# Model and Algorithms for User Identification by Network Traffic

Vasily Gai[1], Irina Ephode[1], Roman Barinov[1], Igor Polyakov[1], Vladimir Golubenko[1] and Olga Andreeva[1]

[1]*Nizhny Novgorod State Technical University, st. Minina, 24, Nizhny Novgorod, 603155, Russia*

**Abstract**

This paper proposes a method of user identification by network traffic. We describe the information model created, as well as the implementation of each of the proposed problem solving stages. During the network traffic collection stage, a method of capturing network packets on the user's device using specialized software is used. The information obtained is further filtered by removing redundant data. During the object feature descriptor construction stage, we extract and describe the characteristics of network sessions from which the behavioral habits of users are derived. Classification of users according to the extracted characteristics of the network sessions is performed using machine learning techniques. When analyzing the test results, the most appropriate machine learning algorithms for solving the problem of user identification by network traffic were proposed, such as: logistic regression, decision trees, SVM with a linear hyperplane and the boosting method. The accuracy of the above methods was more than 95%. The results proved that it is possible to identify a particular user with a sufficiently high accuracy based on the characteristics of the data transmitted through the network, without examining the contents of the transmitted packets. Comparison of the developed model has shown that the proposed model of user identification by network traffic works as effectively as the existing analogues.

**Keywords**

Network traffic, machine learning algorithms, user identification, network traffic analytics, supervised learning

## 1. Introduction

Network traffic is the information transmitted through a computer network using specific rules (protocols) over a period of time. Virtually everyone today owns multiple devices (e.g. smartphones, tablets, laptops, workstations, etc.) which they actively use to exchange, receive and transmit information for various purposes. From this we can conclude that a particular user generates unique network traffic, which is determined by the user's behavioral habits and the characteristics of network sessions. Consequently, there is a fairly strong correlation between network traffic already collected and new data collected over a period of time. This may allow a fairly accurate identification of a particular user from network usage data. It is worth noting that when there is insufficient data to train the algorithm (and especially when using statistics from only one of the devices), not only will the approach fail to show its effectiveness, but it is very likely to produce false predictions. With a sufficient amount of data from a large number of users, the proposed approach will not only identify a specific person (among those who submitted their traffic), but also predict some attributes describing each specific user (e.g. gender, age, etc.) or group of users, which can significantly improve the algorithms of contextual advertising, news targeting, etc., solving a wide range of problems.

*GraphiCon 2021: 31st International Conference on Computer Graphics and Vision, September 27-30, 2021, Nizhny Novgorod, Russia*
EMAIL: iamuser@inbox.ru (V. Gai); zhestckova.natali@yandex.ru (I. Ephode); barinovr@list.ru (R. Barinov); polyakovigor92@gmail.com (I. Polyakov); fullmoonshrine@gmail.com (V. Golubenko); mrroman152@gmail.com (O. Andreeva)
ORCID: 0000-0002-3644-5234 (V. Gai); 0000-0002-0269-3205 (R. Barinov); 0000-0002-1492-9350 (I. Polyakov); 0000-0002-4683-3249 (V. Golubenko); 0000-0001-9581-3028 (O. Andreeva)

## 2. Overview of existing methods

Although a wide variety of algorithms have been created to solve the network traffic classification task, there are currently 2 main approaches to this problem [1] (Figure 1).

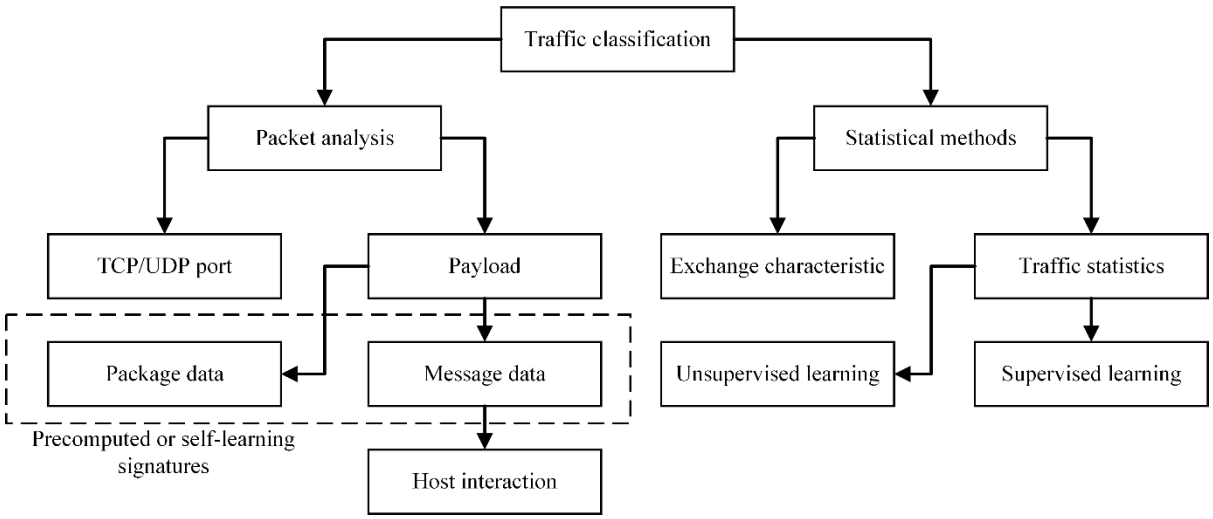


**Figure 1**: Main approaches to solving the task of network traffic classification

### 2.1. Data block-based classification

This method is usually divided into two approaches - the universal approach to network traffic classification and the Deep Package Inspection (DPI) method.

The universal approach to traffic classification is based on the data in the IP packet header - IP address, MAC address and protocol used [2]. It is worth noting the limitations of this method, as the information is taken only from the IP header.

Deep packet analysis (DPI) provides a more accurate solution to the classification problem. Such systems allow recognition of applications and protocols that cannot be identified at the network level (e.g. URLs, contents of messenger messages, Skype voice traffic, BitTorrent p2p packets, etc.). It follows that DPI analyses not only the headers, but also the full contents of packets at all levels of the ISO OSI model starting from the data link layer [3]. The primary mechanism for identifying applications in DPI is signature analysis. All applications have their unique characteristics recorded in a signature database. By comparing the sample from the database with the image of the traffic being analyzed, the application or protocol can be accurately identified.

### 2.2. Classification based on statistical methods

Two different approaches need to be distinguished in statistical methods:

- Network-layer behavioral and statistical algorithms;
- Transport-layer behavioral and statistical algorithms.

The main purpose of behavioral algorithms is to identify the applications generating network traffic. These algorithms are based on the fact that network traffic has statistical characteristics that are unique to certain classes of applications and allow us to separate traffic by application [4, 5].

By analyzing host interactions on a computer network, it is possible to identify the applications running on a computer [6]. The relationship between a class of traffic and its behavioral statistical properties has been described in specialized databases that include empirical models of connection characteristics for a number of specific TCP applications.

## 3. Network traffic classification information model

The proposed method for solving the problem is based on statistical methods. This approach takes a relatively small amount of machine time, but requires additional analysis and data processing. It is worth noting that in-depth analysis of network packets is performed on the packet analyzer side, so the proposed solution is essentially a combination of existing approaches for solving the network traffic classification problem [7]. The feature description will be generated based on the characteristics of network sessions, from which the behavioral habits of the users will be derived.

The stages of developing a user identification model based on the generated network traffic are described by the scheme shown in Figure 2.

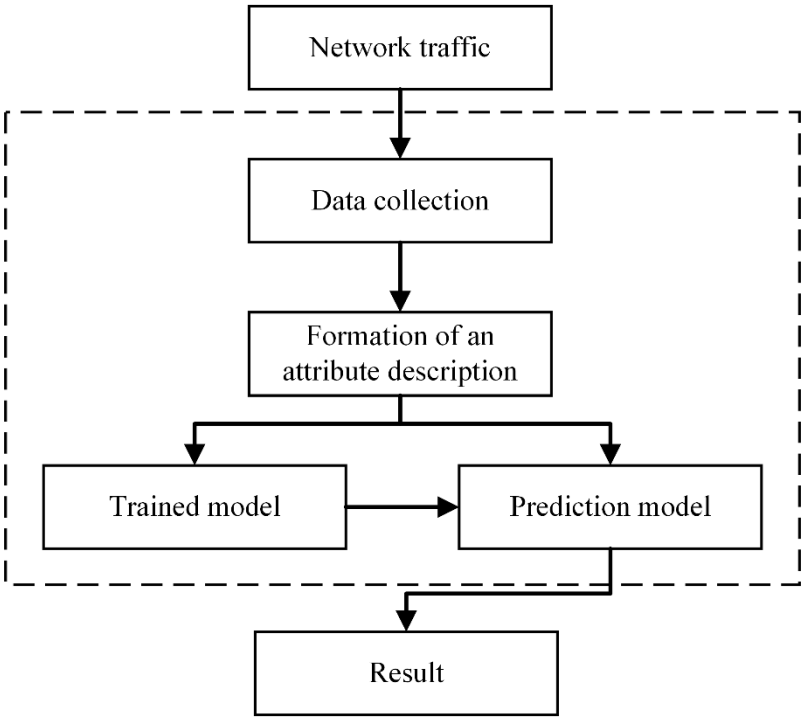


**Figure 2:** Network traffic classification model

The data collection phase is one of the most important steps in the process of solving the problem of user identification from network traffic using machine learning models and algorithms. It is at this stage that the feature space needs to be defined, and redundant data that could degrade the quality of model learning be extracted and removed from the sample.

### 3.1. Selecting a means of collecting network traffic

There exist several ways to collect a user's network traffic [8].

A network router can be used to collect the traffic. The main disadvantage of this way of collecting information is the dependence on hardware, as not all routers allow collecting statistics of network traffic usage (because not all devices of this class support logging and saving information). In addition, data collection requires additional knowledge, skills and documentation, as well as additional processing for the results of the obtained statistics of different levels and priority. Additionally, the information may not be sufficient, as the traffic may be collected solely at the network level. It is also worth noting that another disadvantage of this method is the collection of network-wide statistics, i.e. in order to obtain the required data, additional sifting of irrelevant information must be performed.

Network traffic can be collected by means of the firewall on the PC router. The vast majority of firewalls work exclusively at the network layer, hence why they collect statistics on port requests, which

means that there simply may not be enough information to solve the problem. The process is further complicated by the additional analysis of the data and saving the information in the correct format.

Operating system interfaces or physical network interfaces can also act as a means of collecting network information. Typically, these devices only collect statistics on data transmission over a particular channel, which in the context of this task is of no value.

The most affordable and optimal way is to capture network packets on the user's device using a sniffer. To start collecting statistics on network traffic usage, one must simply install the sniffer and run it. The resulting data can easily be exported to any supported format (including tabular or object description). The data itself will contain all the information needed for analysis, since the software already performs in-depth analysis of transmitted packets.

In this work to collection of statistics on the use of network traffic was performed using WireShark, which allows one to analyze packets transmitted via TCP/IP protocol stack. These packets that define the Internet traffic that can be analyzed. This means that the rest of the traffic must be filtered out.

## 3.2. Generating a feature descriptor

Each training sample object must have features describing the behavior of a particular user. By analyzing the object fields in the traffic describing the transmitted/received packet obtained from the analyzer, it was concluded that user behavior is characterized by the time and place of connection, as well as by characteristics describing the network session, namely:

- User name;
- The country from which the connection was initiated;
- The day of the week on which the connection was opened;
- The hour and minute at which the connection was opened;
- The source IP address and its host name;
- The destination IP address and its host name;
- TCP source port;
- TCP destination port.

The username will be used as the field that defines the class to train the machine learning algorithms with the teacher.

The country from which the connection was started can be identified in two ways, each having its own drawbacks.

The first way is to analyze the connection time. By using the time zone and the operating system time, it cannot be asserted that the user is exactly in the time zone of their permanent residence, as the operating system may not be guided by the time given by the network.

The second way is to calculate the country from the IP address of the host. When used to calculate the user's location, VPNs (Virtual Private Networks) and other similar solutions that may be set up in another country are a major obstacle.

Having analyzed these cases, it was decided to use time zones set by the operating system, as this would give more complete information regarding the user's permanent location.

The day of the week on which the connection was opened can be obtained from the "timestamp" field. Its value is an unsigned int number, specified in UNIX-timestamp, equal to the number of seconds elapsed since midnight on January 1, 1970, Greenwich Mean Time (i.e. zero-time zone, a time zone reference). When retrieved from the database, it is displayed according to the time zone set in the operating system, the global database settings, or a specific session. The number of seconds is always stored according to UTC (Universal Coordinated Time, Greenwich Meridian Solar Time) rather than the local time zone. The time stamp must be converted to a day of the week, a number in the range of 0 to 6.

The hour and minute at which the connection was opened will be obtained and completed in the same way as in the previous point. This kind of data characterizes the behavior of a particular user.

The source IP address and its host name are not only important in terms of address location on the network, but also the name of the machine from which the connection was made, as this data can change but still characterize the connection.

## 3.3. Choosing machine learning algorithms and metrics for assessing model adequacy

Based on the analysis of the most popular machine learning algorithms, it was decided to consider a list of various algorithms to select the most accurate model capable of identifying the user from the generated network traffic, namely:

- Support Vector Machine;
- Naive Bayesian classifier;
- K-nearest neighbours;
- Random forest algorithm;
- Logistic regression;
- Boosting.

The proposed machine learning algorithms cover the most popular methods for solving similar problems. This will allow the results of the algorithms to be compared with each other, which will give a better understanding in selecting an effective model.

During the data analysis stage it was discovered that the number of objects for each class is different - this means that some metrics do not work correctly. For this reason, we decided to estimate the models' adequacy using several of the most popular metrics at once:

- Accuracy;
- Completeness;
- F-measure.

# 4. Computational experiment

In order to conduct a computational experiment, generated network traffic was collected from several users using the WireShark traffic analyzer. The obtained data was saved in JSON format. After analyzing the obtained data, it was decided to transform the data, with each object having a certain structure and feature space, as well as belonging to a class with the subsequent purpose of using the data for training machine learning algorithms with a teacher. These objects were compiled into a CSV table for the purpose of training machine learning algorithms to solve the problem of user identification by network traffic, which boils down to solving the classification problem.

The main goal of the computational experiment is to find the most accurate algorithm for the user identification task based on network traffic.

## 4.1. Data preparation

There are several necessary steps to perform when preparing data for machine learning algorithms: learn to distinguish the packets transmitted through the TCP/IP protocol stack. All packets transmitted through the TCP/IP protocol stack have "tcp" and "ip" fields in the "layers" field. Therefore, all other objects (transmitted via other protocol stacks) can be omitted; select from the set of fields those which can influence the result of the classifier.

This data is extracted from transmitted packets using Deep Package Inspection algorithms and is fundamental to the network connection defined by the TCP/IP protocol stack, so it is these data that are taken as a basis.

After transformations, it is convenient to summarize all data into a table (Table 1), where each new row is an object and each column is a particular attribute.

This kind of data is already suitable for analysis with machine learning algorithms. The total amount of data is 28036 objects (after removing duplicates). Of these, 70% is the training sample, 30% is the validation sample. Total number of classes is 4.

**Table 1**
Example of feature description for 4 users

| Name | Country | Hours | Minutes | Weekday | IP.src | IP.dst |
|---|---|---|---|---|---|---|
| User_1 | Russia | 23 | 12 | 3 | 77.88.21.119 | 172.20.10.8 |
| User_2 | Russia | 11 | 39 | 0 | 93.158.162.221 | 192.168.1.60 |
| User_3 | Russia | 20 | 01 | 6 | 31.216.145.26 | 192.168.1.96 |
| User_4 | Russia | 14 | 42 | 0 | 178.248.233.33 | 192.168.1.195 |

**Table 1 (cont.)**

| IP.src.host | IP.dst.host | TCP.src.port | TCP.dst.port |
|---|---|---|---|
| mc.yandex.ru | 172.20.10.8 | 443 | 53443 |
| S82vla.storage.yandex.net | 192.168.1.60 | 443 | 39312 |
| Gfs270n073.userstorage.mega.co.nz | Iefode | 443 | 52639 |
| qna.habr.com | 68edebd2-8177-4957-93bd58c9107a9b00.local | 443 | 56548 |

## 4.2. Experimental setup and evaluation of results

For the experiment, it is necessary to create different models for solving the classification problem using selected algorithms, and to evaluate their adequacy and accuracy using metrics. Before proceeding, it is worth evaluating the applicability of machine learning algorithms to this class of problems. To do this, it is advisable to build a correlation matrix (Figure 3) based on the collected data. When computing pairwise correlation of features, the Pearson method was used.

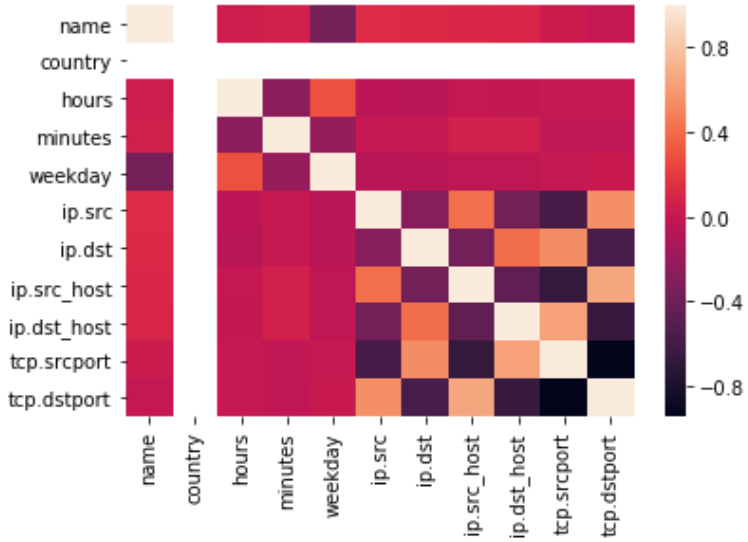


**Figure 3**: Correlation matrix of extracted features

As can be seen in Figure 3, many of the features are dependent on each other, meaning machine learning algorithms can be applied to solve the task.

It is also necessary to assess the extent to which each of the attributes affects the outcome of the algorithm, i.e. how important each particular attribute is in predicting the outcome.

Tables 2-8 show the results of experiments on classification of user traffic using various machine learning algorithms.

**Table 2**
Results obtained using SVM – non-linear hyperplane

| Class | Precision | Recall | F1-score |
|---|---|---|---|
| User_1 | 0.81 | 0.99 | 0.89 |
| User_2 | 0.98 | 0.63 | 0.77 |
| User_3 | 1.00 | 0.29 | 0.46 |
| User_4 | 0.94 | 0.62 | 0.75 |
| Micro avg | 0.84 | 0.84 | 0.84 |
| Macro avg | 0.93 | 0.63 | 0.71 |
| Accuracy | | 0.8434 | |

**Table 3**
Results obtained using SVM – linear hyperplane

| Class | Precision | Recall | F1-score |
|---|---|---|---|
| User_1 | 0.81 | 0.99 | 0.89 |
| User_2 | 0.98 | 0.63 | 0.77 |
| User_3 | 1.00 | 0.29 | 0.46 |
| User_4 | 0.94 | 0.62 | 0.75 |
| Micro avg | 0.84 | 0.84 | 0.84 |
| Macro avg | 0.93 | 0.63 | 0.71 |
| Accuracy | | 0.9998 | |

**Table 4**
Results obtained using logistic regression algorithm

| Class | Precision | Recall | F1-score |
|---|---|---|---|
| User_1 | 1.00 | 1.00 | 1.00 |
| User_2 | 1.00 | 1.00 | 1.00 |
| User_3 | 1.00 | 1.00 | 1.00 |
| User_4 | 1.00 | 1.00 | 1.00 |
| Micro avg | 1.00 | 1.00 | 1.00 |
| Macro avg | 1.00 | 1.00 | 1.00 |
| Accuracy | | 1.0000 | |

**Table 5**
Results obtained using naive Bayes classificator

| Class | Precision | Recall | F1-score |
|---|---|---|---|
| User_1 | 0.81 | 0.99 | 0.89 |
| User_2 | 0.98 | 0.63 | 0.77 |
| User_3 | 1.00 | 0.29 | 0.46 |
| User_4 | 0.94 | 0.62 | 0.75 |
| Micro avg | 0.84 | 0.84 | 0.84 |
| Macro avg | 0.93 | 0.63 | 0.71 |
| Accuracy | | 0.7953 | |

**Table 6**
Results obtained using k-nearest neighbors

| Class | Precision | Recall | F1-score |
|---|---|---|---|
| User_1 | 0.81 | 0.99 | 0.89 |
| User_2 | 0.98 | 0.63 | 0.77 |
| User_3 | 1.00 | 0.29 | 0.46 |
| User_4 | 0.94 | 0.62 | 0.75 |
| Micro avg | 0.84 | 0.84 | 0.84 |
| Macro avg | 0.93 | 0.63 | 0.71 |
| Accuracy | | 0.7834 | |

**Table 7**
Results obtained using random decision forests

| Class | Precision | Recall | F1-score |
|---|---|---|---|
| User_1 | 0.65 | 1.00 | 0.79 |
| User_2 | 1.00 | 0.15 | 0.25 |
| User_3 | 0.00 | 0.00 | 0.00 |
| User_4 | 1.00 | 0.15 | 0.26 |
| Micro avg | 0.67 | 0.67 | 0.67 |
| Macro avg | 0.66 | 0.32 | 0.33 |
| Accuracy | | 0.6710 | |

**Table 8**
Results obtained using Yandex.CatBoost algorithm.

| Class | Precision | Recall | F1-score |
|---|---|---|---|
| User_1 | 0.98 | 0.99 | 0.99 |
| User_2 | 0.99 | 0.98 | 0.98 |
| User_3 | 1.00 | 1.00 | 1.00 |
| User_4 | 0.99 | 0.98 | 0.98 |
| Micro avg | 0.99 | 0.99 | 0.99 |
| Macro avg | 0.99 | 0.98 | 0.99 |
| Accuracy | | 1.0000 | |

All of the algorithms were assessed for their performance during the experiment (Table 9).

**Table 9**
Performance assessment of the algorithms

| Algorithm | Accuracy | Precision | Recall | F1-score |
|---|---|---|---|---|
| SVM – non-linear hyperplane | 0.84 | 0.87 | 0.84 | 0.83 |
| SVM – linear hyperplane | 0.99 | 1.00 | 1.00 | 1.00 |
| Logistic regression | 1.00 | 1.00 | 1.00 | 1.00 |
| Naive Bayes classificator | 0.79 | 0.94 | 0.79 | 0.85 |
| k-nearest neighbors | 0.78 | 0.86 | 0.78 | 0.80 |
| Decision tree | 1.00 | 1.00 | 1.00 | 1.00 |
| Random decision forests | 0.67 | 0.76 | 0.67 | 0.58 |
| Yandex CatBoost | 1.00 | 0.99 | 0.98 | 0.99 |

From the assessment we can see that in the context of the problem of user identification by network traffic, the k-nearest neighbour algorithms, naive Bayesian classifier, SVM with non-linear hyperplane and random decision forests algorithm can be considered as the most unfitting for solving the problem.

In the context of application to the network traffic classification problem, logistic regression methods, decision trees, reference vector method with a linear hyperplane, and boosting can be considered as most accurate.

Of note is the Yandex CatBoost algorithm, which has also proven itself for solving this problem. This algorithm is also internally composed of decision trees. From this, it can be concluded that algorithms that include decision trees are suitable for solving this problem.

From all of the above it can be concluded that the decision trees method proved to be the most effective. It showed excellent results for all metrics, and also proved to be a fast learning algorithm.

## 4.3. Comparing findings with other studies

In a study by Austrian scientists [9], which aimed to detect external influences on the system using various machine learning algorithms to classify network traffic, the results shown in Table 10 were obtained.

**Table 10**

Results of external influence detection algorithms on the [9] researchers' system (Accuracy metric)

| Tool | Complexity (Signature) | Training size | Success rate |
|---|---|---|---|
| DirBuster | Low | 10-50 | High (99%) |
| Burp Suite | None (plain TCP) | 5000+ | Low (40%) |
| Nessus | Complex | 5000+ | Medium (85%) |
| Sqlmap | Low | 10-50 | High (99%) |
| Nikto | Low | 10-50 | High (99%) |

The study [10] defines an application layer protocol using deep learning techniques, which boils down to the task of classifying network traffic. This paper presents the results of the algorithms shown in Table 11.

**Table 11**

Results of the application layer protocol definition model

| Protocol | Precision | Recall |
|---|---|---|
| SSL | 0.9513 | 0.9763 |
| HTTP_Proxy | 0.9174 | 0.9090 |
| MySQL | 0.9989 | 0.9993 |
| SMB | 1.0000 | 1.0000 |
| HTTP_Connect | 0.9967 | 0.9930 |
| Whols-DAS | 0.9943 | 0.9777 |
| Redis | 0.9985 | 0.9974 |
| SSH | 0.9996 | 1.0000 |
| Apple | 0.9640 | 0.9728 |
| Kerberos | 0.9996 | 0.9996 |
| DCE_RPC | 1.0000 | 1.0000 |
| NetBIOS | 1.0000 | 1.0000 |
| FTP_CONTROL | 0.9970 | 0.9973 |
| DNS | 0.9989 | 0.9985 |
| Skype | 0.9779 | 0.9722 |
| LDAP | 0.9996 | 0.9992 |
| AppleiCloud | 0.9679 | 0.9689 |
| AppleiTunes | 0.9520 | 0.9617 |
| MSN | 0.9453 | 0.9230 |

**Table 11 (cont.)**

| Protocol | Precision | Recall |
|---|---|---|
| Gmail | 0.9953 | 0.9973 |
| BitTorrent | 0.9992 | 0.9992 |
| TDS | 1.0000 | 1.0000 |
| IMAPS | 0.9814 | 0.9654 |
| SMTP | 0.9949 | 0.9883 |
| RSYNC | 0.9987 | 0.9993 |
| Avg | 0.98 | 0.96 |

Research [11] studies identification of network traffic generated by malware. The central problem in this paper also boils down to solving the problem of classifying network traffic using methods belonging to a generalized class of artificial intelligence methods. In this study, various algorithms are evaluated using different metrics to assess the adequacy of the model created using machine learning algorithms. The results presented in [11] are shown in Table 12.

**Table 12**
Results of the malware-generated network traffic identification model

| Algorithm | Precision | Recall | F-Measure | ROC Area | Class |
|---|---|---|---|---|---|
| J48 | 0.998 | 0.999 | 0.999 | 0.999 | nonTOR |
| | 0.997 | 0.993 | 0.995 | 0.999 | TOR |
| Weighted Avg | 0.998 | 0.998 | 0.998 | 0.999 | |
| J48Consolidated | 0.999 | 0.998 | 0.999 | 1.000 | nonTOR |
| | 0.991 | 0.998 | 0.994 | 1.000 | TOR |
| Weighted Avg | 0.998 | 0.998 | 0.998 | 1.000 | |
| BayesNet | 0.996 | 0.982 | 0.989 | 0.999 | nonTOR |
| | 0.918 | 0.980 | 0.948 | 0.999 | TOR |
| Weighted Avg | 0.982 | 0.982 | 0.982 | 0.999 | |
| jRip | 1.000 | 1.000 | 1.000 | 1.000 | nonTOR |
| | 0.998 | 1.000 | 0.999 | 1.000 | TOR |
| Weighted Avg | 1.000 | 1.000 | 1.000 | 1.000 | |
| OneR | 1.000 | 0.997 | 0.999 | 0.999 | nonTOR |
| | 0.987 | 1.000 | 0.994 | 0.999 | TOR |
| Weighted Avg | 0.998 | 0.998 | 0.998 | 0.999 | |
| RepTREE | 0.997 | 0.987 | 0.992 | 0.999 | nonTOR |
| | 0.923 | 0.983 | 0.951 | 0.998 | TOR |
| Weighted Avg | 0.984 | 0.983 | 0.984 | 0.998 | |

The results obtained in [9-11] and this study are comparable with each other on various metrics, which suggests the correctness of this work.

## 5. Conclusion

We have performed the task of data collection, consisting of determining the appropriate method of data collection and partitioning, as well as conducting in-depth analysis of the obtained data. Based on the analysis, the data preparation for processing with machine learning algorithms, and a feature space was defined. In the analysis of existing approaches to solving the problem of user identification by network traffic, various ways of solving this problem were considered and the most popular algorithms of machine learning were selected for the experiment. In addition, we chose appropriate metrics to assess the adequacy of the model.

Based on the results obtained in the computational experiment, the most accurate models were identified. The average accuracy of the algorithms was more than 95% for all evaluated metrics.

It was proved that based on the characteristics of the data transmitted through the network, it is possible to identify a particular user with sufficiently high accuracy, without examining the contents of the transmitted packets.

## 6. References


[1] Jamuna. A, Vinodh Ewards S.E: "Efficient Flow based Network Traffic Classification using Machine Learning," International Journal of Engineering Research and Applications (IJERA), ISSN: 2248-9622, Vol. 3, Issue 2, 2013, pp.1324-1328.

[2] T. Bujlow, T. Riaz and J. M. Pedersen, "A method for classification of network traffic based on C5.0 Machine Learning Algorithm," 2012 International Conference on Computing, Networking and Communications (ICNC), 2012, pp. 237-241, doi:10.1109/ICCNC.2012.6167418.

[3] Byungchul Park, Young J. Won, Mi-Jung Choi, Myung-Sup Kim, and James W. Hong: "Empirical Analysis of Application-level Traffic Classification using Supervised Machine Learning," IT RD program of MKE/IITA [2008-F-016-01, CASFI] and the EECE division at POSTECH under the BK21 program of MEST, Korea. doi:10.1007/978-3-540-88623-5_55.

[4] V. Paxson, "Empirically derived analytic models of wide-area TCP connections," in IEEE/ACM Transactions on Networking, vol. 2, no. 4, pp. 316-336, 1994, doi:10.1109/90.330413.

[5] V. Paxson and S. Floyd. "Wide-Area Traffic: The Failure of Poisson Modeling," Networking, IEEE/ACM Transactions on. 3, 1995, 226-244, doi:10.1109/90.392383.

[6] T. Karagiannis, K. Papagiannaki, and M. Faloutsos. 2005. BLINC: multilevel traffic classification in the dark. SIGCOMM Comput. Commun. Rev. 35, 4, 2005, 229–240. doi:10.1145/1090191.1080119.

[7] M. Wang, Y. Cui, X. Wang, S. Xiao and J. Jiang, "Machine Learning for Networking: Workflow, Advances and Opportunities," in IEEE Network, vol. 32, no. 2, 2018, pp. 92-99, doi: 10.1109/MNET.2017.1700200.

[8] Alisha Cecil: "A Summary of Network Traffic Monitoring and Analysis Techniques", URL: https://www.cse.wustl.edu/~jain/cse567- 06/ftp/net_monitoring/index.html.

[9] P. Fruhmrt, S. Schrittwieser, E.R. Weippl, "Using machine learning techniques for traffic classification and preliminary surveying of an attacker's profile", St. Polten University of Applied Sciences.

[10] Z. Wang, "The application of deep learning on traffic identification," 2015. URL: http://www.blackhat.com.

[11] A. Cuzzocrea, F. Martinelli, F. Mercaldo and G. Vercelli, "Tor traffic analysis and detection via machine learning techniques," 2017 IEEE International Conference on Big Data (Big Data), 2017, pp. 4474-4480, doi:10.1109/BigData.2017.8258487.