Application of Intelligent Data Analysis Methods for Information Security Problems

Vadim Vagin(✉), Sergey Antipov, Marina Fomina, and Oleg Morosin

Department of Application Mathematics, National Research University "MPEI", Moscow, Russian Federation
vagin@appmat.ru, antisergey@gmail.com, m_fomina2000@mail.ru, omorsik@gmail.com

Abstract. In this paper, we consider the ideas and approaches to the intelligent data analysis methods in problems of information security. Solving network security problems is a complex task, involving the large number of factors and requiring finding reasonable compromises between maintaining security, the stable work, enhancing operating expenses and functional restrictions of complex information systems. There are considered the ways to apply inductive concept formation methods for analyzing network traffic, as well as argumentation methods for an automated support of security solutions. The proposed approach allows to give numerical assessments of the quality of recommendations developed by the system, thereby helping to solve an important task - the task of choosing the way to react to suspicious activity in the system. In addition, the example of handling the dangerous situations arising in the system is given.

1 Introduction

Up-to-date computer networks are large distributed systems of software and devices interacting with each other to exchange, store and process information. Networks connect different types of devices by communication channels. The intensified loading the networks, their complication and the increased number of attacks on networks are the reason of importance of researches of network security problems. In connection with the urgent need to ensure the security of computers interacting in the networks, the system development able to recognize the suspicious network activity becomes very important. Many up-to-date network security systems use only certain rules, laid down in them manually based on the analysis of expert experience, and do not have the possibility of self-learning. To increase the security of the systems through rapid detection of attacks, methods of intelligent data analysis are currently used. They can effectively operate in the face of unexpected situations, not described by experts. Let's list the main tasks that can be solved with the help of intelligent data analysis methods in the field of information security.

(1) Rapid recognition of threats based on the analysis of information stored in the Knowledge Base of an intelligent system.

A. Abraham et al. (eds.), *Proceedings of the Second International Scientific Conference "Intelligent Information Technologies for Industry" (IITI'17)*, Advances in Intelligent Systems and Computing 679, DOI 10.1007/978-3-319-68321-8_2

(2) Optimization of the search for malicious sources and fighting with malicious software (malware), which can also be self-learning.
(3) Generalize information in the learning process, and build new rules for detecting malware on the basis of gained experience for a more powerful protection system.

The paper is organized as follows. In Sect. 1, the main concepts of the argumentation theory will be given and an example of the application of the argumentation system for solving the problem of selecting the control action in the presence of suspicious activity will be presented. In Sect. 2, generalization methods will be considered for the problem of finding anomalies in the analysis of network traffic. Algorithms for finding anomalies in sets of time series, as well as the results of machine experiments, are presented. At last, conclusions on the work are given.

2 Application of Argumentation Methods for the Choice of Control Actions in the Presence of Suspicious Network Activity

Modern security systems must handle large amounts of information, that is often noisy and contradictory. Often security systems use mechanisms of classical logic (see, for example, [1, 2]) to describe the rules of invasion detection and other suspicious actions of users in information systems. At the same time, false triggering of protection mechanisms is extremely undesirable, since it can lead to significant financial losses. The simplest example is firewalls, containing thousands [1] of rules for detecting suspicious activity, under triggering of which the blocking of data exchange with the user is brought about. Reducing the percentage of false actions of protection systems is an important problem [3], the solution of which will significantly improve the quality of computer security systems. Usually the question of the data protection is considered only as the fact detection of the invasion and the question of possible consequences of false triggering protection systems is usually not viewed [4]. Here, it is proposed to describe the mechanism for decision making not only in terms of assessing the plausibility of malicious user actions, but also in terms of assessing the potential risks from the application of protection mechanisms. For this purpose, it is suggested to use the mechanism of argumentation theory to determine the need for protective measures taking into account possible risks from false triggering protection systems. Argumentation is a formal approach to decision making that has proved its efficiency in a number of areas. Argumentation is usually understood as the process of constructing assumptions about a certain analyzable problem. Typically, this process involves conflicts detection and ways of problem solutions.

We will consider the argument as a pair consisting of a set of premises and a conclusion [5]. All interrelations between arguments will be represented on an inference graph. It is a graph that shows a way of building new arguments from already existing ones and the conflicts between arguments. Besides arguments, argumentation system contains the inference rules of two types. *Defeasible rules* are the inference rules whose reliability is questionable. In the natural language, such rules are formulated by expressions such as “as a rule”, “usually”, “normally”, “likely”, etc. The arguments obtained by such rules are called defeasible. Defeasible rules are denoted by $M \mid\Rightarrow N$. In inference graphs, they

are depicted by dashed arrows, and defeasible arguments are depicted by double ovals. *Undercutting rules* are the inference rules that formalize the conflict of the defeat type. These rules indicate that there are arguments that defeat the connection between two other arguments connected by a defeasible implication. For example, there is an argument E that undercuts the defeasible relationship $C \mid\Rightarrow D$ between the arguments C and D. Such undercutting rules are written in the form $E \Rightarrow (C @ D)$. In inference graphs, the undercutting arguments and the arguments defeated by them are connected by a bold dashed arrow.

To apply the theory of argumentation in problems requiring the calculation of quantitative assessments of the argument reliability (such as, for example, network security tasks) the mechanism of justification degrees for arguments is applied. Argumentation systems with the justification degrees were considered in [6]. Let's present a simplified example

Let us consider an example of the network security problem given in [4]. For clarity, the statement of the problem will be simplified. Let there be a complex information system protected by some security system. The security system in the case of detection of suspicious activity informs about a threat that has arisen, its type and the assessment of threat reality probability. Suppose that there are several open ports in the system and let the web service be running on port 80 to handle client requests. By default, when a threat is detected on one of the ports, this port is blocked. As a result of port blocking, all services using this port must be stopped. If the web service is stopped, a critical error will occur and the company will incur serious costs. If a threat is not serious - you cannot admit the occurrence of critical errors as a result of the application of protective measures. Network worms belong to the class of not very dangerous network invasions. Suppose that the security system has detected suspicious activity on port 80, similar to the attack of a network worm.

In the formal language, this problem will take the following form, where *A1-A5* are the original arguments, *R1* is the undercutting rule, *R2* is the defeasible rule:

A1: *attack*(*port*_80, *warm*) – the suspicion of a network worm attack on port 80 is detected;
A2: *use*(*web_service*, *port*_80) – port 80 is used by the web *service*;
A3: ∀x ∀y (*block*(*x*) & *use*(*y*, *x*) → *stop*(*y*)) – when the port is blocked, all services using this port must be stopped;
A4: *stop*(*web_service*) → *critical_error* – stopping the web service is a critical error;
A5: ~ *serious_attack*(*warm*) – network worms belong to the class of not very dangerous network invasions;
R1:∀*y* ~ *serious_attack*(*y*) & *critical_error* ==>∀*x attack*(*x*,*y*) @ *block*(*x*) – the undercutting rule stating that if the threat is not very serious - it is impossible to admit the emergence of critical errors as a result of the application of protective measures;
R2: ∀*x*∀*y attack*(*x*,*y*)) |=> *block*(*x*) – the defeasible rule that states that by default when a threat is detected on one of the ports, this port is blocked.

Figure 1 shows the inference graph for the given problem. Let's analyze the solution of this problem in steps. Arguments *A1–A5* are given initially. The defeasible argument *A6: block (port_80)* is obtained using the argument *A1: attack (port_80, warm)* and the defeasible rule *R2*:∀*x*∀*y attack*(*x*,*y*)) |=> *block*(*x*). Arguments *A7* and *A8* are drawn by means

of the skolemization procedure from the argument *A5* (the variables bound by the universal quantifier are replaced by free variables, denoted by _x and _y, respectively). Argument A9: *block*(_*x*) → (*use*(_*y*, _*x*) → *stop*(_*y*)) is inferred from the argument A8: *block*(_*x*) & *use*(_*y*, _*x*) → *stop*(_*y*) using the rule of premise disconnection. The defeasible argument *A10: (use (_y, port_80) -> stop (_y))* is derived from the defeasible argument A6*: block (port_80)* and the argument *A9*: *block*(_*x*) → (*use*(_*y*, _*x*) → *stop*(_*y*)) by the Modus Ponens rule. The defeasible argument *A11: stop (web_service)* is inferred from the defeasible argument *A2: use (web_service, port_80)* and the argument *A10: (use (y, port_80) -> stop (y))* also by the Modus Ponens rule. The defeasible argument *A12: critical_error* is obtained from the defeasible argument A4: *stop*(*web_service*) → *critical_error* and the argument *A11: stop (web_service)* according to the Modus Ponens rule. Arguments *A12: critical_error* and *A5: ~ serious_attack (warm)* in accordance with the undercutting rule R1:∀*y* ~ *serious_attack*(*y*) & *critical_error* ==>∀*x attack*(*x*,*y*) @ *block*(*x*) undercut the defeasible inference between the arguments *A1: attack (Port_80, warm)* and *A6: block (port_80)*, making the *A6* argument defeated.

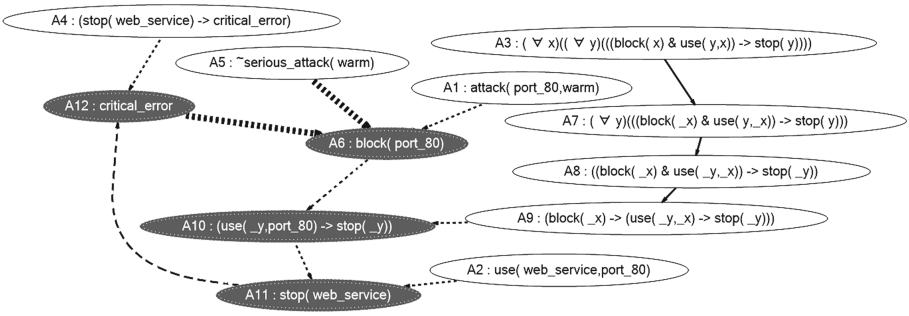

Fig. 1. The example of the argumentation application for decision making about port blocking

As a result of solving this problem by argumentation based on defeasible reasoning, the argument *A6: block (port_80)* became defeated, i.e. a possible attack of the network worm on port 80 is not so great to decide to block all traffic on this port because this will lead to significant costs.

3 The Problem of Detecting an Unauthorized Access Based on the Analysis of Network Traffic Using the Inductive Concept Formation

At present, using components of foreign production is the cornerstone of production of technical means and the software for the majority of computing systems. However, at the same time there is a threat of information leakage due to using the functional capabilities negatively affecting on safety of processed information. Such malicious functional possibilities could use for organizing hidden channels in round of well-known protection means at passing defended information on computer networks.

One of the ways of information losses being discussed from 1990s [7] is possibility of malicious codes to use parameters of transfer protocols for coding defendable hidden information. In case a protection system implements encoding the information transferred according to the IP protocol, a malicious code can use such parameters as a lengths of packets, temporal intervals between packets for coding transferred unauthorized information. Taking account of a high complexity of up-to-date software and closing its program codes for researching by developers, often it is impossible to produce software researches for detecting similar malicious program functions.

3.1 The Method of the Anomaly Detection

Transferring encoded data on communication links is brought about by the conversion of bit sequences to electromagnetic signals. The data presented by bits or bytes are transferred with a velocity defined by the number of bits in a time unit. Such parameters of the physical layer of network protocols as a bit rate, an encoding method, a transfer scheme and a signal range are defined by standards that are developed by the competent organizations. The process of physical data transfer on a certain interval can be considered as time series.

The analysis of a traffic will be a basis for detection of malicious program functions in an information exchange system: if parameters of a typical information exchange on certain protocol are known, then the abnormal template of behavior in a computer network obtained by traffic analysis in the case of exchange according to this protocol could speak that at analyzed system there is a malicious program function, and anomalies in a traffic are caused by actions of such programs.

In general, the time series ***ts*** is an ordered sequence of values

$$\boldsymbol{ts} = < ts_1, ts_2, \ldots, ts_i, \ldots, ts_q >,$$

describing a process flow, where the index i corresponds to a time label. ts_i values can be sensor indications, product prices, exchange rates and so on.

The anomaly detection problem [8] is set up as the task of searching for patterns in data sets that do not satisfy some typical behaviors. The anomaly or "outlier" is defined as an element that stands out from a data set belonging to it and differing significantly from the other elements of a sample.

Let's consider, thus, the problem of anomaly detection in sets of time series. The problem of anomaly detection in time series sets is formulated as follows. Let $TSStudy = <tsstudy_1, tsstudy_2, \ldots, tsstudy_m>$ be a set of objects where each object is time series. We call it a study set. Each of the time series in a study set represents some "normal" behavior or a process flow. Based on the analysis of *TSStudy* one needs to build a model to distinguish the instances of time series *TSStudy* from test sample sets:

$TStest = <tstest_1, tstest_2, \ldots, tstest_n>$ that could be "normal" or " abnormal".

We propose the method of the anomaly detection in the sets of time series, This method is a modification of an "exact exception problem" [9] that is described as follows: for the given set of objects I, one needs to get an exclusion-set I_x. To do this, there are introduced:

- the function of dissimilarity $D(I_j)$, $I_j \subseteq I$: defined on $P(I)$, where $P(I)$ is a set of all subsets I_j for I receiving positive real values;
- the cardinality function $C(I_j)$: $I_j \subseteq I$, defined on $P(I)$ and receiving positive real values such that for any two subsets $I_k \subseteq I$, $I_l \subseteq I$, $(k \neq l)$ $I_k \subset I_l \Rightarrow C(I_k) < C(I_l)$;
- the smoothing factor $SF(I_j) = C(I \backslash I_j) \cdot (D(I) - D(I \backslash I_j))$, that is calculated for each $I_j \subseteq I$.

Then $I_x \subset I$ will be considered as an exclusion-set for I with respect to $D(I)$, and $C(I)$, if its smoothing factor $SF(I_x)$ is maximal [9].

Informally, an exclusion-set is a smallest subset of I, that makes the largest contribution to its dissimilarity. A smoothing factor shows how much the dissimilarity of a set I can be reduced, if to exclude a subset I_j from it. A dissimilarity function can be any function that returns low values if elements of a set are similar to each other and higher values if elements are dissimilar. This method was adapted for the task of the anomaly detection in collections of time series. Let us introduce the following changes.

As a set of objects I we use $TSStudy \cup \{tstest_j\}$ for each $tstest_j \in TSTest$. The function of dissimilarity (I_j) for time series is defined as follows. Let $I_j \subseteq I$ be a subset of time series. Each time series is considered as a vector of its values. Further in shot, any individual time series is designated by $a \in I_j$. Calculate an average value on vector coordinates $\overline{I_j}$. Then the dissimilarity function for time series is computed as a sum of squared distances between $\overline{I_j}$ and vectors $a \in I_j$:

$$D(I_j) = \frac{1}{N} \cdot \sum_{a \in I_j} \left| a - \overline{I_j} \right|^2 \text{ where } \overline{I_j} = \sum_{a \in I_j} \frac{a}{\left| I_j \right|} \text{ and } N - \text{the number of elements } I_j.$$

The cardinality function is given by the formula $C(I \backslash I_j) = 1/|I_j| + 1$. The formula for calculating the smoothing factor is $SF(I_j) = C(I \backslash I_j) \cdot (D(I) - D(I_j))$.

If an exclusion set for $I = TSStudy \cup \{tstest_j\}$ contains $tstest_j$, then $tstest_j$ is an anomaly. For anomaly detection in collections of time series two algorithms were developed [10]: TS-ADEEP for study sets, containing a single class of time series and TS-ADEEP-Multi for study sets containing several classes of time series.

The generalization of the TS-ADEEP-Multi algorithm is quite obvious: splitting a study set to a single class subsets and consequently applying the TS-ADEEP algorithm, we can determine whether the considered time series is an anomaly or not. If time series is an anomaly for each subset, then time series is an anomaly for the whole study set [11]. The results of the software modelling described in this paper were obtained on the original dataset "TRAFFIC". This dataset contains examples of time series on the bases of which the detection task of cases of information exchange with unauthorized data access has been solved.

3.2 Experimental Results

The possibility of applying algorithms for anomaly searching in collections of time series to the problem of detection of cases of atypical information exchange on a network that assumes existence of malicious codes has been researched. This problem was complicated because it is very hard to receive a representative sample which would be rather

exact and at the same time exactly describing all possible variants of behavior in an information system. Also it is necessary to note that obtaining a sample for the normal behavior of an information systems is enough easily than for abnormal one because the normal behavior could be modeled in laboratory conditions while abnormal behavior happens extremely rare. Moreover, the abnormal behavior is dynamic by the nature, and therefore there can be new types of anomalies which weren't represented in the original study samples.

It is proposed the following problem solution. There are reference models presented by time series which reflect parameter changes of the protocol depending on types of information exchange. For comparing, the time series are used, that present the real behavior of an information system in the case of a data interchange. The comparison of these two models of information exchange allows to look for behavior types in the case of information exchange, different from standard, i.e. the anomalies. As an illustration of the method, the exchange protocol by the FTP files was chosen. On the basis of analysis of a network traffic under transferring files in accordance with the FTP protocol in various conditions (including simultaneous transferring several files), the data set was obtained that represents a study sample for creating of a model of reference data transferring. Experiment details are given in [12]. The following variants of data transferring were researched:

- transmission according to the FTP protocol (standard);
- simultaneous transmission according to the FTP protocols and ping (FTP-traffic was analyzed);
- simultaneous transmission according to the FTP protocols and UDP (FTP- traffic was analyzed).

The example of normal data exchange running using the FTP and UDP protocols is given in Fig. 2. Having this information about data transferring on a network, it is necessary to define whether data transferring is "suspicious", what could testify about a possible compromise of network infrastructure, and malicious code existence. As test data, among others, there were used specially generated time series simulating the

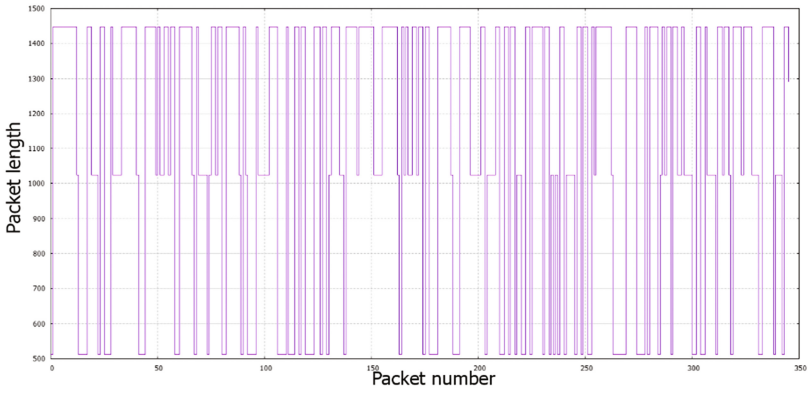

Fig. 2. Time series representing the normal flow of information exchange

transfer of unauthorized data. Figure 3 shows time series having the irregular structure and presenting the anomaly. Such time series, having a pronounced irregular structure, are anomalies. The original "TRAFFIC" data set was constructed by combining normal and abnormal time series that were used in the experiments further.

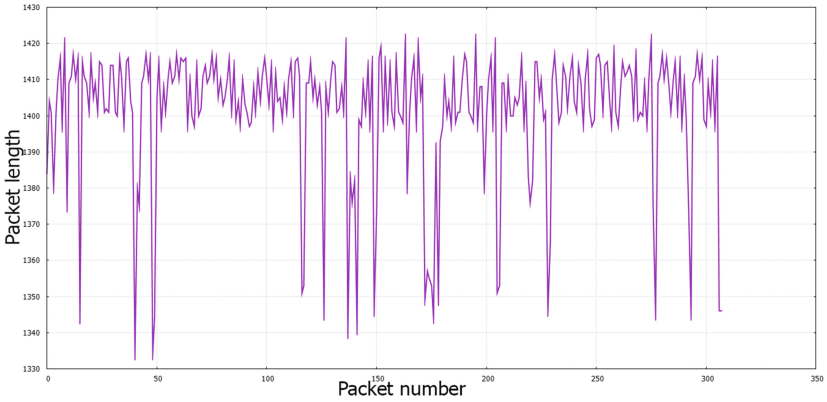

Fig. 3. Time series representing the abnormal flow of information exchange

To analyze one type of traffic (FTP protocol, FTP and ping protocols, FTP and UDP protocols), the TS-ADEEP algorithm proposed in [10] was used. The obtained experimentally data were viewed as a set of time series where ts_i values represent lengths of packages. These data used in the experiment were subjected to a preliminary treatment consisting of two stages: normalization and a subsequent discretization of the normalized time series with a transition to symbolical data presentation, and where an alphabet size (the number of used symbols) has been varied depending on a task. The process of preliminary data conversion is based on the ideas of the SAX algorithm [13]. Changing parameters - an alphabet size and a time series dimensionality - one can obtain an optimal representation of time series for their using by TS-ADEEP and TS-ADEEP-Multy algorithms.

Results of recognizing the anomalies in the case of data transferring according to the above-named protocols are given in Table 1. As it is seen from results, for the given task it was succeeded to reach the accuracy of anomaly classification up to **100%**.

By the simultaneous analysis of all traffic types considered in the experiment, the TS-ADEEP-Multy algorithm was used. Here the problem becomes complicated because the normal behavior can correspond to one of several classes. Presented in the Table 2 results show, that for the case of several classes it is possible to reach the classification accuracy of anomalies up to 100% at selection of parameters of time series normalization.

Table 1. The accuracy of the anomaly detection in data sets “Traffic” with the single class for the TS-ADEEP algorithm

		Dimensionality of time series						
		210	150	100	50	30	20	10
Number of alphabet symbols	5	71,43	82,14	60,71	64,29	60,71	46,43	67,86
	10	92,86	96,43	100	96,43	82,14	85,71	64,29
	15	92,86	100	100	96,43	92,86	82,14	85,71
	20	92,86	100	100	96,43	92,86	82,14	82,14
	25	92,86	100	100	96,43	92,86	82,14	92,86
	30	92,86	100	100	96,43	92,86	92,86	82,14
	40	92,86	100	100	96,43	92,86	92,86	92,86
	50	92,86	100	100	96,43	92,86	92,86	92,86

Table 2. The accuracy of the anomaly detection in data sets “Traffic” with several classes for the TS-ADEEP-Multy algorithm

		Dimensionality of time series						
		210	150	100	50	30	20	10
Number of alphabet symbols	5	85,71	89,29	57,14	64,29	60,71	46,43	67,86
	10	96,43	96,43	100	96,43	96,43	85,71	67,86
	15	92,86	100	100	96,43	92,85	82,14	67,86
	20	96,43	100	100	96,43	96,43	96,43	75,00
	25	96,43	100	100	96,43	96,43	82,14	82,14
	30	96,43	100	100	96,43	96,43	96,43	92,86
	40	96,43	100	100	96,43	96,43	96,43	92,86
	50	96,43	100	100	96,43	96,43	96,43	96,43

4 Conclusion

The proposed idea of using argumentation in network security systems, according to the authors, will make such systems more flexible and will make it possible to assess the appropriateness of using certain protective mechanisms. The use of argumentation will allow giving qualitative and quantitative assessment of the recommendations developed by the system, thereby solving an important task - the task of choosing the way to react to suspicious activity in the system.

The research also examined methods for detecting anomalies when solving the task of analyzing network traffic in order to detect malware. The algorithms implementing the search for anomalies were software modelled. The results of experiments showed high accuracy of detection of anomalies, what indicates good prospects for using the proposed methods and software.

Acknowledgment. This paper was prepared under the financial support of the RFBR Grant No. 17-07-00442, 15-01-05567 and Russian Federation Presidents Grant for Young Scientists MK-2897.2017.9.

References

1. Ou, X., Govindavajhala, S., Appel, A.W.: MulVAL: a logic-based network security analyzer. In: USENIX Security (2005)
2. Eronen, P., Zitting, J.: An expert system for analyzing firewall rules. In: Proceedings of the 6th Nordic Workshop on Secure IT Systems, pp. 100–107 (2001)
3. Khosravifar, B., Bentahar, J.: An experience improving intrusion detection systems false alarm ratio by using honeypot. In: 22-nd International Conference on Advanced Information Networking and Applications, AINA 2008, pp. 997–1004 (2008)
4. Bandara, A., Kakas, A., Lupu, E., Russo, A.: Using argumentation logic for firewall policy specification and analysis. Lecture Notes in Computer Science, pp. 185–196 (2006)
5. Pollock, J.: How to reason defeasibly. Artif. Intell. **57**, 1–42 (1992)
6. Vagin, V., Morosin, O., Fomina, M.: Inductive inference and argumentation methods in modern intelligent decision support systems. J. Comput. Syst. Sci. Int. **55**, 79–95 (2016)
7. Covert channel. https://en.wikipedia.org/wiki/Covert_channel
8. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: a survey. ACM Comput. Surv. **41**, 1–58 (2009)
9. Arning, A., Agrawal, R., Raghavan, P.: A linear method for deviation detection in large databases. In: Proceedings of KDD 1996, pp. 164–169 (1996)
10. Antipov, S., Fomina, M.: Problem of anomalies detection in time series sets. J. Prog. Prod. Syst. **2**, 78–82 (2012). (in Russian)
11. Fomina, M., Antipov, S., Vagin, V.: Methods and algorithms of anomaly searching in collections of time series. In: Proceedings of the First International Scientific Conference "Intelligent Information Technologies for Industry" (IITI 2016), pp. 63–73 (2016)
12. Antipov, S.G., Vagin, V.N., Fomina, M.V.: Detection of data anomalies at network traffic analysis. In: Open Semantic Technologies for Intelligent Systems - Conference Proceedings Minsk, Belarus, pp. 195–198 (2017)
13. Lin, J., Keogh, E., Lonardi, S., Chiu, B.: A symbolic representation of time series, with implications for streaming algorithms. In: Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery - DMKD 2003, pp. 2–11 (2003)

http://www.springer.com/978-3-319-68320-1

Proceedings of the Second International Scientific Conference "Intelligent Information Technologies for Industry" (IITI'17)
Volume 1
Abraham, A.; Kovalev, S.; Tarassov, V.; Snasel, V.; Vasileva, M.; Sukhanov, A. (Eds.)
2018, XVI, 547 p. 194 illus., Softcover
ISBN: 978-3-319-68320-1