Internet Traffic Classification Using Feed-forward Neural Network

Wengang Zhou†*, Leiting Dong▾, Lubomir Bic*, Mingtian Zhou†, and Leiting Chen†
† School of Computer Science & Engineering
University of Electronic Science and Technology of China,
Chengdu, Sichuan, China, 610054
*Donald Bren School of Information & Computer Sciences
University of California, Irvine, CA 92697, USA
▾Henry Samueli School of Engineering
University of California, Irvine, CA 92697, USA

***Abstract*- Many network activities can benefit from accurate traffic classification and categorization, such as QOS control, network security monitoring, and traffic accounting. In this paper, a new approach based on feed-forward neural network is proposed for accurate traffic classification, which eliminates the disadvantages of port-based or payload-based classification methods. Extensive experimentation and comparison have been carried out to explore this new approach; it has been found out that, combined with a fast correlation-based feature selection filter, better performance and more accurate classification results can be obtained using neural network method compared to other techniques. For its good performance and elimination of accessing the contents of the packets, the proposed technique is expected to have a promising application prospect in internet traffic classification.**

I. INTRODUCTION

In recent years, traffic classification and categorization of internet flows has become a hot topic with a wide range of applications. Accurate traffic classification can quickly identify malicious flows and contribute to the control of network attacks. Besides, the accurate identification of different types of network flows can be helpful to the full use of network resources, which leads to a more optimized performance of the overall network services. Furthermore, traffic classification technologies can help network operators track the traffic changes and analyze the different requirements with different network applications, which meets the diverse needs of network users.

The most common identification method of network applications depends on the use of well-known ports: The Internet Assigned Numbers Authority (IANA)[1] provides Port mapping table, network traffic of a particular port belongs to a particular network application. However, some applications have no ports that are assigned or registered in IANA. Furthermore, many network applications, such as P2P and passive FTP, use a large number of random ports unknowable in advance for data transmission. So an exact identification based on well-known ports has become ineffective.

The research is supported in part by the 11th Five-Year National Science and Technology Surporting Item- Key technologies research of International trade regional economic cooperation and circulation promotion.(2009BAH46B03）

Subsequently, a payload-based analysis technique[4] is used in many cases. The approach analysis packet payloads to find out whether this packet contains the particular signatures of known applications, thus determine which category the packet is. However, this technique can only match the existing characteristic signature[3] of internet applications and is unable to identify any other traffics. As the increasing number of new applications and the growing popularity of packet loads encryption, the effectiveness of this method is gradually declining. In addition, direct analysis of session and application layer content may result in violation of the relevant privacy regulation.

A more reliable technique for the identification of network applications is the use of machine learning approach[12]. Because of the proved efficiency, accuracy and robustness of neural network classification methods in other research and industrial applications, in this paper, a feed-forward neural network-based classifier is proposed to serve as the machine learning algorithm for traffic classification. Several numerical experiments are conducted to evaluate the performance of proposed technique. From experiment results, one can see this technique gives good and robust classification while other techniques may suffer from inappropriate assumptions.

The remainder of this paper is organized as follows: Section II briefly review previous methods of traffic classification. Section III analyses the classification technologies used in the proposed neural network classifier. Section IV shows collection and composition of the training and test data sets. Section V describes the relevant classification assessment criteria and assessment strategies. Our experiment and test results are outlined in section VI. Section VII concludes this paper and discusses the future work.

II. RELATED WORKS

Zuev and Moore et al [2] applied a probabilistic model-based Bayes method to traffic classification. This method requires independence between each of the properties and assumes all features are subjected to the Gaussian distribution, but the original network flows are difficult to meet these restricted conditions, therefore, the problem significantly reduces the overall accuracy of classification.After combining with FCBF(fast correlation-based filter) algorithm and kernel

estimation technology[2], the accuracy of classification has been greatly improved.

Karagiamnis et al [5] used a set of various behavioral characteristics to classify all types of traffic. This method focuses on the higher level characteristics, including the social, functional, and application behaviors, and do not just use the characteristics of transport layer. Because this method relies on some specific attributes of network environments, the performance of classification may be unstable in different network environments.

Roughan et al [6] introduced two simple machine learning approach, including the K-nearest neighbor(referred as K-NN) and linear discriminant analysis(referred as LDA), to handle the traffic classification. However, K-NN method requires calculating individually the similarity between the training samples and the unidentified traffic flows. Not only this procedure has large computational overhead, but the classification performance is easily interfered by the noise data. LDA methods usually need complex preprocessing operation and lead to a lot of extra overhead.

Erman et al [7] presented a clustering method to handle the traffic classification problem, an unsupervised clustering approach is expected to be able to identify new network applications by review the relations in some new clusters. However, the efficiency is limited for the use of this method requires mark these clusters by hand in order to obtain classification results of network traffic.

Wei li et al [8] applied C4.5 decision tree algorithm to identify internet traffic. In this method, a classifier model was built by using entropy of training date set, and flows were identified by searching the decision tree. In the meantime, this paper also looked at the connection among accuracy, completeness, latency and throughput.

III. CLASSIFICATION TECHNOLOGIES

A. *Structure of neural network*

Neural network is a complicated system that is highly nonlinear by simulating human nervous system. In fact, neural network consists of a large number of neurons with certain type of structure and tries to model various phenomena of the real physical world. In this paper, a feed-forward neural network[9] is used for traffic classification. Although the

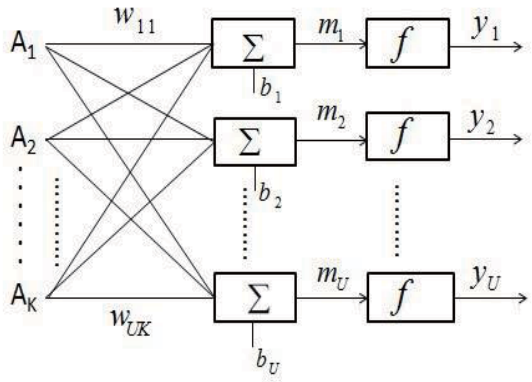

Fig.1. One hidden layer network with K input elements and U neurons

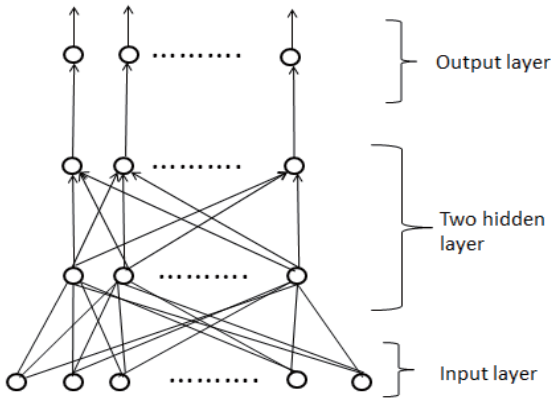

Fig.2. A feed-forward neural network with four-layer

structure of feed-forward neural network is adjustable to satisfy different practical requirements, it is usually composed of the input layer, hidden layer and output layer. Fig. 1 shows the structure of a single hidden layer neural network. Typically, the transfer function f is determined by the user, while other parameters are determined by training.

In fig.1, $A_j\left(1 \le j \le K\right)$ is the input of this layer, $w_{ij}\left(1 \le i \le U; 1 \le j \le K\right)$ is the weight between corresponding neurons, $b_i\left(1 \le i \le U\right)$ is the bias added to the neuron i, f is the transfer function of this layer, $y_i\left(1 \le i \le U\right)$ is the output of the corresponding neuron, $m_i = w_{i1}A_1 + w_{i2}A_2 + ...+ w_{ik}A_k + b_i, (1 \le i \le U)$, $y_i = f(m_i)$.

When various layers with structure described by Fig. 1 are put together, with the output of previous layer as the input of the next layer, a feed-forward neural network is then constructed. For example, Fig. 2 shows the schematic drawing of a typical four-layer feed-forward neural network.

After its initial configuration, the neural network needs to be trained according to the feedback (the corresponding output for the input data) through adjusting the value of the weight matrix w and bias b . While different training methods are available, in order to avoid over-fitting, a Bayesian regularization method is adopted here to optimize network parameters as well as to retain good generalization.

B. *Bayesian regularization*

It is desired that a neural network trained on one dataset would also perform well in another dataset. Therefore, Bayesian regularization is used as training technique here for this purpose. Bayesian regularization technique minimizes a linear combination of squared errors and squared network parameters to prevent the model from over-fitting for datasets. The objective function [10] is described as follows:

$$F = \alpha E_D + (1-\alpha)E_W$$

where E_D is the sum of squared errors, which can be calculated as

$$E_D = \sum_{i=1}^{n} (d_i - y_i)^2$$

and E_W is the sum of squares of the network weight and biases, which takes the form:

$$E_W = \sum_{i=1}^{U} w_i^2$$

α is the parameter indicating relative weighing factor between E_D and E_W

In Bayesian framework, the weights and biases of the network are assumed to be random variables with specified distributions. The regularization parameter α is related to the unknown variances associated with these distributions. It is therefore estimated using statistical techniques. And weights and biases are determined by optimization procedures. More details are given in [10].

C. FCBF (Fast Correlation-Based Filter) algorithm

FCBF algorithm uses symmetrical uncertainty (SU) as the goodness measure of features and selects good attribute sets for classification based on the correlation analysis of features and classes, which consists of two steps[11]:

(1) distinguishing the relevance between a feature and a type of class.
(2) distinguishing the redundancy between a feature and other relevant features.

In FCBF, C-correlation (the correlation between a feature and a class) [11] is used to initially select some features that have been highly correlated with the class, and then, F-correlations (correlations between features) [11] are utilized for further selection to obtain an effective subset of features for the neural network.

Once the FCFB(Fast Correlation-Based Filter) algorithm is applied and the most important features are chosen, the training and testing data sets are used to train the neural network, and the performance of the corresponding model can be evaluated.

D. Naive Bayes classifier

Naive Bayes is a simple probabilistic classifier based on Bayesian theorem and certain assumption regarding the distribution of features in each class. For a given set of possible flow classes $C = \{C_1, C_2, ...C_j .....C_n\}$ and an arbitrary network flow x, the conditional probability of class C_j for the given flow x can be described by Bayes' theorem:

$$P(C_j \mid x) = \frac{P(x \mid C_j) P(C_j)}{\sum_{j=1}^{n} P(C_j) P(x \mid C_j)} \quad (1)$$

where $P(C_j)$ is the priori probability of class C_j

$P(x \mid C_j)$ is the conditional probability of flow x given it belongs to class C_j

Since the network flow x can be abstracted as attributes vector $(A_1, A_2, ...A_k)^T$, Naive Bayes classifier makes certain assumptions that all feature properties are independent and are subjected to Gaussian distribution. Then the conditional probability can be written as follows with only mean value and standard deviation as uncertainties:

$$P(x \mid C_j) = \prod_{i=1}^{k} P(A_i \mid C_j) \quad (2)$$

And then, equation (1) can be written as:

$$P(C_j \mid A_1, A_2, ...A_k) = \frac{\prod_{i=1}^{k} P(A_i \mid C_j) P(C_j)}{\sum_{j=1}^{n} \prod_{i=1}^{k} P(A_i \mid C_j) P(C_j)} \quad (3)$$

Therefore $P(C_j \mid A_1, A_2, ...A_k)$ can be calculated and the traffic flow is classified such that the highest posterior probability is obtained.

However, in practice, the assumptions of feature independence and the Gaussian distribution are not always appropriate. For example, the length of whole packet is equal to the length of the packet header, coupled with the length of packet payload. Clearly there is a strong correlation between these features. To make the feature independence assumption appropriate, the Naive Bayes approach is also used together with FCFB (Fast Correlation-Based Filter) algorithm by Moore and Zuev[2], which improved the overall accuracy of the classification. On the other hand, Moore and Zuev[2] tried to address this issue by using kernel distribution instead of Gaussian distribution assumption. However, it should be noted that this approach greatly increases computation demand of Naive-Bayes classifier and thus is not appropriate for online traffic classification. In this paper, two classification approaches are compared: the application of the feed-ward neural network and the Naive Bayes classifier with Gaussian distribution assumption.

Table.1. STATISTICS OF DATA SETS

Class	Number of flows	Percent
WWW	328091	86.91%
MAIL	28567	7.567%
BULK	11539	3.056%
DATABASE	2648	0.701%
SERVICE	2099	0.556%
P2P	2094	0.555%
ATTACKS	1793	0.475%
MULTIMEDIA	576	0.152%
INTERACTIVE	110	0.029%
GAME	8	0.002%
TOTAL	377526	100%

IV. DATA SETS

The pre-classified dataset described originally by Moore el [13] was used for the evaluation. This datum was randomly sampled in several different periods from one node on the internet. This site was shared by about 1,000 researchers, technicians and management staff of three research institutions, and connected to the Internet through a full-duplex Gigabit Ethernet link. Each full-duplex traffic on this connection was captured in a full 24 hours period, so the original traffic-set contained all full duplex traffic connected the node in both link directions. Since the original traffic-set is too large, Moore divides it into ten subsets by randomly sampling method. The sampling time of each subset is almost the same (approximately 1680 seconds each) and these non-overlapping random samples are uniformly distributed over the 24-hour interval. The number of flows and the proportion of the various types of network traffic are shown in Table 1. Further details of the description of the various traffic classes and the definition of flow features are given in [13].

A basic requirement of traffic classification is that the flow types are correctly identified. Table 3 shows the frequently used application classes of the data sets in this study. An application class may contain different kinds of data, for example, the class Mail includes SMTP and POP3. TCP/IP traffic flows are the fundamental objects for classification, which is represented as a flow of one or more packets between two hosts of network using network communication protocols. The flow is clarified by the IP five-tuple consisting of the source-IP, destination-IP, source-port, destination-port and the protocol type. In order to focus on the traffic classification process itself, the semantically complete TCP connections are selected to make up the training sets and testing sets, where semantically complete TCP flow is defined as: a bi-directional flow for which one can observe the complete connection set-up(SYN-ACK) and another complete connection tear-down(FIN-ACK). The categories of network applications are shown in Table2.

Table.2. The categories of network applications

Classification	Representative applications
WWW	www
MAIL	smtp, pop2/3,imap
BULK	ftp
DATABASE	sqlnet, oracle, ingres, postgres
SERVICE	dns,ident,1dap,ntp,X11
P2P	Bittorrent, Kazaa, Gnutella
ATTACKS	Internet worm, virus attacks
MULTIMEDIA	Real, Windows media player
INTERACTIVE	ssh, klogin, rlogin, telnet
GAME	Half-life
TOTAL	25 kind of applications

Table.3. Confusion matrix of n class

		Classification Result			
		Class 1	Class 2	.....	Class n
Real Label	Class 1	h11	h12	.....	h1n
	Class 2	h21	h22	.....	h2n
	. . .	. . .	. . .	. . .	. . .
	Class n	hn1	hn2	.....	hnn

The data set is divided into two subsets, Set 1 and Set 2, which are used as training data set and test data set respectively. Apparently, the proportion of each network flow type is consistent with the original data set. There are many redundant feature properties and irrelevant attributes in the sample data sets, which not only decrease the accuracy of neural networks, but also reduce the classification efficiency of neural networks. Therefore, the features of data sets are evaluated and preselected by a Fast Correlation-Based Filter (FCBF) algorithm. The best features with strong correlation between classes but small correlation with other features are selected, and experiments are conducted on these pre-processed datasets

V. EVALUATION METRICS

Usually when machine learning method is used for traffic classification, the classification capabilities of the model shall be estimated for unknown data sets based on the experiment results for test data sets. If the classification model M has been established, together with the test data set $T=\{T_1,T_2,\ldots,T_h\}$ and the class collection $C=\{C_1,C_2,\ldots,C_n\}$, where the network flows T corresponds to the *nth* class of network application, the corresponding confusion matrix can take the form as shown in table 3. From this table may define the following concepts:

(1) TP (True Positive): is the number of the samples which actually have type i among all those correctly classified as type i by the classification model.

(2) FN (False Negative): is the number of the samples which actually have type i among all those classified as another types by the classification model.

(3) FP (False positive): is the number of the samples which do not have type i among all those misclassified as type i by the classification model.

(4) OA (Overall Accuracy): is the proportion of the all instances which is correctly classified as truly types in whole samples.

$$OA=\frac{\sum_{i=1}^{n}TP_i}{\sum_{i=1}^{n}(TP_i+FN_i)}$$

A scale of 0% to 100% is used to represent the accuracy and the overall accuracy. However, overall accuracy is more

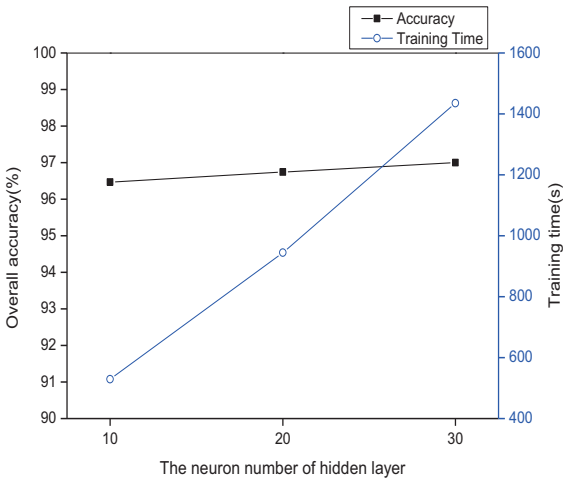

Fig.3 Overall accuracy and training time with different neuron number of hidden layer

frequently used among these metrics, which reflects the overall performance of a classification model.

VI. EXPERIMENTAL RESULTS

In order to evaluate the performance of feed-forward neural network classifier, we design and implement a series of experiments. In these experiments, a typical 3 layer feed-forward network is used and its performance is compared to Naive Bayes classifier.

The platform used for this experiment is: Windows 7 operating system running on a PC computer equipped with an Intel Q8300 2.5GHZ CPU, and 8G system memory.

A. Effect of neuron number of hidden layer

Since neural network is based on a large number of single neurons of the same structure implemented in the hidden layer, it is important to study the effect of the number of neurons in the first place. In this experiment, the performance and overhead of this system were measured and compared for a same training data set while the number of neuron varied from 10 to 30. The results are plotted in Fig. 3, and the standard deviation with various number of neuron is presented in table 4.

In fig.3, X axis represents the neuron number of hidden layer, Y axis on the left represents the overall accuracy of the classification models with different neuron number, the other Y axis on the right represents the training time of this model used in the training process. As can be seen in Fig. 3, with the increase of the number of neurons in the hidden layer, the performance of the neural network has been improved a little, but not much. In the meantime, the training time of this model increased rapidly. Besides, with the increase of the number of neural nodes, the requirements on the computer hardware and the speed of network also became higher. Therefore, in practice, a trade-off has to be made between the accuracy of classification and the overhead of the system. However, in this particular case, since 10 neurons can produce enough accuracy and good robustness, it is chosen as the number of neurons in the hidden layer for the rest experiments.

Table.4. Standard deviation with different neuron number of hidden layer

Neuron number	Standard deviation
10	0.0134
20	0.0174
30	0.0064

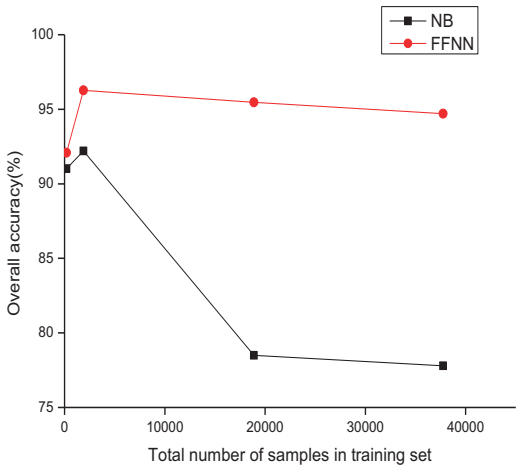

Fig.4 Average overall accuracy with stratified sampling

B. Experiments using stratified sampled subsets as training datasets

As described previously, the whole dataset are divided into Set 1 and Set 2, as training and testing datasets. We firstly selected respectively 0.1% for each type of network applications from set1 (at least 2 samples per class), which constituted the training data set for the experiment. After that, we run the two machine learning algorithms: Naive Bayes method (refer as NB) and the feed-forward neural network (refer as FFNN), and test them by testing data set-Set 2. This experiment is repeated ten times to compute mean value and standard deviate of classification accuracy. Subsequently, the rate of stratified sampling is gradually increased to 1%, 10%, 20%. Each corresponding experiment is repeated ten times, the results from these experiments are shown in Fig. 4 and Fig. 5.

Fig. 4 shows that the traffic classification performance of Naive Bayes method is relatively poor. With the increase of the size of training data set, the average overall accuracy of classification decreased to less than 80%. This is mainly because some properties of the network flow that follow the multi-modal distribution are misrepresented when all the properties are assumed to satisfy the Gaussian distribution. Consequently, the actual features of the network flow are not simulated correctly, and the bad classification is unavoidable. On the other hand, the average classification accuracy of neural network approach remains high and stable, which is highly desired for traffic classification implementation. Fig. 5 further clarifies the changes of standard deviation with the size change of the training data set in the way of the stratified sampling.

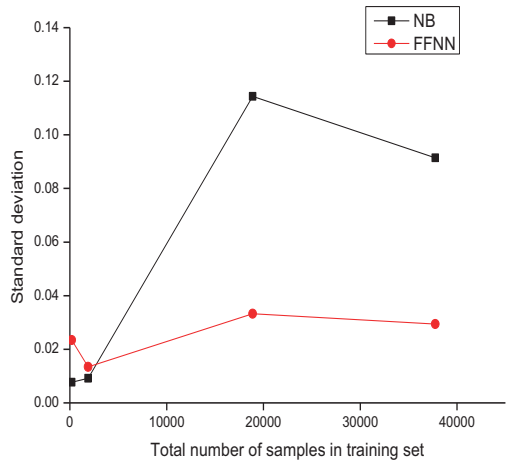

Fig.5 Standard deviation with stratified sampling

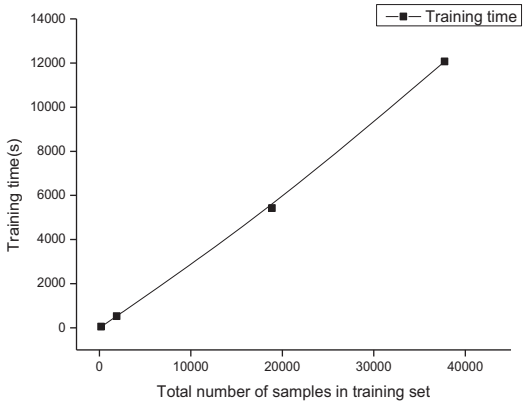

Fig.6 Training time of the neural network with stratified sampling

As can be seen clearly in Fig. 5, compared to the Naive Bayes method, the robustness of the feed-forward neural network approach is better. In the experiments, the fluctuation of the standard deviation value is small for the trained neural network model, which means the performance of classification is more stable and the reliability of this method is expected to be higher in practice.

Figure 6 shows the training time of neural network with stratified sampling. From which it can be seen, the training time increases rapidly at almost an exponential rate with the sampling ratio. However, in practical implementation, the training process can be conducted off-line, thus it does not increase the burden of hardware and internet in the testing process.

VII. CONCLUSION AND FUTURE WORKS

Classification of network traffic using machine learning methods has become prevailing in network measurement. For its efficiency and simpleness, the Naive Bayes method has been considered as a better approach than the port-based or payload-based methods for traffic classification. But its inherent deficiencies are also obvious; particularly, the accuracy and performance are limited due to its inappropriate inherent assumptions. In this paper, the feed-forward neural network is proposed for traffic classification. Comparison of experimental results suggests that this new method has a more stable and robust performance than the Naive Bayes methods does. In addition, the proposed method also does not access the contents of packets, which means this method has wider application in dealing with traffic classification problems. Hence the feed-forward neural network is expected to a promising technique for traffic flow classification.

With the expansion of the internet usage, the statistical features of different network flow classes may vary with time. Therefore, using data sets acquired at a later time to update the model of classification is expected to enhance the unbiasedness and robustness of the neural network in a long period. This research is a valuable area for future work. In addition, Combining neural network with unsupervised clustering techniques to automatically identify and classify new traffic classes in the future shall also be worthy of great effort.

REFERENCES

[1] IANA, http://www.iana.org/assignments/port-numbers (as of May 2011)
[2] A.W. Moore, D. Zuev. “Internet traffic classification using Bayesian analysis techniques”. In: Proc. of the 2005 ACM SIGMETRICS Int’l Conf. on Measurement and Modeling of Computer Systems. Banff, 2005. 50−60.
[3] P. Haffner, S. Sen, O. Spatscheck, and D. Wang. “ACAS: Automated construction of application signatures”. In SIGCOMM’05 MineNet Workshop, Philadelphia, August 22-26, USA, 2005.
[4] A. W. Moore and D. Papagiannaki. “Toward the accurate identification of network applications”. In Proceedings of the Sixth Passive and Active Measurement Workshop (PAM 2005), March 2005.
[5] T. Karagiannis, K. Papagiannaki, and M. Faloutsos. “BLINK: Multilevel traffic classification in the dark”. In: Proc. of the ACM SIGCOMM. Philadelphia, 2005. pp. 229−240.
[6] M. Roughan, S. Sen, O. Spatscheck, and N. Duffield, “Class-of-service mapping for QoS: A statistical signature-based approach to IP traffic classification”. In ACM SIGCOMM Internet Meas. Conf., Sicily, Italy,2004, pp. 135–148.
[7] J. Eerman, A. Mahanti, M. Arlitt. “Internet traffic identification using machine learning techniques”. In: Proc. of the 49th IEEE GLOBECOM. San Francisco, 2006.
[8] L. Wei, A. W. Moore,. “A machine learning approach for efficient traffic classification”. In: Proc. of the 15th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, Istanbul, Turkey October 24-October 26, 2007
[9] C. M. Bishop, Neural Networks for Pattern Recognition. London, U.K.: Oxford Univ. Press, 1995.
[10] F.D. Foresee, M.T. Hagan, “Gauss-Newton approximation to Bayesian regularization”. In: Proceedings of the International Joint Conference on Neural Networks, 1997, pp. 1930–1935
[11] L. Yu and H. Liu, “Feature selection for high-dimensional data: a fast correlation-based filter solution”. In: Proc. of the Twentieth International Conference on Machine Learning (ICML), 2003, pp. 856–863.
[12] S. Zander, T. Nguyen, and G. Armitage, “Automated traffic classification and application identification using machine learning”. In: Proc. of the 30th annual. IEEE conference on local Computer networks. (LCN 30), Sydney, Australia, Nov. 2005, pp. 220–227.
[13] A. W. Moore and D. Zuev. “Discriminators for use in flow-based classification”. Technical report, Intel Research, Cambridge, 2005.