Classification of encrypted Applications of Traffic Mobile Devices using the Data Mining

Oleg I. Sheluhin[1], Viacheslav V. Barkov[2], Mikhail V. Polkovnikov[3]
Department of Information Security
Moscow Technical University of Communications and Informatics
Moscow, Russia
[1]sheluhin@mail.ru, [2]viacheslav.barkov@gmail.com, [3]mnxamoto@mail.ru

Abstract. The problem of determining by a cellular operator what applications a particular network user has used is needed to compile statistics of the most frequently used applications. Such a definition of application statistics helps to not only monitor network status, detect failures, but also, if necessary, restrict access to network resources that, from the point of view of information security, can harm the user. The introduction of methods of data mining and machine learning allows to perform automatic classification, analysis and filtering of malicious and unwanted mobile network traffic applications. Malicious mobile applications can be a threat to the integrity or availability of data, and unwanted ones are a threat to confidentiality. The paper considers classification of network encrypted traffic by application types: email of Mail.ru, Sberbank, Skype, Pikabu, Instagram, Hearthstone and other methods of machine learning using algorithms Naive Bayes, C4.5, SVM, AdaBoost and Random Forest. For the analysis, more than two million network packets were collected from four applications that transmitted encrypted traffic, after which training and test samples were generated. To assess the quality of the qualifier, such criteria as Accuracy, Precision, Recall, F-Measure and Area Under Curve were used.

The use of the InfoGain algorithm showed that to ensure the high quality of classification of traffic of applications that use encryption, it is enough to limit thirteen attributes. Classifier Random Forest is the slowest, but has the best indicators of assessing the quality of classification. The size of the learning sample of the Random Fores algorithm to achieve a sufficiently high quality of classification of mobile applications cannot exceed 300 threads. To ensure high quality thread classification, it is enough to analyze from 16 to 58 packets in a stream depending on the application. Further increase in the number of packets in the stream does not lead to a noticeable improvement in the quality of classification.

Keywords: **classification, machine learning, algorithms, network traffic, application, packet, flow, protocol, network, mobile applications efficiency.**

I. INTRODUCTION

The problem of determining by the cellular operator which applications the network user used is needed to compile statistics of the most frequently used applications. Such application statistics help not only to monitor the network status, detect failures, but also, if necessary, to restrict access to network resources that, from the point of view of information security, can harm the user.

The introduction of machine learning methods allows automatic classification, analysis and filtering of malicious and unwanted mobile applications of network traffic [1–3].

Malicious mobile applications can be a threat to the integrity or availability of data, and unwanted ones are a threat to confidentiality. Classification of traffic of mobile applications that use encryption-using encryption does not imply its decryption. The data inside the packets remains confidential and is only accessible to the user and the remote node.

Mobile applications that use traffic encryption can be divided into three groups. The first group includes applications that use the SSL / TLS transport layer encryption protocol [4] in conjunction with the HTTPS application layer protocol. Examples of such applications are Google, Facebook, Sberbank, etc. The second group includes applications that use the P2P protocol [6] with encryption (BitTorrent, MuTorrent, Vuze, etc.). The third group includes applications that use, in addition to transport-level encryption protocols, their own encryption protocols. Examples of such applications are Skype, WhatsApp, Telegram, etc.

In a situation of complex definition of the type of traffic encryption, it is advisable to use machine learning methods to classify the traffic of mobile applications using encryption.

II. TECHNOLOGY FOR COLLECTING TRAFFIC OF MOBILE APPLICATIONS

To create a database of mobile application traffic, the "Traffic Analysis System" software package was developed, which includes a database server, an application server, a Web application and client software for mobile devices running the Android operating system (mobile client).

The process of traffic collection using the "Traffic Analysis System" software package, as well as the interaction of the components of the software complex with each other and with external mobile applications is shown in Figure 1.

A mobile client of the "Traffic Analysis System" software package is installed on a smartphone or tablet running the Android operating system. This client intercepts network traffic packets of the specified applications that are also installed on this device.

Intercepted packets of network traffic are sent to the application server of the “Traffic Analysis System” software package installed on a server computer controlled by the Windows Server operating system 2016.

The application server of the “Traffic Analysis System” software package group’s network traffic packets into flows and, using the database server, saves data to the database.

Data exchange between the components of the “Traffic Analysis System” software is carried out via the global Internet using the HTTP protocol in JSON format. The application server includes a Web service that provides REST API clients, with which you can access the collection functions of network traffic packets, manage datasets, create and train classifiers, classify and other functions.

With the use of the software complex, traffic of mobile applications of three categories was collected: “With traffic encryption”, “Without traffic encryption”, “With partial encryption of traffic”.

During the collection of traffic of mobile applications using encryption, the network traffic flows of 6 applications were collected: Instagram, Mail Mail.ru, Pikabu, Sberbank-Online, Hearthstone, Skype. Table 1 shows the numerical characteristics of the collected network packets and flows for training and test samples.

To conduct the experiment and generate the initial data on the mobile device, specialized software “Traffic Analyzer” was installed under the management of the Android operating system [7] version 4.4. Figure 2 shows the process of connecting a mobile client to the application server of the “Traffic Analysis System” software package.

Fig. 1. Scheme of collecting mobile traffic

Table 1. Characteristics of the application dataset by application type when analyzing network packets and flows

Application	Training sample		Test sample	
	packets	flows	packets	flows
email Mail.ru	162517	3356	79612	1644
Sberbank	156648	3303	80482	1697
Skype	146315	3329	73443	1671
Pikabu	167182	3325	84220	1675
Instagram	1286600	3357	629695	1643
Hearthstone	151298	3330	75876	1670
Total	2070562	20000	1023328	10000

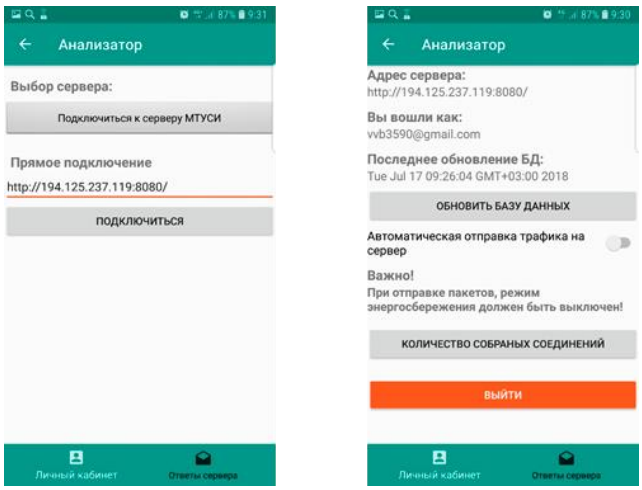


Fig. 2. Graphical user interface of the mobile client of the “Traffic Analysis System” software package: connection to the server and information about the server

Figure 3 shows the graphical user interface of the mobile client of the “Traffic Analysis System” software package during the configuration process to intercept the traffic of the specified applications and sends it to the server in the process of interception of traffic, in the process of viewing the number of collected streams.

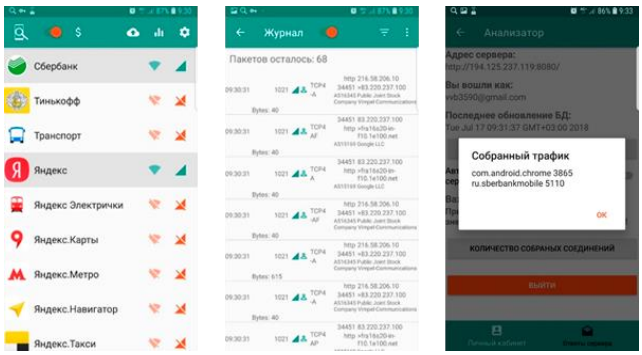


Fig. 3. Graphical user interface of the mobile client of the “Traffic Analysis System” software package

III. RESULTS OF THE CLASSIFICATION OF MOBILE APPLICATION TRAFFIC USING ENCRYPTION

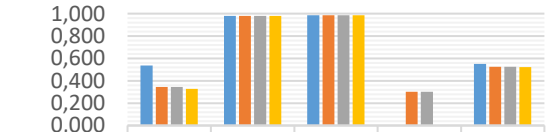
Using the algorithm for selecting the attributes InfoGain [http://www.cs.waikato.ac.nz/ml/weka] of the 23 original attributes, 13 were allocated:

- average size of the data portion from the server side (AverageSizeDataOnTransportLayerFromServer);
- average packet size on the server side (AverageSizeOnTransportLayerFromServer);
- the server's efficiency is the amount of applied application load transferred divided by the total number of

transferred application and transport layer loads (EfficiencyOfServer);

- customer address (FirstIP);
- size of network layer payload on the server side (NetworkLayerPayloadSizeFromServer);
- payload ratio is how many times the client transmitted more bytes of information than the server (RatioOfData);
- the server address (SecondIP);
- standard deviation of the size of the data portion from the client side (StandardDeviationOfDataOnTransportLayerFromClient);
- standard deviation of the data portion size from the server side (StandardDeviationOfDataOnTransportLayerFromServer);
- standard deviation of the packet size from the client side (StandardDeviationOfPacketSizeFromClient);
- standard deviation of the packet size from the server side (StandardDeviationOfPacketSizeFromServer);
- size of the payload of the transport layer from the client side (TransportLayerPayloadSizeFromClient);
- transport server payload size from the server side (TransportLayerPayloadSizeFromServer);

To assess the effectiveness of classification algorithms, the following information search metrics [1] were used: Precision, Recall, F-Measure, ROC curves (Receiver Operating Characteristic Curve), AUC (Area Under Curve) is the area under the ROC curve. Because of the processing of the experimental data, quantitative results were obtained, represented in the form of averaged histograms in Figure 4.



	Naive Bayes	C4.5	Random Forest	AdaBoost	SVM
Precision	0,535	0,980	0,985	0,000	0,552
Recall	0,344	0,980	0,985	0,302	0,524
Accuracy	0,344	0,980	0,985	0,303	0,524
F-мера	0,327	0,980	0,985	0,000	0,521

Fig. 4. Scheme of collecting mobile traffic

An analysis of the results shows that the algorithms C4.5 and Random Forest show the best results of classification. Figure 5 shows the time intervals in milliseconds that the classifiers required for training and testing required. As you can see, the fastest ones at the training stage were Naive Bayes, C4.5 and AdaBoost, and at the testing stage: C4.5, Random Forest, AdaBoost and SVM. The fastest classifiers in both phases are C4.5 and AdaBoost. However, although the AdaBoost classifier is the "fastest", however, it has the worst results of assessing the quality of classification. Classifier Random Forest is the most "slow", but has the best classification rating.

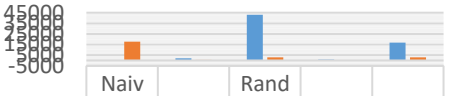
Table 2 shows the AUC values for the Random Forest algorithm, which show the high reliability of the classification of the applications considered.

The conducted researches allow to draw a conclusion that to ensure high quality of the flow classification it is enough to analyze from 16 to 58 packets in the stream depending on the application. The further increase in the number of packets in the stream does not lead to a marked improvement in the quality of the classification.

IV. CONCLUSIONS

Based on the use of the InfoGain algorithm, it is shown that in order to provide a high-quality classification of the considered applications using encryption for data transmission, it is sufficient to limit thirteen attributes. Classifier Random Forest is the slowest, but has the best indicators of assessing the quality of classification.

The size of the training sample algorithm for Random Forest sufficiently high classification quality (accuracy 90%) may not exceed 300 threads. To ensure high-quality classification of streams, it is enough to analyze from 16 to 58 packets of a flow depending on the application. Further increase in the number of packets in the flow more than not leads to a noticeable improvement in the quality of the classification.



	Naive Bayes	C4.5	Random Forest	AdaBoost	SVM
Время обучения	218	1953	42923	968	16813
Время тестирования	17673	563	2782	329	2969

Fig. 5. Time ranges for training and testing classifiers

TABLE I. AUC VALUES FOR RANDOM FOREST ALGORITHM

Class	Instagram	email Mail.ru	Skype	Sberbank	Hearthstone	Pikabu
ROC-AUC	0,9904	0,9885	0,9809	0,9920	0,9888	0,9797

V. REFERENCES

[1] Шелухин О.И., Ерохин С.Д., Ванюшина А.В. Классификация IP-трафика методами машинного обучения. М.: Горячая линия – телеком, 2018. [Sheluhin O.I, Erokhin S.D., Vanyushina A.V. IP traffic classification by machine learning methods. Moscow: Hotline — Telecom, 2018].

[2] Костин Д.В., Шелухин О.И. Сравнительный анализ алгоритмов машинного обучения для проведения классификации сетевого зашифрованного трафика // T-Comm: Телекоммуникации и транспорт. 2016. № 9. С. 46–52. [Kostin D.V., Sheluhin O.I. Comparison of nachine kearning algorithms for encrypted ckassification// T-Comm. 2016. Vol. 10, N 9. P. 43–52].

[3] Шелухин О.И., Смычек М.А., Симонян А.Г. Фильтрация нежелательных приложений трафика подвижной радиосвязи для обнаружения угроз информационной безопасности //

Радиотехнические и телекоммуникационные системы. 2018. № 1. С. 87–98. [Sheluhin O.I., Smychek M.A., Simonyan A.G. Filtering unwanbted mobile ratio traffic applications to detect imnfornatiobn security threats // Radio and Telecommunication Systems. 2018. N 1. P. 87–98].

[4] Шелухин О.И., Ванюшина А.В., Габисова М.Е. Фильтрация нежелательных приложений интернет-трафика с использованием алгоритма классификации Random Forest // Вопросы кибербезопасности. 2018. №2. С. 44–51. [Sheluhin O. ,Vanyushina A., Gabisova M. The filtering of unwanted applications in internet traffic using random forest classification algorithm // Cybersecurity issues. 2018. N 2. P. 44–51]..

[5] Rescorla E. SSL and TLS: Designing and Building Secure Systems. Reading: Addison-Wesley Professional, 2000. T. 1.

[6] Callegati F., Cerroni W., Ramilli M. Man-in-the-Middle Attack to the HTTPS Protocol // IEEE Security Privacy. 2009. T. 7, вып. 1. P. 78–81. DOI:10.1109/MSP.2009.12.

[7] Pouwelse J.A., Garbacki P., Epema D.H.J. et al. The BitTorrent P2P File-sharing system: Measurements and analysis// IPTPS 2005. LNCS. Vol. 3640. Heidelberg: Springer, 2005.

[8] Коматинэни С., Маклин Д., Хэшими С. GoogleAndroid: программирование для мобильных устройств ProAndroid 2. СПб.: Питер, 2011. [Komatinani S., McLean D., Hashimi S. Google Android: programming for mobile devices Pro Android 2. St Petersburg: Piter, 2011].