Applying Big Data Processing and Machine Learning Methods for Mobile Internet of Things Security Monitoring

Igor Kotenko[1,2*], Igor Saenko[1,2], and Alexander Branitskiy[1,2]
[1]Laboratory of Computer Security Problems
of St. Petersburg Institute for Informatics and Automation (SPIIRAS),
14-th line, 39, Saint-Petersburg, 199178, Russia
[2]St. Petersburg National Research University of Information Technologies, Mechanics and Optics,
49, Kronverkskiy prospekt, Saint-Petersburg, Russia
{ivkote, ibsaen, branitskiy}@comsec.spb.ru

Abstract

The paper offers a new approach to Big Data processing for security monitoring of mobile Internet of things elements based on machine learning and its implementation using parallel algorithms. The architecture of security monitoring system is considered. It specifies several machine learning mechanisms intended for solving classification tasks. The classifier operation results are exposed to plurality voting, weight voting and soft voting. The experimental assessment of performance and accuracy of the offered methods is made.

Keywords: Big Data, Machine Learning, Security Monitoring, Mobile Security, Internet of Things, Classifier.

1 Introduction

Further development of the Internet of things (IoT) networks surely assumes distribution of the concept of IoT on mobile computing devices and creation of mobile IoT. Mobile IoT assumes that the computer network envelops not only traditional computer elements (servers, workstations, network devices) and electronic user devices ('things') of different types, but also the mobile computing devices which are connected to remaining elements of the IoT network by means of WiFi, mobile and/or sensor networks. Applications of the mobile IoT are continuously expanding. IoT finds successful application in such areas as medicine, transport control, security monitoring in public places, smart house/city, electricity consumption, industrial production, etc.

However successful development and effective functioning of the mobile IoT are impossible without solving a problem of the mobile IoT security monitoring. Security monitoring consists in continuous collecting the big arrays of heterogeneous data about security events taken place in mobile IoT networks. These data are exposed to the further analysis to detect the signs of possible harmful activity for the purpose of framing of timely measures of counteraction to the existing and perspective attacks against the infrastructure of mobile IoT. Solutions of this problem will promote the substantial increase of mobile IoT security.

One of the most perspective directions of solving this problem is sharing the results obtained in the field of Big Data processing and machine learning. The use of the frameworks implementing parallel stream data handling is of great interest In the field of Big Data processing. Methods and algorithms of machine learning allow one to find regularities in processed data and to solve different problems in

Journal of Internet Services and Information Security (JISIS), volume: 8, number: 3 (August 2018), pp. 54-63
*Corresponding author: Laboratory of Computer Security Problems of St. Petersburg Institute for Informatics and Automation (SPIIRAS), 14-th line, 39, Saint-Petersburg, 199178, Russia

the field of Data Mining in the automated mode. Combining of opportunities of Big Data and machine learning frameworks is considered as a rather perspective direction of solving the online analysis problem for data about mobile IoT security.

The main contribution of the paper is as follows: (1) creation of a framework in which the possibilities of Big Data processing and machine learning algorithms are integrated; (2) implementation in this framework the procedures of analyzing the reference data set, containing data on the mobile IoT traffic; (3) the experimental assessment of the developed framework.

The further structure of the paper is as follows. Section 2 lists some of related works. In section 3 the description of the used data set and the proposed framework architecture are provided. Section 4 contains the results of experiments. Section 5 contains the conclusion and the directions of future research.

2 Related Work

Related works can be divided into three groups: (1) applying machine learning for computer security, (2) development of Big Data frameworks, and (3) implementation of machine learning algorithms.

The works of the first group show that the problem of application of machine learning methods for computer security became extremely relevant. [6] shows that the use of computers in mobile networks (including the mobile IoT) became more ubiquitous and connected, and the attacks against these networks became more pervasive and diverse. Conventional security software requires a lot of human efforts to identity threats in such networks. This labor-intensive process can be more efficient by applying machine learning algorithms. [23] shows a successful implementation of one of the machine learning algorithms, namely a distributed Support Vector Machine (SVM), in order to detect malicious software (malware) in a network of mobile devices. [20] represents a machine learning based system to detect malware in Android devices which is also based on SVM algorithms. However, the bigger effect can be achieved by realizing combined applications of different machine learning algorithms. The manifesto [14] directly claims that one of perspective directions for application of machine learning is the protection of mobile devices together with assisted malware analysis. [10] reviews examples of intrusion detection systems, which are specifically based on machine learning methods due to their adaptability to new and unknown attacks. Among possible attacks, the attacks against mobile IoT networks, such as smart meter energy consumption profiling and surveillance camera robbery, are considered. The analysis of the known works on application of machine learning methods for detection of mobile threats, which is carried out in [1], showed that, firstly, the SVM algorithms have the greatest popularity, and, secondly, that enjoys success rate of the detection systems conducted with machine learning methods are in between of 80–99.6%. It is necessary to mark that in our work we reached bigger value of this index. The IoT security solutions based on machine learning methods were investigated in [25]. These methods included supervised learning, unsupervised learning and reinforcement learning and were applied for IoT authentication, access control, secure offloading and maleware detection. The results of the investigation show that one of main challenge has to be addressed to implement the machine learning based security methods in practical IoT systems are computation and communication overheads. In our proposed framework we set a goal to overcome this challenge.

Development of frameworks for Big Data is considered in many works. As a rule, these frameworks execute the MapReduce operations and are based on the special program systems as Hadoop, Spark, Flink, etc. [12, 24]. At the same time these frameworks have rather big scope. The Big Data frameworks, which are used in mobile networks and mobile IoT for processing the web applications [21], medical data [15], data on bus traffic control [27], are known. The Big Data frameworks developed for the benefit of computer security [9, 11, 16, 17] also exist. However, the machine learning algorithms in these frameworks are not used efficiently. An approach for combination of several classifiers has been already

suggested by the authors [2, 3], but the aspects of Big Data processing have not been considered.

Machine learning algorithms are applied to solve different Data Mining tasks. For classification tasks, the most acceptable algorithms are K-Nearest Neighbors [13], Naive Bayes [26], and SVM [8]. The regression problems are solved with the help of Linear Regression [22], Random Forests [5], and Bagging [4] algorithms. The algorithms of K-Means [7] and Density-Based Spatial Clustering of Applications with Noise are applied in solving the clustering problem [18]. These and other machine learning algorithms are actively developed. However, their implementation in parallel computing systems, especially for security problems, is considered poorly. Therefore, the results considered in the paper are relevant.

3 Data Set and Framework Architecture

3.1 Data Set and its Preprocessing

For carrying out the experiments we have used the data set "detection_of_IoT_botnet_attacks_N_BaIoT" [19]. This data set was generated on the basis of network traffic transmitted between 9 IoT devices. The collected records were made in the form of archived csv-files in which each line consists of 115 attributes separated by a comma. Each record belongs one of 11 classes among which 10 classes are attacks and 1 class is a benign class.

In total, the data set contains 7009270 records, some of which are duplicated (namely instances of classes gafgyt tcp and gafgyt udp). Therefore, the first step of preprocessing the data set is to remove the identical records. Such a procedure will allow to train classifiers using different instances and thus to provide the best coverage of the training sample. This allowed to exclude from consideration 115347 records. The next step is applying the min-max normalization of each attribute. Such a procedure is aimed at reducing the strong variability in the values of individual parameters, which will allow to consider all parameters within the same range of values. The third step is a principal component analysis (PCA) which compresses input vectors to vectors of smaller dimension without loss of significant informativeness about the parameters of the original vector. For this purpose, a linear transformation of the centered vector is performed using the eigenvectors of the covariance matrix of the training sample.

Table 1 contains the size characteristics of the investigated data set.

Figures 1 and 2 demonstrate projection of one of the training subsamples onto the first two and three principal components for the device Danmini Doorbell.

From these figures it can be seen that the training subsamples of different classes can be closely located, which complicates the process of constructing the classification model. Therefore, we should consider a higher-dimensional space, e.g. a 10-dimensional space of features.

Figure 3 shows the absolute correlation dependence of the first 10 principal components and the class label. In particular, the last column depicts the dependence degree between components and predicted class labels. This figure shows that the components themselves are independent of each other (the value of the correlation coefficient is close to zero), and the second component is most significant for recognizing the class label (the absolute value of their pairwise correlation is equal to 0.724).

3.2 Framework Architecture Description

Figure 4 shows the framework architecture designed for mobile Internet of things security monitoring. In includes three levels: decomposition of data set, compression of feature vectors and classification of instances. On the first level an initial data set is divided into several smaller sets that will allow to process each of these chunks in parallel mode without using locks (semaphores or mutexes). The second level

Table 1: Size characteristics of the investigated data set

Classes		Devices								
		Danmini Doorbell	Ecobee Thermostat	Ennio Doorbell	Philips B120N10 Baby Monitor	Provision PT 737E Security Camera	Provision PT 838 Security Camera	Samsung SNH 1011 N Webcam	SimpleHome XCS7 1002 WHT Security Camera	SimpleHome XCS7 1003 WHT Security Camera
benign traffic	all	40395	13111	34692	160137	55169	91555	46817	42784	17936
	unique	40395	13111	34692	160137	55169	91555	46817	42784	17936
gafgyt combo	all	59718	53012	53014	58152	61380	57530	58669	54283	59398
	unique	59718	53012	53014	58152	61380	57530	58669	54283	59398
gafgyt junk	all	29068	30312	29797	28349	30898	29068	28305	28579	27413
	unique	29068	30312	29797	28349	30898	29068	28305	28579	27413
gafgyt scan	all	29849	27494	28120	27859	29297	28397	27698	27825	28572
	unique	29849	27494	28120	27859	29297	28397	27698	27825	28572
gafgyt tcp	all	92141	95021	101536	92581	104510	89387	97783	88816	98075
	unique	85227	87877	93903	85628	96658	82687	90454	82147	90709
gafgyt udp	all	105874	104791	103933	105782	104011	104658	110617	103720	102980
	unique	100182	99195	98355	100093	98424	99028	104683	98181	97438
mirai ack	all	102195	113285	–	91123	60554	57997	–	111480	107187
	unique	102195	113285	–	91123	60554	57997	–	111480	107187
mirai scan	all	107685	43192	–	103621	96781	97096	–	45930	43674
	unique	107685	43192	–	103621	96781	97096	–	45930	43674
mirai syn	all	122573	116807	–	118128	65746	61851	–	125715	122479
	unique	122573	116807	–	118128	65746	61851	–	125715	122479
mirai udp	all	237665	151481	–	217034	156248	158608	–	151879	157084
	unique	237665	151481	–	217034	156248	158608	–	151879	157084
mirai udpplain	all	81982	87368	–	80808	56681	53785	–	78244	84436
	unique	81982	87368	–	80808	56681	53785	–	78244	84436
all instances		7009270								
unique instances		6893923								

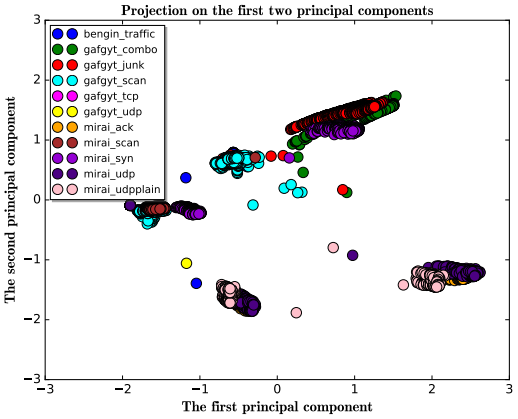

Figure 1: Projection of the training sample onto the first two principal components

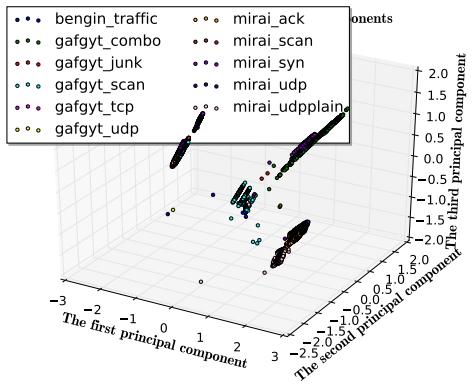

Figure 2: Projection of the training sample onto the first three principal components

implies usage of PCA. At the third level the classifiers are placed, which first perform the adjustment of their parameters, i.e. learn, and then predict the class label of the analyzed instance by its features.

A multi-level scheme for combining the classifiers was used for carrying out the experiments. Its representation is shown in Figure 5.

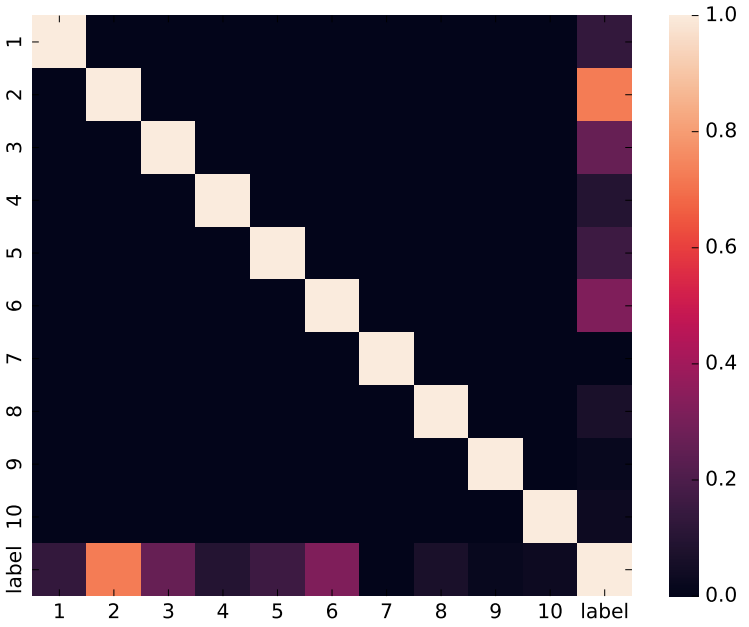

Figure 3: Correlation dependence on the training set containing the first 10 components and the class label

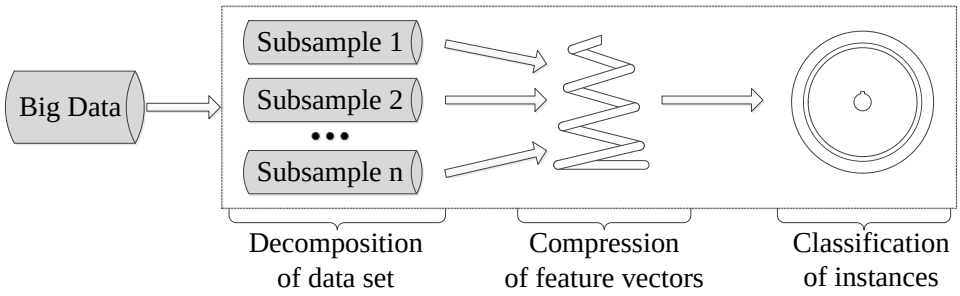

Figure 4: Framework architecture for Big Data processing based machine learning methods

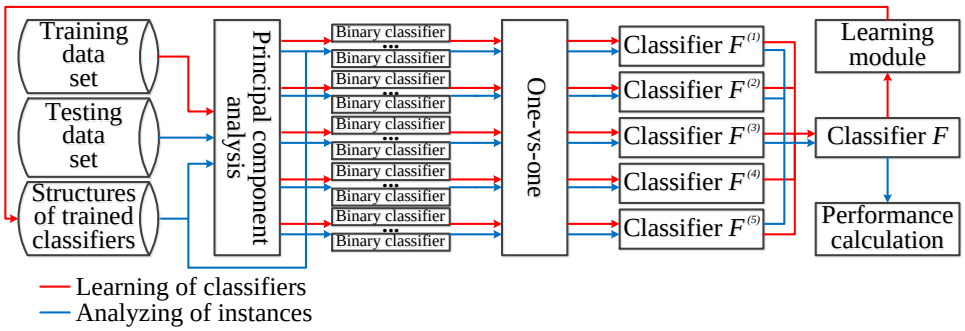

Figure 5: Multi-level scheme for combining the classifiers

After processing by PCA the input vector is analyzed using various binary classifiers. In the role of such classifiers we have used the support vector machine (SVM), k-nearest neighbors method (k-NN), gaussian naïve Bayes (GNB), artificial neural network (ANN) and decision tree (DT). Each of these classifiers was duplicated $\binom{11}{2} = 55$ times where 11 means a total number of classes. Each copy of the specific classifier is trained using a subsample containing only two classes. The usage of such a fragmentation of the training set allows to reduce the time costs for training of classifiers by introducing a parallel mode, and also to tune classifiers more sensitive to recognizing objects belonging to two classes (instead of eleven). The created binary classifiers are combined into a multi-class model $F^{(i)}$ using the one-vs-one approach ($i = 1, \ldots, 5$). The resulting classification is performed using a classifier F constructed on the basis of the plurality voting (PV), weight voting (WV) or soft voting (SV). After completing the training procedure the structures of classifiers are stored for the possibility of their deserialization and their performance calculation in the mode of analyzing of instances.

4 Experiments and Discussion

The training data set consists of 27500 unique instances (2500 unique instances per class). The testing data set was formed from the remaining unique elements which were not seen in the training set. The training and testing processes were performed seven times for each of nine IoT devices, and each time the random generation of the contents of the training set was provided. In the role of performance indicators the accuracy (*ACC*) and difference of true positive rate and false positive rate ($TRP - FPR$) were used. Table 2 contains maximum values of performance indicators calculated for five basic classifiers and their combinations.

Table 2: Maximum values of performance indicators of classifiers and their combinations

Classifiers $F^{(1)}, \ldots, F^{(5)}, F$ and performance indicators *ACC* and $TPR - FPR$		Devices								
		Danmini Doorbell	Ecobee Thermostat	Ennio Doorbell	Philips B120N10 Baby Monitor	Provision PT 737E Security Camera	Provision PT 838 Security Camera	Samsung SNH 1011 N Webcam	SimpleHome XCS7 1002 WHT Security Camera	SimpleHome XCS7 1003 WHT Security Camera
SVM	*ACC*	90.8382%	98.0729%	71.1154%	89.8452%	97.2226%	87.6968%	99.2009%	88.5611%	88.1011%
	$TPR - FPR$	99.8995%	99.8572%	99.8734%	99.8919%	99.715%	99.8098%	99.8621%	gray!4099.8204%	gray!4099.8283%
k-NN	*ACC*	99.1377%	97.1443%	99.4354%	96.8944%	97.2106%	97.4817%	99.1488%	98.1248%	97.6554%
	$TPR - FPR$	99.8406%	99.7115%	99.768%	99.8746%	99.6782%	99.7548%	99.7588%	99.5898%	99.7527%
GNB	*ACC*	75.6666%	71.4082%	64.2376%	78.8012%	72.9288%	75.9799%	66.0597%	69.9567%	68.1603%
	$TPR - FPR$	99.4261%	99.5928%	99.3554%	99.3009%	99.6172%	99.6609%	99.7334%	98.1972%	99.172%
ANN	*ACC*	90.8075%	88.6726%	71.3483%	91.1945%	86.6745%	88.7023%	73.699%	89.4466%	88.0869%
	$TPR - FPR$	99.6634%	99.6577%	99.6457%	99.3168%	99.5923%	99.7528%	99.761%	99.5902%	99.5274%
DT	*ACC*	98.9457%	97.5543%	gray!4099.5212%	97.9707%	gray!4097.4918%	98.0422%	gray!4099.5082%	97.6697%	97.3664%
	$TPR - FPR$	gray!4099.9122%	gray!4099.8919%	gray!4099.8828%	gray!4099.9228%	gray!4099.8378%	gray!4099.8583%	gray!4099.8967%	99.6671%	99.806%
PV	*ACC*	99.4611%	98.9797%	99.4803%	98.3458%	95.9375%	98.8028%	99.39%	gray!4099.211%	99.1102%
	$TPR - FPR$	99.8691%	99.7947%	99.8341%	99.8813%	99.7109%	99.7927%	99.8253%	99.7464%	99.8072%
WV	*ACC*	99.4749%	98.9523%	99.4803%	gray!4098.3464%	95.9631%	98.8423%	99.39%	99.1908%	99.0829%
	$TPR - FPR$	99.8694%	99.8023%	99.8341%	99.887%	99.704%	99.8001%	99.8293%	99.7529%	99.808%
SV	*ACC*	gray!4099.5026%	gray!4099.0225%	99.4639%	98.3362%	95.9651%	gray!4098.8498%	99.363%	99.1911%	gray!4099.1385%
	$TPR - FPR$	99.8643%	99.7955%	99.8368%	99.8828%	99.7072%	99.7929%	99.7881%	99.7525%	99.7742%

For six IoT devices out of nine possible, an increase of the indicator *ACC* is observed in the case of using combined classifiers PV, WV, SV in comparison with the usage of separate basic classifiers SVM, k-NN, GNB, ANN, DT. If we consider a specific combined classifier, namely SV, then we obtain a total increase in the indicator *ACC* by 3.3072% in comparison with the greatest value of the indicator *ACC* demonstrated among the basic classifiers.

To accelerate the analysis of records, the data set was divided into n approximately equal chunks ($n = 1, \ldots, 8$), each of which was processed by a separate parallel thread. Figures 6 and 7 demonstrate dependence of time of processing the data sets on the number of threads. In the case of a training set the rate of data processing was increased in 7.065 times in the transition from one thread to eight, and in the case of a testing set — 6.296 times.

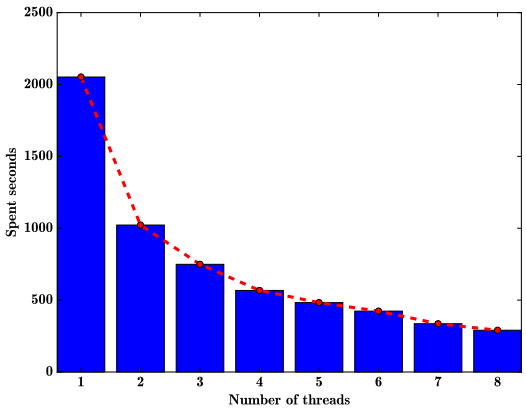

Figure 6: Dependence of time of the training data set (27500 instances) processing on the number of threads

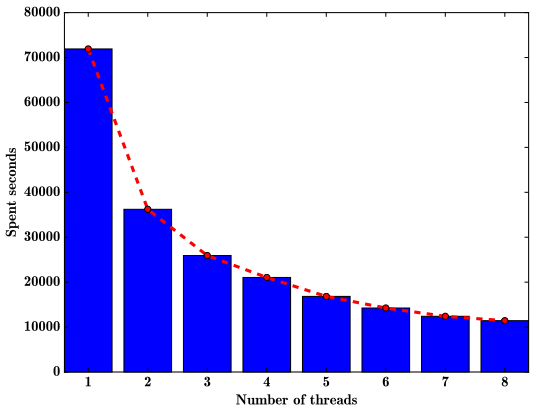

Figure 7: Dependence of time of the testing data set (969039 instances) processing on the number of threads

5 Conclusion

The paper presents the new approach to mobile IoT security monitoring based on using methods of Big Data processing and machine learning. The architecture of the Big Data and Machine Learning framework in which thread functioning of five machine learning mechanisms intended for solving the classification problem is realized is offered. Classifier operation results are exposed to majority voting.

The experimental assessment of performance and accuracy of the framework operation is made on the data set generated on the basis of the network traffic transmitted between mobile IoT devices. Assessment showed that the offered framework provides the gain in information processing productivity and the higher accuracy of detection of attacks. It says about sufficient efficiency of the developed framework and high prospects of integration of Big Data and machine learning algorithms.

The further direction of researches relates with implementation of the developed framework in the environment of the special software such as Hadoop and Spark.

Acknowledgments

This research was partially financially supported by grants of RFBR (projects No. 16-29-09482, 18-07-01369 and 18-07-01488), by the budget (the project No. AAAA-A16-116033110102-5), and by

Government of the Russian Federation, Grant 08-08.

References

[1] B. Arslan, S. Gunduz, and S. Sagiroglu. A review on mobile threats and machine learning based detection approaches. In *Proc. of the 4th International Symposium on Digital Forensic and Security (ISDFS'16), Little Rock, Arkansas, USA*, pages 7–13. IEEE, April 2016.

[2] A. Branitskiy and I. Kotenko. Hybridization of computational intelligence methods for attack detection in computer networks. *Journal of Computational Science*, 23:145–156, August 2016.

[3] A. Branitskiy and I. Kotenko. Network anomaly detection based on an ensemble of adaptive binary classifiers. In *Proc. of the 7th International Conference on Mathematical Methods, Models, and Architectures for Computer Network Security (MMM-ACNS'17), Warsaw, Poland*, volume 10446 of *Lecture Notes in Computer Science*, pages 143–157. Springer-Verlag, August 2017.

[4] L. Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, August 1996.

[5] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, October 2001.

[6] P. K. Chan and R. P. Lippmann. Machine learning for computer security. *Journal of Machine Learning Research*, 7:2669–2672, December 2006.

[7] A. Coates and A. Y. Ng. Learning feature representations with k-means. In G. Montavon, G. B. Orr, and K.-R. Müller, editors, *Neural networks: Tricks of the trade*, pages 561–580. Springer, second edition, January 2012.

[8] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, January 1995.

[9] P. Derbeko, S. Dolev, E. Gudes, and S. Sharma. Security and privacy aspects in mapreduce on clouds: A survey. *Computer Science Review*, 20:1–28, May 2016.

[10] V. Ford and A. Siraj. Applications of machine learning in cyber security. In *Proc. of the 27th International Conference on Computer Applications in Industry and Engineering (CAINE'14), New Orleans, Louisiana, USA*, pages 27–32. ISCA, October 2014.

[11] C. P. Garware and B. A. Tidke. A security framework for big data computing through distributed cloud data centres in g-hadoop. *International Journal of Computer Science and Mobile Computing*, 5(6):355–360, June 2016.

[12] A. Holmes. *Hadoop in practice*. Manning Publications Co., 2012.

[13] H. V. Jagadish, B. C. Ooi, K.-L. Tan, C. Yu, and R. Zhang. idistance: An adaptive b+-tree based indexing method for nearest neighbor search. *ACM Transactions on Database Systems*, 30(2):364–397, June 2005.

[14] A. D. Joseph, P. Laskov, F. Roli, J. D. Tygar, and B. Nelson. Machine learning methods for computer security. *Dagstuhl Manifestos*, 2(9):109–130, February 2013.

[15] M.-J. Kim and Y.-S. Yu. Development of real-time big data analysis system and a case study on the application of information in a medical institution. *International Journal of Software Engineering and Its Applications*, 9(7):93–102, July 2015.

[16] I. Kotenko, A. Kuleshov, and I. Ushakov. Aggregation of elastic stack instruments for collecting, storing and processing of security information and events. In *Proc. of the 14th IEEE Conference on Advanced and Trusted Computing (ATC'17), San Francisco, California, USA*, pages 1–8. IEEE, August 2017.

[17] I. V. Kotenko, I. Saenko, and A. Kushnerevich. Parallel big data processing system for security monitoring in internet of things networks. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, 8(4):60–74, December 2017.

[18] H.-P. Kriegel, P. Kröger, J. Sander, and A. Zimek. Density-based clustering. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(3):231–240, April 2011.

[19] Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, D. Breitenbacher, A. Shabtai, and Y. Elovici. N-baiot: Network-based detection of iot botnet attacks using deep autoencoders. *IEEE Pervasive Computing*, 13(9), May 2018.

[20] J. Sahs and L. Khan. A machine learning approach to android malware detection. In *Proc. of the 2012 European Intelligence and Security Informatics Conference (EISIC'12), Odense, Denmark*, pages 141–147.

IEEE, August 2012.

[21] M. Scherbakov, D. Kachalov, V. Kamaev, N. Scherbakova, A. Tyukov, and S. Strekalov. A design of web application for coomplex event processing based on hadoop and java servlets. *International Journal of Soft Computing*, 10(3):218–219, January 2015.

[22] G. A. Seber and A. J. Lee. *Linear Regression Analysis*. John Wiley & Sons, second edition, January 2012.

[23] A. S. Shamili, C. Bauckhage, and T. Alpcan. Malware detection on mobile devices using distributed machine learning. In *Proc. of the 20th International Conference on Pattern Recognition (ICPR'10), Istanbul, Turkey*, pages 4348–4351. IEEE, August 2010.

[24] A. G. Shoro and T. R. Soomro. Big data analysis: Apache spark perspective. *Global Journal of Computer Science and Technology: Software & Data Engineering*, 15(1):1–8, January 2015.

[25] L. Xiao, X. Wan, X. Lu, Y. Zhang, and D. Wu. Iot security techniques based on machine learning. *CoRR*, abs/1801.06275, January 2018.

[26] H. Zhang. The optimality of naive bayes. In *Proc. of the 2004 Seventeenth International Florida Artificial Intelligence Research Society Conference (SIFAIRS'04), Miami Beach, Florida, USA*, pages 562–567. AAAI Press, January 2004.

[27] N. Zygouras, N. Zacheilas, V. Kalogeraki, D. Kinane, and D. Gunopulos. Insights on a scalable and dynamic traffic management system. In *Proc. of the 18th International Conference on Extending Database Technology (EDBT'15), Brussels, Belgium*, pages 653–664. EDBT, March 2015.

Author Biography

Igor Kotenko graduated with honors from St.Petersburg Academy of Space Engineering and St. Petersburg Signal Academy. He obtained the Ph.D. degree in 1990 and the National degree of Doctor of Engineering Science in 1999. He is Professor of computer science and Head of the Laboratory of Computer Security Problems of St. Petersburg Institute for Informatics and Automation. He is the author of more than 200 refereed publications. Igor Kotenko has a high experience in the research on computer network security and participated in several projects on developing new security technologies. For example, he was a project leader in the research projects from the US Air Force research department, via its EOARD (European Office of Aerospace Research and Development) branch, EU FP7 and FP6 Projects, HP, Intel, F-Secure, etc. The research results of Igor Kotenko were tested and implemented in more than fifty Russian research and development projects. The research performed under these contracts was concerned with innovative methods for network intrusion detection, simulation of network attacks, vulnerability assessment, security protocols design, verification and validation of security policy, etc. He has chaired several International conferences and workshops, and serves as editor on multiple editorial boards.

Igor Saenko graduated with honors from St. Petersburg Signal Academy. He obtained the Ph.D. degree in 1992 and the National degree of Doctor of Engineering Science in 2001. He is Professor of computer science and Leading Researcher of the Laboratory of Computer Security Problems of St. Petersburg Institute for Informatics and Automation. He is the author of more than 200 refereed publications and participated in several Russian and international research projects. His main research interests are security policy management, access control, management of virtual computer networks, database management, knowledge modeling, soft and evolutionary computation, information and telecommunication systems.

Alexander Branitskiy graduated with honors from the Department of Mathematics and Mechanics of St. Petersburg State University with a degree in "Software and Administration of Information Systems" in 2012. He is a junior researcher of Laboratory of Computer Security Problems of St. Petersburg Institute for Informatics and Automation. Main research interests are information security, artificial intelligence, and computer networks.