Application of machine learning in the classification of traffic in telecommunication networks: working with network modeling systems

Anton

Abstract. The issues of classification of online traffic in the framework of the work of network infrastructure modeling systems are considered. The main classifiers C4.5 Decision Tree, Random forest Method, SVM, KNN are considered. The parameters responsible for the speed of the platform are substantiated. The 8CoS model is described. The parameters Accuracy, Sensitivity, Specificity are defined. As part of load testing, a method with the least load on the computing power of the platform, C4.5, was identified. The parameters of the model building time and the general processing time for the case with the number of classification instances up to 2000 are determined. The points at which the C4.5 model gives advantages are identified. Each method was evaluated in terms of classification accuracy and processing time. C4.5 achieved a high percentage of accuracy - 98% with a CPU load of 23.

1 Introduction

Technologies of work in VEB are developing rapidly. It is difficult to imagine a company whose work is built without a corporate portal, and a specialist without access to the Internet. As part of the work, data is transmitted via the TCP and UDP protocols; without effective monitoring, auditing and intelligent control over traffic, it is impossible to ensure safe operation within telecommunication networks. To model telecommunications networks, a large number of applications are used, for the traffic of which it is necessary to carry out a classification process, to manage network resources and detect anomalies. This approach allows you to form a strategy for working with specific applications and their audit in the network. Study of traffic at the application level, Internet traffic is a product of complex multicomponent systems and includes components generated by network equipment, operating systems, and applications [1]. Each application operating on the network can be attributed to one of the basic categories [2] shown in Figure 1. Our work will take place within the framework of a traffic classification model based on 8 blocks. This model is an improved model of the 5CoS model [3]. For real-time traffic classes, the 5CoS model defines a division into voice and video traffic. This traffic class has...
the highest priority. Infrastructure management traffic is defined under Critical traffic. The best effort class is split into default for all data traffic. The scavenger and call signaling classes are unchanged.

Fig. 1. Traffic classification model based on 8 blocks

It must be taken into account that different applications have different characteristics and can directly or indirectly affect the operation of the network, determine the order of transmission and the formation of traffic flows. As part of the experiment, a classification model was tested that provides real-time classification of telecommunications network traffic with an accuracy of 99.7-99.9%. The experiment is carried out within the REMOTETOPOLOGY telecommunications infrastructure modeling system using the behavioral decision analysis algorithm based on the C4.5 decision tree algorithm [4,5]. Our approach differs from classical approaches to traffic classification in that there is no binding to OSI model layers 4 and below.

Increasing the speed of work is based on the rejection of a deep analysis of the "payload" of the transmitted data. In our work, we consider an approach to machine learning and traffic classification for real-time operation. Within the framework of the corporate infrastructure, our task is to demonstrate the possibility of obtaining a result with high accuracy and the ability to provide a high speed of the system.

The literature review revealed a large number of different methods for solving the problem of traffic classification. Most modern approaches are focused on the classical version, when working with the application type, port number, IP addresses, and so on [6,7]. Part of the work considers traffic analysis based on IP addresses and pre-prepared signatures. Such approaches allow solving intrusion detection problems by making intrusion detection systems faster and lighter [8,9].
In [10], the authors consider logistic regression (LR), decision tree (DT), random forest (RF), gradient boosted tree (GBT), and linear support vector machine (SVM) algorithms to detect SYN-DOS attacks on systems. The results showed that all algorithms used led to high identification accuracy more than 99%.

In [11], the authors used a K-means approach based on relative density and distance, which had a slightly higher recognition rate than other methods, and shows the possibility of streaming traffic classification.

In [12], the authors, using an intrusion detection system (IDS) as an example, describe a new approach to detecting network intrusions using multi-stage pattern recognition with deep learning. In the presented work, network objects are converted into four-channel (red, green, blue and alpha images) images. The images are then used for classification to train and test the pre-trained ResNet50 deep learning model. The proposed approach is evaluated using two publicly available control datasets, UNSW-NB15 and BOUN Ddos. On the UNSW-NB15 dataset, the proposed approach achieves 99.8% accuracy in detecting a general attack. In the BOUN DDoS dataset, the proposed approach provides 99.7% accuracy in detecting a DDoS attack and 99.7% accuracy in detecting normal traffic.

In [13], the authors propose a work based on a cascaded multi-classifier with two-phase intrusion detection (TP-ID), which can be trained to monitor incoming traffic for any suspicious data. TP-ID applies supervised learning algorithms in two steps. At each stage, a set of classifiers is checked, the best of which is selected to build the model. System performance is evaluated using detection rate, false positive rate, mean absolute percentage error, and classification rate. The proposed approach classifies traffic anomalies with a detection rate of 99%, 0.43% FPR, and a 99.51% correct classification rate.

In [14], the authors solve the problem of monitoring and identifying abnormal traffic in the network. In the paper, traffic characteristics were determined using a method based on mutual information, and then a comparison of mathematical models was made, including k-nearest neighbor (KNN), back propagation neural network (BPNN) and Elman. The parameters were then optimized using the Grasshopper Optimization Algorithm (GOA) based on BPNN and Elman defects to obtain GOA-BPNN and GOA-Elman models. The GOA-Elman model showed the best performance in traffic recognition with 97.33% accuracy, and also performed well in traffic classification.

In [15], the authors consider the use of big data analysis and cloud computing technology to ensure network security. The paper introduces a technology for collecting data about anomalous network behavior, and also discusses the Flume data collection component and the distributed Kafka technology; the data processing process and the corresponding algorithm for processing anomalous network behavior are analyzed. The accuracy of detecting anomalous traffic on the network using big data is 96.4%, and the false positive rate is 2.23%.

Based on the review, it can be assumed that a combination of a limited number of network flow characteristics has the necessary properties for traffic classification and detection of anomalies in the operation of a telecommunications network. In addition, this study shows the possibility of reducing the time of data collection. Our work is aimed at the possibility of classifying traffic and detecting anomalies within sessions of work on the corporate portal in the range from 15 minutes to 4 hours.

The results presented in the review can be called incomplete from the standpoint of increasing accuracy. In the works, traffic databases are mainly taken from free access and their analysis is carried out. At the same time, the possibility of improving accuracy by analyzing the traffic of a specific corporate network and reducing the number of parameters is not considered.
Insufficient analysis of flow characteristics in corporate networks and network infrastructure modeling systems. Most of the considered systems were tested in laboratory conditions, without access to the enterprise infrastructure. Speed intrusion detection issues in the traffic flow. In most works, the analysis was carried out with an emphasis on searching for intrusions without analyzing the main data streams, which allows us to speak about the insufficient frequency coefficient of zero false negatives, that is, the detection of traffic classes, and already in the class of legitimate and illegitimate traffic in the stream.

Partial data. Most of the works used standard Internet traffic as objects of classification, which may differ from the traffic of corporate networks and differs from the traffic that passes in network infrastructure modeling systems. Practical criteria for real-time traffic classification systems may vary. First of all, they focus on applications in the corporate network. The main areas of development are to increase the values of four indicators, such as: system accuracy, completeness, delays and throughput in telecommunication networks. The algorithm under consideration can play a key role in optimizing the traffic of network infrastructure modeling systems, which will increase the comfort of work, the quality of training of specialists, and will also allow timely identification of illegitimate users and illegitimate data traffic flows in the system [16].

2 Materials and methods

Our study uses a semi-automated machine learning approach to generate a classifier that will classify traffic into application classes. Before building the final classifier, a number of mechanisms will be applied to select the feature set and classification algorithm [17] and the size of the observation window. The traffic will be divided into classified objects. In our work, the object of the system will be the data flow. It will be defined as a bidirectional session between client and server with the same set of host-IP, client-IP, host-Port, clientPort and timestamp of the first packet header data. The server and client of the flow will be determined based on observation of packet data.

The REMOTETOPOLOGY platform will be used as a platform for the implementation of the experiment. Within this platform, the main client-server interactions will be data transfer within the control protocols: HTTP\HTTPS, SSH, TELNET, VNC, RDP and similar network infrastructure management protocols. These protocols will be defined as control protocols and should have the highest priority. As part of the network infrastructure modeling system, it is assumed that any network protocols supported by the infrastructure can work. The system will collect a set of characteristics of individual network flows with a different subset of features in order to select an algorithm for subsequent classification. The model will be trained on a subset of features and provide a classification of unknown flows.

2.1 Working with data

As part of our experiment, the work of a group of people with REMOTETOPOLOGY is considered. Traffic is collected in the first interval of about 15 minutes and in the second interval of about 4 hours. Traffic is collected using a self-developed monitoring unit located in the server solution. The data received during the week from a group of 30 people performing work on modeling the enterprise network is being processed. Data is processed and classified using 8CoS traffic classification mechanisms. The resulting data sets are about 4 GB of data containing about 2 million packets. Such traffic can be defined as traffic containing a moderately complex combination of applications. Table 1 shows an example of received traffic as a percentage of streams, packets and bytes. It is worth considering.
Table 1. Composition of DATASETS.

<table>
<thead>
<tr>
<th>Class</th>
<th>By flow</th>
<th>By Packets</th>
<th>By Bytes</th>
<th>By Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example application</td>
<td>Set 15</td>
<td>min (%)</td>
<td>Set 4 hours (%)</td>
<td>Set 15 min (%)</td>
</tr>
<tr>
<td>VOICE</td>
<td>0.32</td>
<td></td>
<td>0.12</td>
<td>0.98</td>
</tr>
<tr>
<td>VIDEO</td>
<td>-</td>
<td></td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>CRITICAL DATA</td>
<td>10.86</td>
<td></td>
<td>12.34</td>
<td>10.33</td>
</tr>
<tr>
<td>NETWORK CONTROL</td>
<td>1.03</td>
<td></td>
<td>1.47</td>
<td>12.86</td>
</tr>
<tr>
<td>BEST EFFORT</td>
<td>8.13</td>
<td></td>
<td>10.78</td>
<td>54.56</td>
</tr>
<tr>
<td>DATABASE</td>
<td>76.34</td>
<td></td>
<td>70.84</td>
<td>16.83</td>
</tr>
<tr>
<td>SCAVENGER</td>
<td>2.14</td>
<td></td>
<td>3.08</td>
<td>1.79</td>
</tr>
<tr>
<td>CALL SIGNALING</td>
<td>1.18</td>
<td></td>
<td>1.37</td>
<td>2.65</td>
</tr>
<tr>
<td>Illegitimate traffic (attack traffic)</td>
<td>-</td>
<td></td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

2.2 Online approach to traffic classification

Most of the sources reviewed in the review used the offline traffic approach. As part of the study, we believe that such an approach is impossible, since it will not allow conducting deep analytics based on current data and making decisions in close to real time. Our task is to improve accuracy and reduce delays. Most applications can benefit from early identification of traffic. Our approach considers the characteristics that are taken within the observation window for only a few packets. In this case, we may not take into account the entire dataset of the flow. Most of the papers describe the fact that for classification it is necessary to collect the largest amount of information. Collecting a lot of information for example, analyzing all flows or all flow characteristics can lead to large delays in the network infrastructure. Such an approach will take significant computational resources of the system. In order for our system to work in close to real time mode and not affect the throughput, we will fix some of the characteristics from a limited number of packets. Thus, we will only work with a part of the total flow. In order to obtain a high level of real-time confidence, we will sacrifice some information, which may lead to a decrease in accuracy. The most difficult task is to maintain the relationship between accuracy, delay and throughput, which determines the choice of flow characteristics and the size of the observation window, as well as the choice of classification algorithm.

2.3 Definition of characteristics

Our complete feature set contains 220 different features based on the description [18]. Each characteristic has a different distribution within the data sets and is included in one of the collections. Collections define the computational cost of working with a given feature to find a subset of a given set of features within an upper bound on the computational overhead. We need to determine the necessary and sufficient information that will lead to the required classification accuracy. In fact, we limit the allocated resources by reducing the number of characteristics in the set of flow characteristics, for this we use the correlation-based filtering method [19].
3 Results

This section discusses results from Weka simulations for C4.5 Decision Tree and Random forest Method, support vector machine (SVM), and KNN. Estimates are made based on the received data set. Methods are compared according to the following parameters.

Sensitivity determines the proportion of true positive results, characterizes the classifier by how much positive results are true.

\[
\text{sensitivity} = \frac{TP}{(TP+FN)}
\]

Specificity determines the accuracy with which negative results can be correctly detected, solves the problem of detecting negative results.

\[
\text{specificity} = \frac{TN}{(TN+FP)}
\]

Accuracy shows the accuracy of the selected classifier and is defined as the ratio of the total number of TP/TN to the total amount of data.

\[
\text{accuracy} = \frac{TN+TP}{(TN+TP+FN+FP)}
\]

3.1 System performance

Comparison of the characteristics of classifiers is presented in Table 2. The comparison graph is shown in Figure 2.

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Accuracy</th>
<th>Sensitivity</th>
<th>Specificity</th>
</tr>
</thead>
<tbody>
<tr>
<td>C4.5 Decision Tree</td>
<td>0.98</td>
<td>0.879</td>
<td>0.896</td>
</tr>
<tr>
<td>Random forest Method</td>
<td>0.997</td>
<td>0.856</td>
<td>0.975</td>
</tr>
<tr>
<td>SVM</td>
<td>0.992</td>
<td>0.796</td>
<td>0.981</td>
</tr>
<tr>
<td>KNN</td>
<td>0.987</td>
<td>0.927</td>
<td>0.992</td>
</tr>
</tbody>
</table>
The load average indicator is related to the processor load and the average load on the platform [21]. At the time of the experiment, the processor load is monitored in a time interval of at least 15 minutes to obtain an average platform load. CPU load is the exact value of how much processor time is allocated for the calculation of a particular task. It should be taken into account that the average load value measures the “traffic” of the CPU, i.e., the number of memory accesses. Thus, a situation may occur when the processor is not loaded, but at the same time, the average load indicates a high CPU wait time. Table 3 shows the processor load in % and the average value of the load on the platform. To compare the test results, the processor load is normalized in the range from 0 to 100%. Average load values are normalized for the use of a multiprocessor multi-core system. This normalization implies that 100% of the CPU load is taken to be the use of all available cores in the system. An average value below 1 means that there is no process in the queue and all incoming requests are processed immediately. A value greater than 1 means that processes should wait for processing. A value higher than 1 in our case will not be acceptable, because speeds close to real time are needed.
Processing time is an additional comparison factor between classifiers. Most research focuses on classification accuracy and memory consumption. The studies did not take into account the processing time factor. When classifying network traffic, processing time is one of the main parameters. Our study estimates the time spent building models and the time spent classifying the entire dataset. Table 4 shows the values of the model building time C4.5 Decision Tree, Random forest Method, SVM, KNN. The extreme values will be C4.5 and Random forest. SVM, KNN values will be average. It takes 0.18 seconds to build a model to classify 1000 instances of Random forest, while the C4.5 model takes 0.08 seconds. It takes 0.23 seconds to build a model to classify 2000 instances of Random forest, while the C4.5 model takes 0.26 seconds. It takes 3.15 seconds to build a model to classify 8000 instances of Random forest, while the C4.5 model takes 10.1 seconds. It should be noted that the main number of instances in the classifier in the problem being solved does not exceed 2000 instances. Model C4.5 shows clear benefits on a limited number of instances. The overall processing time (spent on classifying the entire data set) for the C4.5 model for classifying 1000 instances was 1.7 seconds, for Random forest 1.23 seconds. As the number of instances increases, the elapsed time increases, for 8000 instances C4.5 will take 124 seconds, and Random forest 71 seconds.

Table 4. Table model building time and general processing time
4 Discussion

It can be argued that the discriminative learning technique on the example of C4.5 is a tool capable of providing the required level of reliability and performance under given conditions. It should be taken into account that the chosen methodology was adapted to the tasks of systems for modeling telecommunication networks and classifying traffic in them. Characteristics are not subjected to additional processing to increase their uniqueness. In addition, a smaller number of flow characteristics are taken than in the experiments of other...
It can be concluded that the performance of a highly specialized algorithm within the strict constraints on the conditions can give greater performance. In our experiment, it was shown that it is possible to achieve the required accuracy under the condition of less utilization of computing resources. For most applications, such as routing protocols, data transfer protocols, or illegitimate traffic, the required accuracy can be achieved within a binary classification. This method will be less expensive in terms of hardware resources compared to multi-class classification.

In addition, it can be noted that the elimination of temporal anomalies and the adaptability of the model to unconsidered network applications will be possible with the use of incremental learning tools. Incremental learning is possible with gradual updating of datasets based on newly obtained data from the model [22]. Adding classical classifiers based on ports to the considered model and/or marking traffic with additional tools [1] can significantly increase the efficiency of the solution. The C4.5 model is obsolete and is considered as a reorganization concept. When replacing it with more modern models, you can also improve the results.

5 Conclusion

The parameters Accuracy, Sensitivity, Specificity are defined. As part of the load simulation, a method with the least load on the computing power of the platform, C4.5, was identified. The parameters of the model building time and the general processing time for the case with the number of classification instances up to 2000 are determined. The points where C4 model gives advantages. Each method was evaluated in terms of classification accuracy and processing time. C4.5 achieved a high percentage of accuracy - 98% with a CPU load of 23.4% and a model building time with up to 2000 instances of 0.26 seconds.

References

6. J. Zhao et al., Information Fusion 72, 22-47 (2021)
7. O. Salman et al., Annals of Telecommunications 75(11), 673-710 (2020)
12. J. Toldinas et al., Electronics 10(15), 1854 (2021)
20. B. Gregg, Linux load averages: Solving the mystery, Brendan Gregg’s Blog 18 (2017)

https://doi.org/10.1051/e3sconf/202340203001