Identification of the local area network using machine learning

I V Voronin, A I Gazin, V S Ziyautdinov, T A Zolotareva, D M Skudnev and O V Selishchev

Lipetsk State Pedagogical P Semenov-Tyan-Shansky University, Lipetsk, 398020, Russia

E-mail: ole9153@yandex.ru

Abstract. This article discusses the use of a network traffic analyzer based on a neural network to detect and resolve problems that occur in local area networks. Several packet capture approaches have been investigated that affect the speed and accuracy of network traffic analysis. Various methods for classifying network traffic are given, special attention is paid to the machine learning method. The main advantages and disadvantages of various teaching methods have been identified. The process and basic steps of clustering the training set are described. As a result of the study, a method was identified for creating a high-quality system that allows identifying the state of a local area network.

1. Introduction

Currently, there are a large number of different methods and tools based on them helping to detect and fix problems that arise in the local area network (LAN). The burden of identifying network problems lies with the organization’s system administrator. To do this, he needs to timely accumulate relevant data on the operation of the network, analyze them, monitor the operation of network equipment included into LAN. All this work occupies most of the working time of personnel engaged in the maintenance of the organization’s IT infrastructure. In order to optimize the system administrator’s operating time, as well as to reduce the time required to perform routine operations, it is necessary to use modern technologies in the workplaces of specialists involved in network analysis. These technologies should be closely integrated with a modern decision support system (DSS).

2. Generalized scheme of network traffic analysis algorithms

When studying methods of analyzing network traffic, a sequence of steps was identified, each of which leads to an increase in the level of representation of the object of analysis.

First of all, a packet is captured, which passes through a network connection that is under control. As a result of this step, the analyzed object is received in the form of network packets. There are several packet capture approaches that affect the speed and accuracy of the analysis, and depend on the available processing power.

- For a quick traffic analysis, an approach called slicing is used. With this approach, only the data stream prefix will be exposed to analysis. This leads to a very fast analysis, at minimal...
cost. But at the same time, the quality of the analysis suffers, therefore this method is used only for traffic classification by protocols.

- If traffic type monitoring is necessary, then a method is used in which only certain data packets are captured that are selected for a specific attribute. This approach is called sampling.

- If the task is to accurately analyze traffic, then interception of all packets is necessary. This approach is called deep packet capture (DPC). This approach is used in cases of network security.

After the packet is captured, it is aggregated into streams. The result of this step is to obtain a new object for analysis which is data flow.

The final step is to classify the stream. After this operation is completed, it becomes possible to further process the obtained object, the specific form of which depends on the applied task.

3. Network traffic classification

The task of classifying network traffic can be defined as a large number of data streams in $P = \{x_1, x_2, ..., x_n\}$ network, where any data stream in $x_i$ network can be characterized by various attributes. In addition, the network data stream is characterized by traffic classes. At the same time, the attributes of the data of network packets are most often taken as the stream size, average packet length, or data packet transmission duration, while various information transfer protocols (FTP, HTTP, etc.) are used as traffic.

Network traffic can be classified using various methods. One widely used classification method is machine learning. This method has become widespread due to the ability to continuously acquire new knowledge or transform the structure of knowledge. In the process of creating network traffic classification models, the data used for training is the basis for creating classifiers.

Logically, the machine learning procedure can be divided into the process of creating a classification model and the classification process. Machine learning methods are divided into teaching methods with a teacher and teaching without a teacher [1, 4]. For training with a teacher, it is necessary to create a knowledge structure first that will subsequently be used to classify new patterns. Thus, training is reduced to submitting to the machine input a set of typical examples, the belonging of which to certain classes is known in advance.

As a result of such a teaching process, a model is constructed with the teacher for classification based on the analysis and generalization of the presented samples, i.e. a model is created for the relationship of input and output.

The main disadvantage of learning with a teacher is the inability to detect new applications, due to the lack of a learning set in the knowledge base. If there is a need to identify new applications in the classification of network traffic, then it is advisable to use a teaching method without a teacher. When classifying network traffic methods without a teacher do not need initial manual marking of the input data, they are only based on the similarity between the classified objects and the statistical characteristics of the network data stream are used as input. This method will allow grouping newly identified applications into a cluster.

Let us describe the clustering process of the training set: there are a lot of $P = \{x_1, x_2, ..., x_n\}$ network data streams, the number of $N$ clusters, into which the streams need to be divided, is also known. As a result, it is necessary to determine $x : P \rightarrow \{K_1, K_2, ..., K_N\}$, where $K_i$ is a cluster, and the flow should be assigned to only one $1 \leq i \leq N$, $D = \bigcup_{j=1}^{N} K_j$, $K_i \cap K_j = \emptyset$ and $\forall i \neq j$ cluster.

The goal of clustering is to split the network stream into $N$ clusters optimally, with the similarity metric selected for clustering. To determine the optimality of clustering, the root-mean-square error of splitting sweets into clusters should be minimized.
There are various similarity metrics. The metric can be selected based on the spatial arrangement of objects or other characteristics characterizing the clusters implicitly. $x_i$ and $x_j$ flows are at a distance, which is determined as the result of applying a certain metric in the space of characteristics. To determine the distance separating such $x_i$ and $x_j$ flows, it is advisable to use the Euclidean metric:

$$l(x_i, x_j) = \left( \sum_{k=1}^{p} (x_{ik} - x_{jk})^2 \right)^{1/2} \quad (1)$$

Based on equation (1), it can be seen that with increasing Euclidean distance, the similarity between two selected flow vectors decreases.

Among the many clustering algorithms, it is convenient to use the fast and simple k-means algorithm. This algorithm is excellent for clustering traffic transmitted over a local computer network. The main steps of clustering according to the k-means algorithm are as follows:

- A random $\mu_i$ cluster center is selected from the training set, where $i=1,2,\ldots,k$.
- A network data stream with similar properties is found and added to the cluster.
- Based on the newly added network flows, the cluster centers are recalculated, after which, based on the new centers, the flows are redistributed.
- The criteria for stopping the algorithm are calculated; if the criteria are not met, a return to step 2 occurs.

The criterion for stopping the algorithm is the minimum change in the mean square error of the partition:

$$E = \sum_{i=1}^{k} \sum_{j=1}^{n} l(x_j, \mu_i) \quad (2)$$

The main drawback of the k-means algorithm is the sensitivity to the initial settings. For example, if it is incorrect to select the initial center of the cluster, then instead of the global optimum, a local optimum will be found. As a result, the k-means algorithm must be repeated many times to obtain a reasonable separation of flows. This leaves its mark on the speed of the algorithm.

4. Neural network traffic analyzer

The most common methods of machine learning are in the application of artificial neural networks. Artificial neural networks (ANNs) allow solving problems in the field of processing and recognition of various images more efficiently than classical approaches [2]. For example, one of the first approaches for detecting and classifying computer network problems is signature analysis. Its basis is the finding of matches of the found sequence with the base sample, by bitwise comparison. Thus, you can find a signature indicating the presence of harmful code in the processed traffic.

At the same time, in order to use ANNs in the tasks of analyzing the traffic of computer networks effectively, one just need to qualitatively train ANNs, i.e. to get the opportunity to identify all the problematic events correctly that occur during the classification of network traffic.

To solve this problem, one can use Hamming ANN. It can be used as a solution to the problem of classifying binary vectors. The basis of this ANN work includes the procedures that are aimed at finding a reference image among all the submitted noisy input vectors.

Hamming ANN is used to determine whether an object belongs to a particular class, which is defined by $X$ vector. This vector has bipolar features that can take values 1 and -1, and has $N$ dimension. It is assumed that there are $M$ classes, each of which is characterized by its own representative - $X_v, v = 1,2,\ldots,V$ object [1].
The data is based on images of reference vectors and feature vectors selected by experts and which correspond to the selected images. The Hamming neural network analyzer processes the data according to the scheme shown in figure 1.

Hamming ANN consists of $N$ inputs to which bipolar features of the object are supplied. Further, the processing of the obtained characteristics takes place, after which one of $K$ outputs is activated, indicating a certain class to which the object presented at the input belongs.

**Figure 1.** Data processing scheme in a Hamming neural network analyzer.

To attribute a specific object to the desired class, the square of the distance between $X$ and $X_q$ vectors is calculated by the formula:

$$R(X, X_q) = \sum_{j=1}^{N} (x_j - x_{qj})^2$$

where $q=1,2,...,Q$, $x_j$ and $x_{qj}$ are bipolar signs of the input image and standard $q$, $j = 1, N, q = 1, Q$.

If $\min_q R(X, X_q) = R(X, X_{q_0})$, then ANN will classify the object received at the input to $q$ class.

Having performed $R(X, X_q)$ transformations, we can determine that instead of finding $k$ minimizing of $R(X, X_q)$ by $k$ index, which is the number of the standard, we can use the maximization of the scalar product of $X$ and $X_q$ vectors: $\max_q R(X, X_q^T) = X \cdot X_q^T$.

The scalar product of two vectors can realize a neuron with a potential defined by the formula

$$p_k = \sum_{i=1}^{N} x_i x_{ki} = \sum_{i=1}^{N} x_i w_{ki},$$

where $w_{ki} = x_{ki}$ is the synaptic coefficient of the $k$-th neuron.

So, the form of synaptic neuron coefficients is used to store the characteristics of reference objects (images) in the Hamming ANN. Thus, in order to apply Hamming’s neural network classifier in practice, it is necessary to convert all features of an object represented by integers or real numbers into a bipolar code. After that, $N$ will denote the total number of bipolar digits of the code of all signs used for recognition.

The number of neurons in the Hamming ANN depends on the number of reference images stored in the database, and we will denote this number as $M$. Let us suppose that writing a new image to the database with reference images is accompanied by a detailed calculation of the feature vector. This
means that, in addition to the images themselves, the base of reference images also contains vectors of real numbers, calculated by the formula \( X_q = \{ x_{q1}, x_{q2}, \ldots, x_{qN} \} \), where \( q = \overline{1,M} \).

To apply the Hamming ANN for the analysis of network traffic successfully, it is necessary to present real signs in the form of binary code. Let us suppose that the length of \( x_{qi} \) feature code, for \( q = \overline{1,M} \) and \( i = \overline{1,N} \), is \( J \). The total number of binary features that enter the Hamming ANN input can be denoted as \( N_0 = JN \). We will denote the binary signs \( y_0 \) as \( i = \overline{1,N_0} \). Each of the \( M \) neurons of the Hamming ANN has \( N_0 \) synaptic coefficients that represent the feature code of the corresponding reference image in the database. Let us note that the basis of the reference images are not codes, but real numbers \( x_{qi} \), but \( q = \overline{1,M}, i = \overline{1,N} \). Codes are expressed in the operation of the Hamming network. This approach makes it possible to use any of the available coding schemes.

5. Conclusion
The analysis of network traffic allows determining the statistical parameters of the computer network functioning, it also allows getting information about users in the network. The machine learning method is used as a method for creating a high-quality system that allows identifying LAN state; and a neural network network traffic analyzer based on an optimized Hamming neural network is used as a self-learning model [5].

Acknowledgements
This work was carried out in the framework of RFBR grant No. 19-47-48002.

References
[1] Merkov A B 2011 Image recognition. Introduction to statistical training methods (URSS editorial)
[2] Rapoport G N 2011 Biological and artificial intelligence. Consciousness, thinking and emotions (Librocom)
[3] Zolotareva T A 2015 Algorithmic implementation of an intelligent decision support system for identifying the state of a local area network Vestnik of RGRTU 51
[4] Ziyautdinov V S and Kornev P A 2009 Use of neural networks in reconstructing the shape of periodic signals Reports of Tomsk State University of Control Systems and Radioelectronics 2(20) 59-64
[5] Ziyautdinov V S, Zolotareva T A, Voronin I V and Skudnev D M 2016 Analytical support of an intelligent decision support system for identifying the state of a local computer network Fundamental Research 10(2) 280-4