ISSN 0146-4116, Automatic Control and Computer Sciences, 2021, Vol. 55, No. 8, pp. 1121–1128. © Allerton Press, Inc., 2021.
Russian Text © The Author(s), 2020, published in Problemy Informatsionnoi Bezopasnosti, Komp'yuternye Sistemy.

An Encrypted File Detection Algorithm

A. V. Kozachok[a], *, V. I. Kozachok[a], and A. A. Spirin[a]

[a] *Academy of the Federal Guard Service of the Russian Federation, Oryol, 302034 Russia*

**e-mail: alex.totrin@gmail.com*

Received May 18, 2021; revised May 18, 2021; accepted June 20, 2021

Abstract—Despite the availability of data leak detection and prevention tools, there are currently a growing number of confidential data leaks through the fault of insiders. One of the possible data leak channels is encrypted or compressed data transfer, because the existing data leak detection tools use content data analysis methods. This article presents an algorithm of detecting encrypted and compressed data that is based on the statistical model of pseudo-random sequences and allows detecting encrypted and compressed data to an accuracy of 0.99.

Keywords: statistical data analysis, classification of encrypted and compressed data, machine learning, binary data analysis, pseudo-random sequences

DOI: 10.3103/S0146411621080162

INTRODUCTION

According to the InfoWatch expert analytics center, the first three quarters of 2020 saw the theft of 9.93 billion personal data (PD) and payment data records. Compared with the similar period in 2019, the amount of leaks and compromised records went down worldwide by 7.4 and 1.4%, respectively. Compared with the 69.5 million PD and payment data leaks in the first three quarters of 2019, their amount in the similar period in 2020 went down by 29.2%. The reduction in the number of leaks registered (discovered) around the world stems mainly from the influence of COVID-19 on private businesses and state-run enterprises: many companies could have reduced the control over information assets as a result of the hurried restructuring of their workflow and redeploying a large share of employees to remote work, and a large part of incidents was left unregistered. For the statistics on the data leaks registered by source, see Fig. 1.

At the same time, the number of leaks detected in Russia continued to grow even despite the pandemic, which can be juxtaposed with an intermittent growth of the number of applications for buying or handling products for leak control and monitoring employee activities. For the distribution of insiders by various categories of corporate personnel, see Fig. 2.

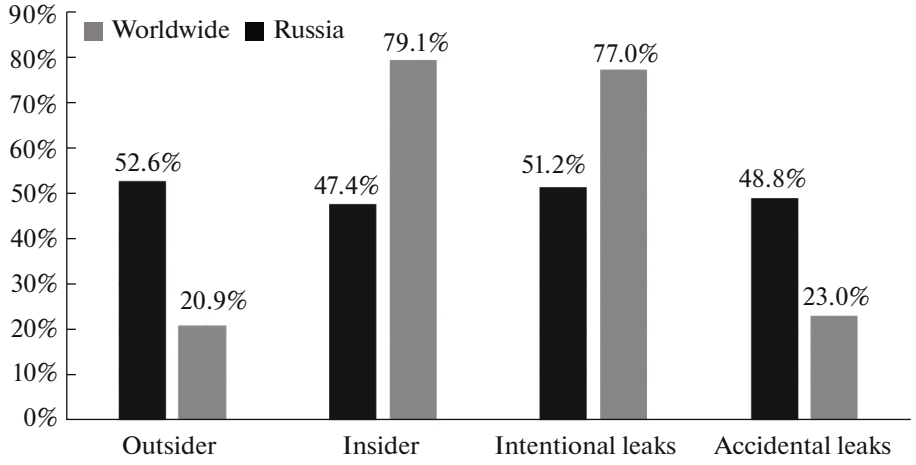

Fig. 1. Statistics of data leaks by source.

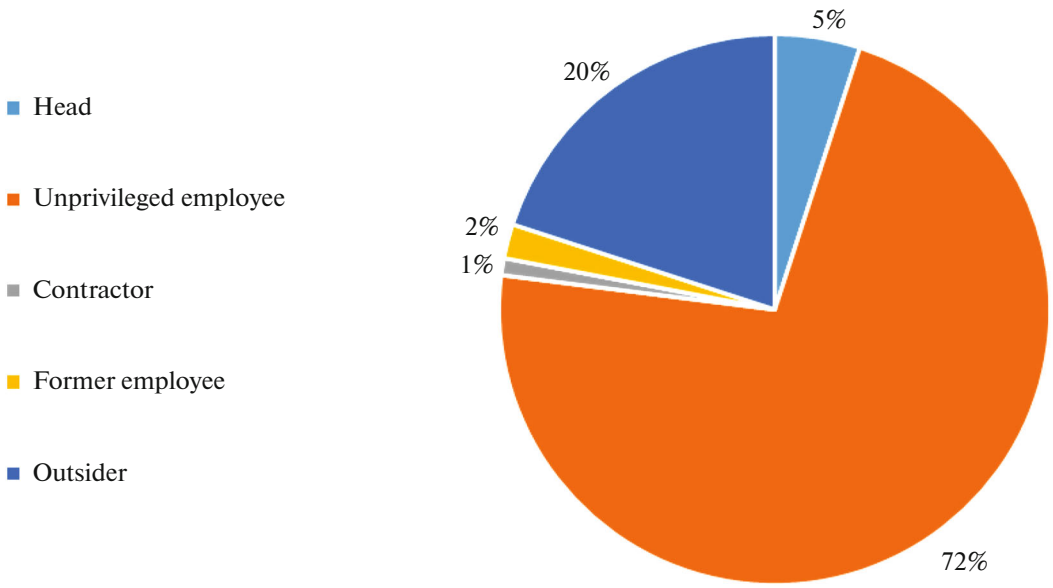

Fig. 2. Distribution of intruders by categories.

At the same time, major incidents became much more numerous; as a result of each of them, there was at least a million PD and payment data records leaks: from six in January—September 2019, the number of these incidents in the first nine months of 2020 grew to fifteen [1].

As noted in work [2], threats through the fault of insiders are the most dangerous threats for many organizations, including public establishments. In this case, malicious activities are taken by trusted persons inside organizations, which causes major damage.

The first part of this article considers the model of information security threats in corporate data transfer networks and formulates the study objective. The second part presents an algorithm of detecting encrypted files and the results of practical tests conducted to estimate the accuracy the classification of files by the developed algorithm.

1. A MODEL OF INFORMATION SECURITY THREATS CAUSED BY INSIDERS IN CORPORATE DATA TRANSFER NETWORKS

To evaluate the actions of insiders, a model of information security threats in corporate data transfer networks has been developed, and it is presented in Fig. 3.

According to this model, an insider can be an ordinary or a privileged corporate employee, or malware installed in a hidden manner. When confidential information is transferred beyond the corporate perimeter, the insider can use information encryption to transfer this information through data leak detection and prevention tools. A leak can occur, because the available DLP systems exhibit a low level of accuracy in detecting encrypted data due to their statistical similarity with data of various classes, such as compressed images, archives, video and audio files [3, 4].

The available DLP systems analyze the service information inherent to data transfer or on the basis of searching for various signatures and regular expressions directly in the data. Several works point out that encrypted or compressed confidential data cannot be detected [5, 6].

Several researchers note that there are no efficient and accurate methods of classifying high-entropy sources, for example, data encryption and compression algorithms [7, 8].

The insider has encryption and compression tools available for use, which allows concluding that the classification of encrypted and compressed data is a relevant task. There are several reasons why the considered approaches are not reliable solutions for detecting encrypted data transfers.

First of all, traffic analysis methods are inapplicable, because data transfer beyond the controlled corporate perimeter should never come to pass. It is necessary to develop tools for analyzing data before they are sent outwards, for example, after they are loaded to an email server.

Secondly, neural networks consider mainly file headings with magic bytes of high discriminative ability.

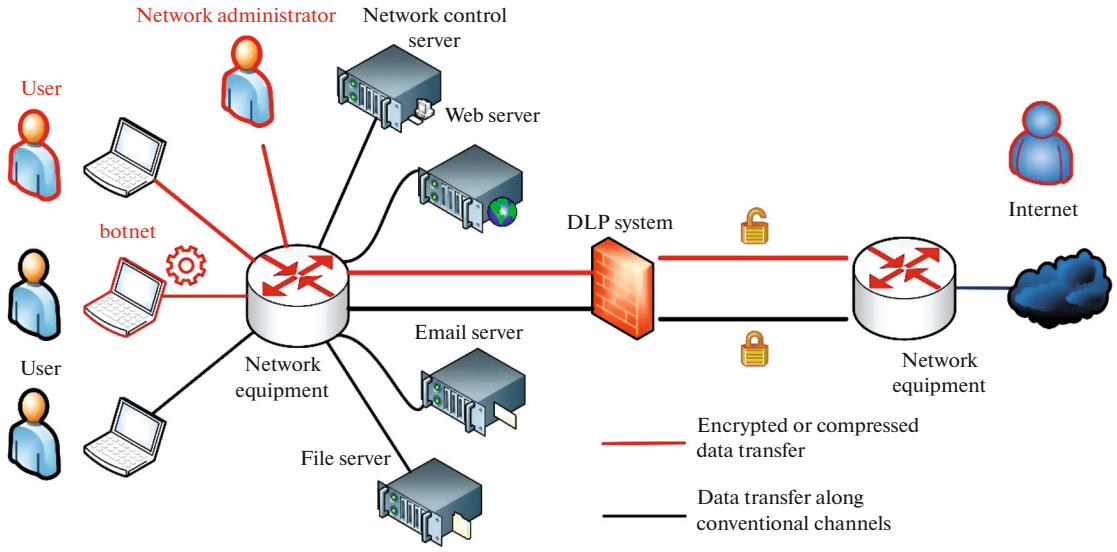

Fig. 3. Model of information security threats from insiders.

Thirdly, the considered entropic and other approaches also handle file headings with magic bytes.

Since data leak detection and prevention tools deal with insiders, there is usually open access to files and data due for being sent by the insider beyond the controlled corporate perimeter. This is why we should consider content methods of data analysis. For the results of analyzing a subject domain, see Table 1.

Despite a broad diversity of encrypted and compressed data classification methods, all of them have one common flaw: analysis of digital signatures contained in file headings.

Thus, the goal of this study is to develop an encrypted data detection algorithm that will allow classifying and separating to a high degree of accuracy encrypted and compressed files from open-access files that circulate in corporate networks and include office docs, images, and text data. The developed algorithm must not consider digital signatures and context information.

2. FILE CLASSIFICATION ALGORITHM

For the encrypted file detection algorithm, see Fig. 4.

The first step for ensuring that the algorithm works correctly is to define high-entropy files and files with even byte distributions, that is, encrypted and compressed files. Modules 1–7 separate potentially dangerous data from data legitimately used in corporate networks.

According to work [17], if a file's entropy exceeds 6.5, this file potentially contains encrypted or compressed sequences. Thus, module 1 calculates the entropy of analyzed data. If this threshold value is exceeded, the file is forwarded to module 3; otherwise, the analyzing process is finished.

Table 1. Results of analyzing content classifier investigations

Authors	Year	Features	Classification algorithm	Accuracy
[9]	2017	Statistical features	SVM	0.607
[10, 11]	2017	Byte distribution	GA, NS	0.98
[12]	2019	Byte distribution	CHC (VGG-16, ResNet)	0.999
[13]	2019	Word frequency	XGBoost	0.99
[14]	2019	NIST tests, data unit entropy	HEDGE (high entropy distinguisher)	0.72
[15]	2020	Byte distribution	HC	0.8-1
[16]	2020	Entropy	Hidden Markov chains	0.52

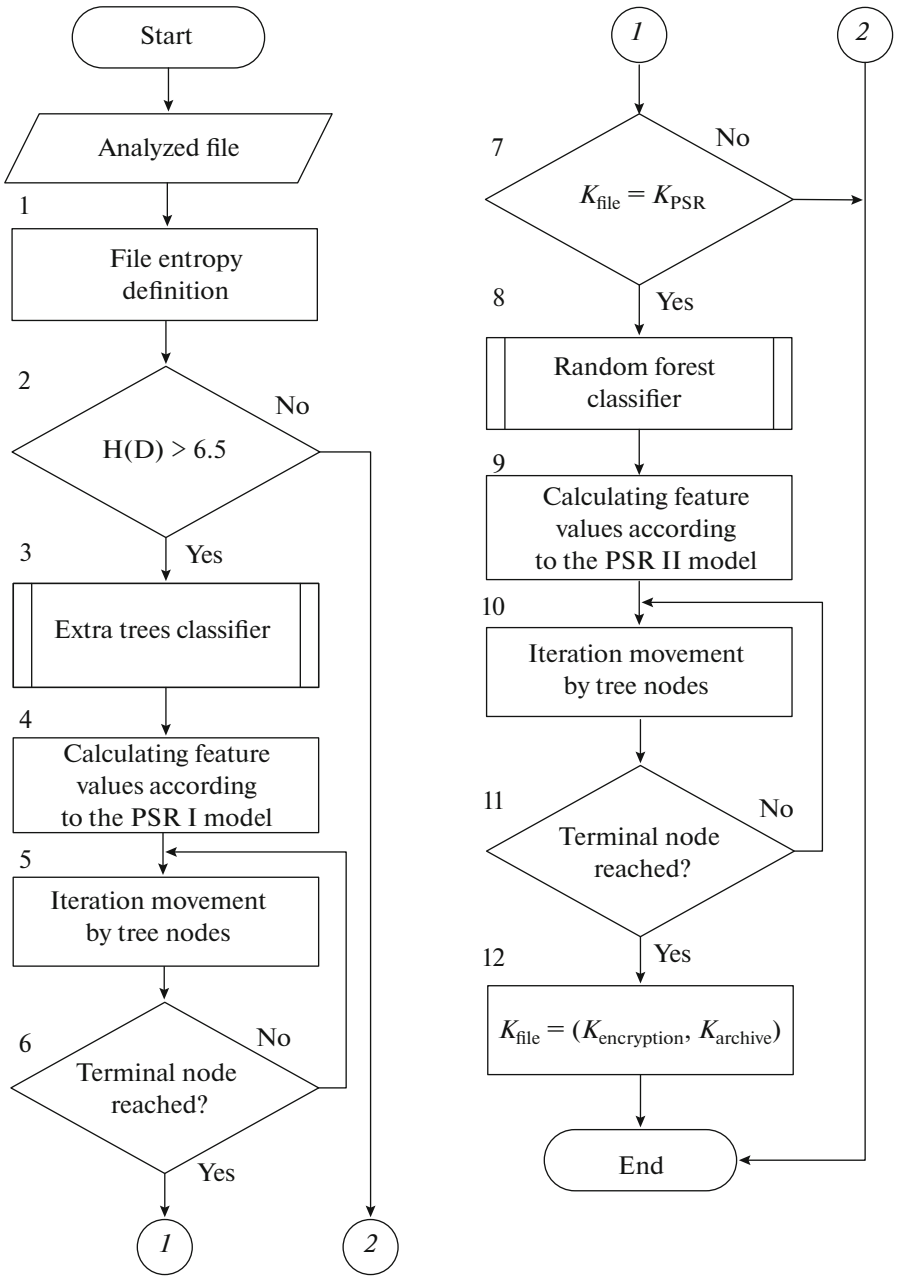

Fig. 4. Encrypted file detection algorithm.

In module 3 the file is analyzed by the trained classifier based on the extra trees algorithm. This stage implies training the classifier, setting up its hyperparameters, and defining the most significant features of the PSR model that allow classifying encrypted/compressed data as accurately as possible by separating them from other classes.

The features defined in module 3 are transferred to module 4, where the values of the features of the analyzed file are calculated by the model of pseudo-random sequences [18].

At stages 5 and 6 the iteration movement across the tree nodes is executed. The process is terminated when the terminal tree node is reached containing the respective class token assigned to the analyzed file. If the file is recognized as a pseudo-random sequence, the algorithm continues running; otherwise, the algorithm terminates.

In module 8 the analyzed file is transferred to the entry of the random forest classifier. At this stage, the classifier is trained, its hyperparameters are defined, and key features are calculated that allow classi-

Table 2. Accuracy estimation of four data classes by various machine learning algorithms

Algorithms	Metrics							Time (sec)
	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	
Random forest classifier	0.9998	1.0000	0.9992	0.9998	0.9998	0.9994	0.9994	41.612
Light gradient boosting machine	0.9998	1.0000	0.9992	0.9998	0.9998	0.9992	0.9992	76.886
Extra trees classifier	0.9998	1.0000	0.9989	0.9998	0.9998	0.9992	0.9992	76.886
Decision tree classifier	0.9996	0.9995	0.9991	0.9996	0.9996	0.9986	0.9986	14.370
Logistic regression	0.9994	0.9994	0.9972	0.9994	0.9994	0.9977	0.9977	53.388
Linear discriminant analysis	0.9978	0.9983	0.9921	0.9978	0.9978	0.9922	0.9922	14.620
Ridge classifier	0.9977	0.0000	0.9916	0.9977	0.9976	0.9918	0.9918	2.634

Table 3. Accuracy estimation of four data classes by the extra trees algorithm

Separation number	Accuracy	AUC	Recall	Prec.	F1	MCC
0	1.00	1.00	1.00	1.00	1.00	1.00
1	1.00	1.00	1.00	1.00	1.00	1.00
2	1.00	1.00	1.00	1.00	1.00	1.00
3	1.00	1.00	1.00	1.00	1.00	1.00
4	1.00	1.00	1.00	1.00	1.00	1.00
5	0.9994	1.00	0.9972	0.9994	0.9994	0.9979
6	1.00	1.00	1.00	1.00	1.00	1.00
7	1.00	1.00	1.00	1.00	1.00	1.00
8	1.00	1.00	1.00	1.00	1.00	1.00
9	0.9994	1.00	0.9972	0.9994	0.9994	0.9979
Mean	0.9999	1.00	0.9994	0.9999	0.9999	0.9996
SD	0.0002	0.00	0.0011	0.0002	0.0002	0.0008

fying encrypted and compressed sequences between each other; in other words, a binary operation is executed. The resulting features are transferred to block 9 for calculating their values in the analyzed file.

The analyzed file is classified in modules 10–12; as a result, this file is associated with a token of encrypted or compressed data.

The extra trees algorithm is chosen on the basis of the conducted experiments. For the results of estimating the accuracy of the multiclass classification of encrypted/compressed (aes, camellia, des, rc4, GOST 34.12 "Grasshopper", zip, rar, 7z, gz, xz, bz2) data, images (jpg), text (txt), and tabulated MS Office data (xls) see Table 2.

The conclusion derived proceeding from the findings is that, according to both, the accuracy metric and the other metrics, all of the tested algorithms exhibit a high level of accuracy. Since it was fairly difficult to opt for a specific algorithm with the help of metrics, we considered the temporal operating characteristic of those algorithms. The shortest learning time is characteristic of the extra trees classifier. Because the learning time is directly pro rata with the file classification time, this algorithm was chosen as the best one.

The chosen classifier was evaluated proceeding from the experiments on the basis of the earlier generated sample consisting of files of four classes. A crossover check was conducted on the basis of 10 divisions of data in subsamples. For the results of the check, see Table 3.

The average classification accuracy achieved using the resulting classifier with various metrics was 0.99.

For the boundaries separating the classes according to the two key features, see Fig. 5.

As follows from analyzing Fig. 5, it can be concluded that the extra trees classifier can very accurately separate encrypted/compressed, text, graphic, and tabular files.

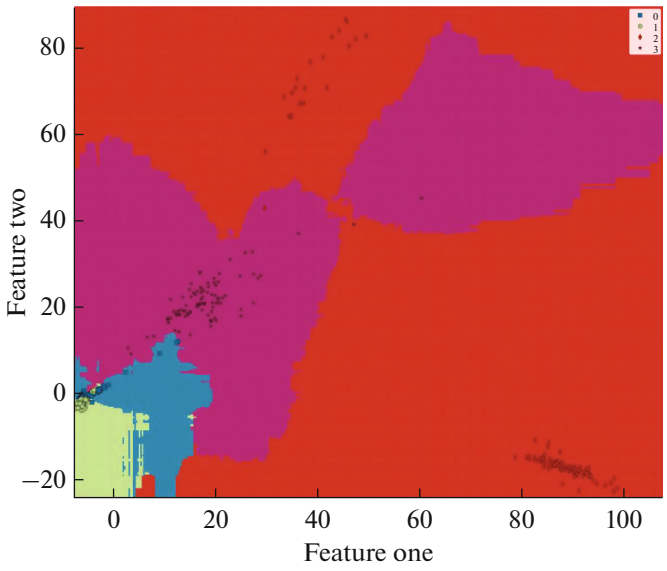

Fig. 5. Separating boundaries among the four classes.

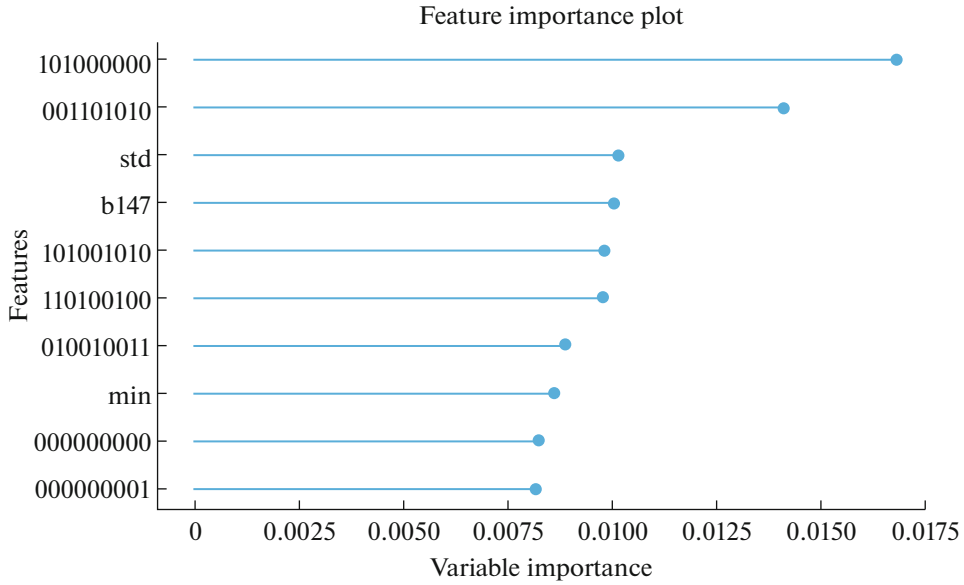

Fig. 6. Importance of features in using the extra trees algorithm.

The importance of features in the multiclass classification conducted by the extra trees algorithm is shown in Fig. 6.

The importance estimation of the features according to Shapley values is presented in Fig. 7. These values are calculated as

$$\omega_i(p) = \sum_{S \subseteq N/\{i\}} \frac{|S|!(n-|S|-1)!}{n!}(p(S \cup \{i\}) - p(S)),$$

where: $p(S \cup \{i\})$ is the model prediction based on the ith feature, $p(S)$ is the model prediction without this feature, n is the number of features, and S is the set of values without the ith feature.

The Shapley value for each feature is calculated for each file in the data sample; then, the calculated values are summed by module and the weight of each feature is defined.

The feature weight values calculated according to Shapley values allow selecting the most significant features that allow classifying highly entropic data from those legitimately used in the system and shrinking the analyzed space of features by rejecting the features with the lowest weights.

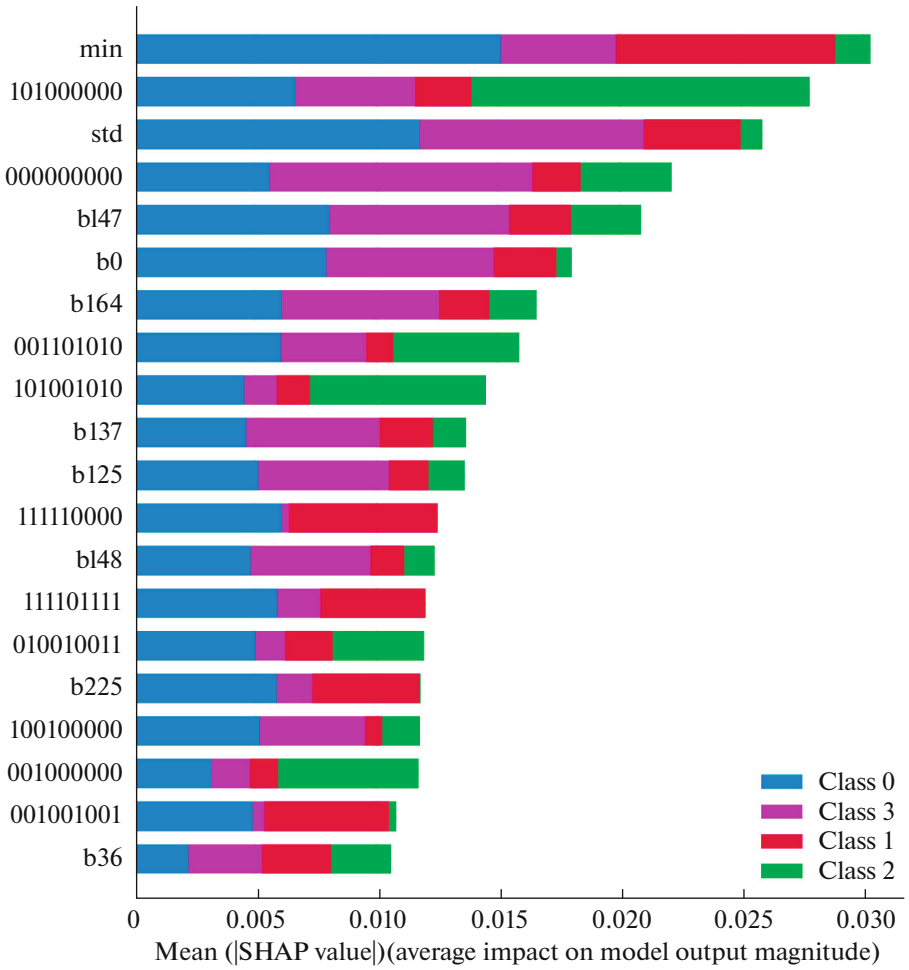

Fig. 7. Defining weight of features according to Shapley value.

CONCLUSIONS

The main contribution of this work is exposed below.

1. Several works on information security have been analyzed for examining the application of data classification methods and ML algorithms. The conclusion about the weaknesses of the existing approaches has been made; the requirements on the developed approach to classifying encrypted and compressed data before their transfer to the external network have been suggested.

2. The model of PRSs shaped by data encryption and compression algorithms (pseudo-random sequences) has been suggested that differs from its counterparts, considering the distribution of binary subsequences of N bits in length.

3. Limitations have been formed for use in actual work: the data chunks necessary for achieving the maximal accuracy of PRS classification must be fairly large and reach at least 600 KB. When chunks of about 50 KB (accuracy by metrics) are used, the fraction of right answers is 0.81. The strengths of the suggested PSR classification method are that the PSR model does not take into consideration file headings and magic bytes of compressed PSRs.

The developed approach has shown a high level of accuracy in classifying encrypted and compressed sequences of 0.97 and can be used to improve existing DLP systems or adopted in email servers for analyzing email attachments before they are sent beyond the corporate perimeter.

FUNDING

This study was financially supported by the Ministry of Science and Higher Education of the Russian Federation (grant IB), scientific project 18/2020.

CONFLICT OF INTEREST

The authors declare that they have no conflicts of interest.

REFERENCES

1. Leakages of restricted data access for 9 months of 2020. www.infowatch.ru/analytics/reports/utechki-informatsii-ogranichennogo-dostupa-otchet-za-9-mesyatsev-2020. Cited June 16, 2021.
2. Le, D.C., Zincir-Heywood, N., and Heywood, M.I., Analyzing data granularity levels for insider threat detection using machine learning, *IEEE Trans. Network Service Manage.*, 2020, vol. 17, no. 1, pp. 30–44. https://doi.org/10.1109/TNSM.2020.2967721
3. Yu, X., Tian, Z., Qiu, J., and Jiang, F., A data leakage prevention method based on the reduction of confidential and context terms for smart mobile devices, *Wireless Commun. Mobile Comput.*, 2018, vol. 2018, p. 5823439. https://doi.org/10.1155/2018/5823439
4. Karampidis, K., Kavallieratou, E., and Papadourakis, G., Comparison of classification algorithms for file type detection: A digital forensics perspective, *Polybits*, 2017, vol. 56, pp. 15–20. https://doi.org/10.17562/PB-56-2
5. Cheng, L., Liu, F., and Yao, D., Enterprise data breach: causes, challenges, prevention, and future directions, *WIREs Data Mining Knowl. Discov.*, 2017, vol. 7, p. e1211. https://doi.org/10.1002/widm.1211
6. Doroud, H., Aceto, G., de Donato, W., Jarchlo, E.A., Lopez, A.M., Guerrero, C.D., and Pescape, A., Speeding-up DPI traffic classification with chaining, *IEEE Global Communications Conf. (GLOBECOM),* Abu Dhabi, United Arab Emirates, 2018, IEEE, 2018, pp. 1–6. https://doi.org/10.1109/GLOCOM.2018.8648137
7. Hahn D., Apthorpe N., and Feamster N., Detecting compressed cleartext traffic from consumer internet of things devices, 2018. arXiv:1805.02722 [cs.CR]
8. Wood, D., Apthorpe, N., and Feamster, N., Cleartext data transmissions in consumer IoT medical devices, *Proc. 2017 Workshop on Internet of Things Security and Privacy,* Dallas, Tex., 2017, New York: Association for Computing Machinery, 2017, pp. 7–12. https://doi.org/10.1145/3139937.3139939
9. Wang, F., Quach, T.-T., Wheeler, J., Aimone, J.B., and James, C.D., Sparse coding for n-gram feature extraction and training for file fragment classification, *IEEE Trans. Inf. Forensics Secur.*, 2018, vol. 13, no. 10, pp. 2553–2562. https://doi.org/10.1109/TIFS.2018.2823697
10. Karampidis, K. and Papadourakis, G., File type identification-computational intelligence for digital forensics, *J. Digital Forensics, Secur. Law*, 2017, vol. 12, no. 2, p. 6. https://doi.org/10.15394/jdfsl.2017.1472
11. Srinivas, M., Nayak, A., and Bhatt, A., Forged file detection and stagnographic content identification (FFDASCI) using deep learning techniques, *CEUR Workshop Proc.*, Moscow, 2019, Basarab, M. and Markov, A.S., Eds., Moscow: CEUR Workshop Proceedings, 2019. http://ceur-ws.org/Vol-2380/paper_142.pdf
12. Konaray, S.K., Toprak, A., Pek, G.M., Akçekoce, H., and Kılınç, D., Detecting file types using machine learning algorithms, *Innovations in Intelligent Systems and Applications Conf.,* Izmir, Turkey, 2019, IEEE, 2019, pp. 1–4. https://doi.org/10.1109/ASYU48272.2019.8946393
13. Casino, F., Choo, K.-K.R., and Patsakis, C., HEDGE: Efficient traffic classification of encrypted and compressed packets, *IEEE Trans. Inf. Forensics Secur.*, 2019, vol. 14, no. 11, pp. 2916–2926. https://doi.org/10.1109/TIFS.2019.2911156
14. De Gaspari, F., Hitaj, D., Pagnotta, G., De Carli, L., and Mancini, L.V., EnCoD: Distinguishing compressed and encrypted file fragments, *Network and System Security. NSS 2020*, Kutyłowski, M., Zhang, J., and Chen, C., Eds., Lecture Notes in Computer Science, vol. 12570, Cham: Springer, 2020, pp. 42–62. https://doi.org/10.1007/978-3-030-65745-1_3
15. Mousavi, S.S., Detecting disk sectors data types using hidden Markov model, *17th Int. ISC Conf. on Information Security and Cryptology (ISCISC)*, Tehran, 2020, IEEE, 2020, pp. 60–64. https://doi.org/10.1109/ISCISC51277.2020.9261906
16. Matveeva, V.S., Information entropy and its application for information security tasks, *Bezop. Inf. Tekhnol.*, 2014, vol. 21, no. 3, pp. 30–36.
17. Kozachok, A.V. and Spirin, A.A., Model of pseudo-random sequences generated by encryption and compression algorithms, *Program. Comput. Software*, 2021, vol. 47, no. 4, pp. 249–260. https://doi.org/10.1134/S0361768821040058

Translated by S. Kuznetsov