples. When the value of precision is higher, it indicates that the trained model has stronger ability to distinguish negative samples. The f-score is a combination of the above two indicators. When the value of the f-score is higher, it means that the training classification model is more robust and more conducive to classification.

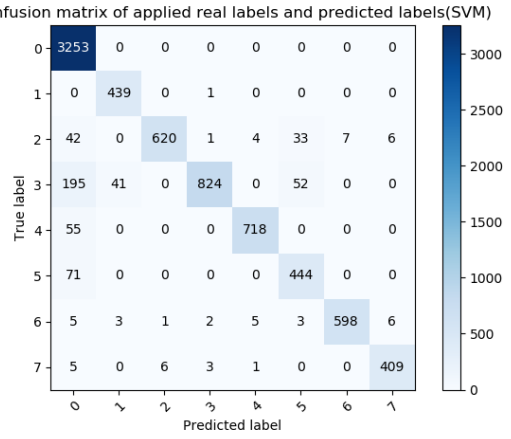


Fig.2 Classification results obtained by using SVM without assistance from the source domain

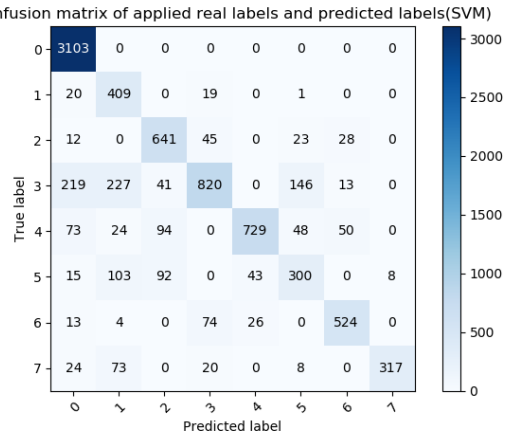


Fig.3 Only use the source domain data to train the classification results obtained by SVM

Table.4 Both training data and test data come from the target domain

Model	*Train_dataset*	*Pred_dataset*	*Mean Accuracy*	*f-score*
SVM	part of entry12	The other part of entry 12	0.885	0.842

Table.5 Training data and test data come from different domains

Model	*Train_dataset*	*Pred_dataset*	*Mean Accuracy*	*f-score*
SVM	entry1,entry2,entry3,entry4	entry12	0.796	0.706

Table.6 Selection and setting of transfer learning data sets

Dataset	*Set name*	*Sample counts*
Source	entry1,entry 2,entry3,entry4	71617
Target	part of entry12	11772
Test	The other part of entry 12	7848

Table.7 Comparison of different model results when using multiple data sets

model	*Mean Accuracy*
SVM	0.758
TrAdaBoost	0.975
Dynamic-TrAdaBoost	0.993

4.3. Experimental settings

Traditional machine learning method SVM and AdaBoost-based transfer learning algorithm TrAdaBoost are used as comparison methods for traffic analysis. The training data comes from the source domain dataset and part of the target domain dataset. The test data uses the target domain dataset and the K-fold cross-validation method is used to verify the model. For comparison, the training data of the comparison group experiment only comes from the source field, and the test data are all the data of the target field. The results are shown in Figure 4 and Figure 5. Accuracy and f-score were selected as evaluation indicators.

4.4. Experimental results

To carry out the comparison of multi-facets, SVM is first used as the training model. If both the training data and the test data come from the target field, the accuracy of the prediction reaches 0.885, which is due to the insufficient amount of data in the target field. When the training data is all from the

source domain and the test data is all from the target domain, the accuracy rate is only 0.796. This is because the target data is obtained when the network environment changes, and the source and target domains are no longer Obey the same distribution. When using the source domain data set to assist the target domain in training the model, TrAdaBoost improves the accuracy rate from 0.758 to 0.975, while the accuracy rate of Dynamic--TrAdaBoost reaches 0.993, integrates dynamic $N \to \infty$ and ignores each base When the classification error on the source domain data set is obtained, all the source domain samples are correctly classified

5. Conclusion

It is a good idea to apply the knowledge of the source domain to the target domain using the method of transfer learning, but there are many problems in this process. In this paper, we use the Dynamic-TrAdaBoost method that introduces a modification factor to solve the problem of weights falling too fast in the source domain, and the problem of not considering the error rate of the base classifier in the source domain. Help the training classification of the target domain. The experiments in the data set prove the superiority of our introduction of this method. In the future, we will continue to improve this method to use distributed to speed up model training and solve the problem of label imbalance.

The machine learning algorithm is used to solve the classification problem in a specific network environment, and the problem of the distribution of test environment data and the training environment data is not the same as the concept shift and the increase of new applications [14], Using transfer learning to solve the problem of different data distribution in the source domain and the target domain. Most of the research at this stage is based on such a mapping from a single source domain to a single target domain. The labeled data of the data set in the actual environment sometimes comes from multiple source domains [15]. The target domain is similar to multiple source domains [16], and the multiple source domains are also different. The data distribution of different data is multiple sub-domains under one large domain.

Acknowledgments

This work was partly financially supported through grants from the CERNET Innovation Project (NGII20180409). The authors thank the anonymous reviewers for their helpful suggestions.

References

[1] Zou Tengkuan, Wang Yuying, Wu Chengrong. A review of network background traffic classification and recognition research [J]. Journal of Computer Applications, 2018: 0-0(in Chinese).

[2] Cheng Guang, Chen Yuxiang. Encrypted traffic recognition method based on support vector machine [J]. Journal of Southeast University: Natural Science Edition, 2017, 47 (4): 655-659(in Chinese).

[3] Rezaei S, Liu X. Deep learning for encrypted traffic classification: An overview[J]. IEEE communications magazine, 2019, 57(5): 76-81.

[4] Wang P, Chen X, Ye F, et al. A Survey of Techniques for Mobile Service Encrypted Traffic Classification Using Deep Learning[J]. IEEE Access, 2019, 7: 54024-54033.

[5] Aceto G, Ciuonzo D, Montieri A, et al. Mobile encrypted traffic classification using deep learning: Experimental evaluation, lessons learned, and challenges[J]. IEEE Transactions on Network and Service Management, 2019.

[6] Chowdhury S, Liang B, Tizghadam A. Explaining Class-of-Service Oriented Network Traffic Classification with Superfeatures[C]//Proceedings of the 3rd ACM CoNEXT Workshop on Big DAta, Machine Learning and Artificial Intelligence for Data Communication Networks. ACM, 2019: 29-34.

[7] Lim H K, Kim J B, Heo J S, et al. Packet-based Network Traffic Classification Using Deep Learning[C]//2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIC). IEEE, 2019: 046-051.

[8] Shapira T, Shavitt Y. FlowPic: Encrypted Internet Traffic Classification is as Easy as Image Recognition[C]//IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). IEEE, 2019: 680-687.
[9] Sun G, Liang L, Chen T, et al. Network traffic classification based on transfer learning [J]. Computers & electrical engineering, 2018, 69: 920-927.
[10] Zhuang Fuzhen, Luo Ping, He Qing, et al. Research progress of transfer learning [J]. Journal of Software, 2015, 26 (1): 26-39(in Chinese).
[11] Dai W, Jin O, Xue G R, et al. Eigentransfer: a unified framework for transfer learning[C]//Proceedings of the 26th Annual International Conference on Machine Learning. ACM, 2009: 193-200.
[12] Al-Stouhi S, Reddy C K. Adaptive boosting for transfer learning using dynamic updates[C]//Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer, Berlin, Heidelberg, 2011: 60-75.
[13] Moore A, Zuev D, Crogan M. Discriminators for use in flow-based classification[R]. 2013.
[14] Zhang J, Chen X, Xiang Y, et al. Robust network traffic classification [J]. IEEE/ACM Transactions on Networking (TON), 2015, 23(4): 1257-1270.
[15] Duan L, Tsang I W, Xu D. Domain transfer multiple kernel learning[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2012, 34(3): 465-479(in Chinese).
[16] Zhang Qian, Li Haigang, Li Ming, et al. Instance transfer learning method based on multi-source dynamic TrAdaBoost [J]. Journal of China University of Mining & Technology, 2014, 43 (4): 713-720(in Chinese).