30. Fingerprint matching using minutiae and texture features // Proceedings 2001 International Conference on Image Processing (Cat. No.01CH37205). 2002. doi: https://doi.org/10.1109/icip.2001.958106
31. Impact of artificial «gummy» fingers on fingerprint systems / Matsumoto T., Matsumoto H., Yamada K., Hoshino S. // Optical Security and Counterfeit Deterrence Techniques IV. 2002. doi: https://doi.org/10.1117/12.462719
32. Model of stegosystem images on the basis of pseudonoise codes / Riznik O., Yurchak I., Vdovenko E., Korchagina A. // In Perspective Technologies and Methods in MEMS Design (MEMSTECH), 2010 Proceedings of VIth International Conference. 2010.
33. Fries M., Fischbach R., Houdeau D. U.S. Pat. No. 6.347.040. Washington, DC: U.S. Patent and Trademark Office, 2002.

Запропоновано метод виявлення шахрайства при інсталюванні мобільних додатків. Розроблений метод на відміну від існуючих використовує всі наявні дані, незалежно від типів, розмірності і розбіжності цих даних та перетворює такі дані до однорідних коефіцієнтів на основі запропонованого методу шкалювання. Такий підхід дозволяє підвищити точність розв'язання задачі та побудувати відкриту до розширення базу знань з характеристиками шахраїв та правилами виявлення користувачів-шахраїв. Розроблена система шкал для переводу різнорідних даних до однорідних коефіцієнтів, яка дозволила побудувати математичну модель процесу шкалювання. Розроблено алгоритм шкалювання різнорідних масивів даних на основі запропонованих шкал та математичної моделі процесу шкалювання великих масивів різнорідних даних, що дозволило всю множину даних привести до двох однорідних груп. Запропоновано алгоритми обробки отриманих груп однорідних даних та виявлення користувачів-шахраїв. Розроблені алгоритми з використанням коефіцієнтів схожості між характеристиками користувачів формують шаблони шахраїв, визначають характеристики та залежності користувачів-шахраїв, що дозволяє підвищити ефективність та швидкість процесу виявлення шахраїв. Була запропонована схема процесу виявлення шахраїв, що використана в інтелектуальній системі автоматичного виявлення шахраїв для проведення експериментальних досліджень. За результатами експериментальних досліджень отримана точність визначення шахраїв на заданій репрезентативній вибірці 99,14 %. Результати експериментальних досліджень показали ефективність автоматичного виявлення шахраїв та можливість розширення форматів та характеристик користувачів-шахраїв на основі інтелектуального аналізу і баз знань

Ключові слова: виявлення шахрайства, різнорідні дані, інсталювання мобільних додатків, аномалії в даних, шкалювання даних

UDC 004.8:044.89
DOI: 10.15587/1729-4061.2019.155060

DEVELOPMENT OF A METHOD FOR FRAUD DETECTION IN HETEROGENEOUS DATA DURING INSTALLATION OF MOBILE APPLICATIONS

T. Polhul
Postgraduate student*
E-mail: tanapolg93@gmail.com
A. Yarovyi
Doctor of Technical Sciences,
Professor, Head of Department*
E-mail: a.yarovyy@vntu.edu.ua
*Department of Computer Science
Vinnytsia National Technical University
Khmelnytske shose str., 95,
Vinnytsia, Ukraine, 21021

1. Introduction

Promotion of mobile software is typical for the present-day IT market. For such activity, companies should spend a lot of money on marketing campaigns. One of the options of determining effectiveness of a marketing campaign consists in checking the number of mobile app installations provided by the company. It is worth to know at this step that a certain part or entire set of mobile app installations could be performed in a fraudulent way. Knowing the actual number of organic mobile app installations and the number of fraudulent installations, one can determine real cost of a marketing campaign and whether it is effective. Note that fraudulent users are called fraudsters. Therefore, development of a system for automatic detection of fraudsters and marketing campaigns using fraudulent methods of installation is a relevant task in this field.

2. Literature review and problem statement

Complexity of the problem that arises in this study consists first and foremost in uncertainty of the "fraud" concept in technical literature. For example, fraud is considered in [1] as a some sort of anomalies in data. In its turn, anomaly can be defined as contextual (conditional) anomaly "if a copy of data is abnormal in a particular context." Methodology for detecting this type of anomaly "takes into account

difference between the user-specified environmental and indicator features during the anomaly detection process" [2]. Also, the authors mention point and collective anomalies. Anomalies are considered to be point "if a certain copy of data can be regarded as abnormal with respect to other data" [3]. Anomalies are considered collective "if the set of corresponding data copies is abnormal in relation to the whole set of data". Therefore, when identifying collective anomalies, it is necessary to look for "the elements inside a group that are more relevant to each other than to the external elements" [4]. However, such definitions do not allow mathematical description of fraud, link it to the data sets that are carriers of fraud and automate the process of its search in large data sets. When solving the problem, fraud will be considered as an anomaly, namely, as a deliberate generation of anomaly in the data on the process being studied by a third party (fraudster) or a mechanism for a particular purpose.

Secondly, when searching for fraud in data during installation of mobile applications, it is necessary to clearly identify types and formats of data in which search is made because choice and analysis of processing methods depend on them. Let us consider the groups of input data used when installing mobile apps. These include:

– numerical data, for example, continuous data: action time, or discrete data (number of friends in social networks, number of purchases, etc.);

– qualitative data, for example, categorical data: the name of the mobile platform of the user (iOS, Android, etc.), user IP, user device ID, user information, user's social network etc., or dichotomous data: connected to social nets or not, installation confirmed or not, purchase confirmed or not, etc.;

– plural data: sets of numerical or qualitative data, for example, a set of time of each event, a set of types of user events, etc.

As it can be seen, in fraud detection systems when installing mobile apps, there are input data both numerical and qualitative as well as sets of numerical and qualitative data. Besides, data are not compared with each other, they have different dimensions and accept values from different ranges. That is, data are heterogeneous. It should be noted that the problem of data heterogeneity occurs when detecting anomalies in different areas. To overcome it, methods for reducing dimension of the input data vector described in [5 8] or such method of overcoming heterogeneity as one-hot encoding which is considered, e.g. in [5] are often used. The first of these methods introduces uncertainty in data since it can discard data important for correct decision making and a possibility of further result substantiation based on the initial input data. The second of these methods can convert qualitative data in numerical ones only if the final set of categories of a certain qualitative feature is known. However, e. g. in the case of IP address, the set of all possible values (categories) is unknown in advance, so the second method does not work with such data.

Therefore, in the course of studying the methods of detecting anomalies, their sources were analyzed [1, 5–25] and machine learning methods for Big Data processing were distinguished. They can be divided as follows:

– classification methods: many of them were mentioned, e. g. in [5]. Among them, expert systems are distinguished. They are considered, e. g. in [9] and used in detection of anomalies in the medical area, in credit cards, in image processing, when detecting network intrusions. However, drawback of expert systems consists in the fact that when new fraudulent patterns appear, user has to track them and add new rules to the system. Bayesian net is considered separately. It is used to detect anomalies in medical data, image processing, sensor nets. Bayesian net is mentioned, e. g. in [6, 10–12]. However, heterogeneous data are not used in this method because of peculiarity of net construction. There are also other methods such as Support Vector Machine, the method of k-nearest neighbors, classification methods based on neural networks. According to [1], the latter method is most often used in detecting credit card fraud, in image processing or in detecting network intrusions. But as shown in the sources discussed, these techniques are effective when homogeneous data are used;

– clustering methods which are divided into hierarchical (taxonomy) and non-hierarchical or exact and fuzzy methods. Among the methods of this group, k-means clustering method, graph methods including, e. g. an algorithm for selecting connected components can be mentioned. FOREL algorithm and agglomerative hierarchical clustering can also be mentioned. Most of these methods are discussed in [7] and applied in a professional information and analytical resource [13]. It should also be noted that one of the clustering methods is based on the use of similarity coefficients as is shown, e. g. in the authors' studies [14–18] and in [19]. Clustering methods are most often used in network intrusion detection discussed in [20] and in the case of credit card fraud detection. However, such methods feature use of only homogeneous data;

– statistical methods are considered, e. g. in [21, 22]. In this group of methods, spectral method that is most often used in mobile phone anomaly detection and in detection of anomalies in sensor networks can be mentioned. Also, non-parametric statistical modeling which is used in detecting net intrusions and in trouble shooting in mechanisms can be mentioned. Another method that belongs to this group is parametric statistical modeling which is considered in [23, 24]. Statistical profiling using histograms is discussed in [25]. It also refers to this group of methods. However, when using these methods, there are no procedures for reducing data to homogeneous data.

In particular, clustering methods are not suitable for solving the problem of fraud detection when installing mobile apps since these methods are learning methods without a teacher, that is, the methods which identify groups of similar users on their own. However, in the problem under consideration, it is necessary to determine in advance classes in which users should be grouped.

Also, it should be noted that all of the above methods work with homogeneous data, so the task of developing a method for overcoming heterogeneity of input data in mobile applications remains relevant. Therefore, it is expedient to develop a method for detecting frauds as anomalies occurring in installation of mobile applications which would enable analysis of heterogeneous data to detect anomalies in these data in contrast to the abovementioned methods.

3. The aim and objectives of the study

The study objective was to study the possibility of overcoming data heterogeneity to enable automatic fraud detection during installation of mobile applications without losing accuracy of the results obtained and the possibility of substantiation of obtained results.

To achieve the objective, the following tasks should be solved:

– to classify heterogeneous data when installing mobile applications which would enable further development of the method of fraud detection during installation of mobile applications;

– to develop a method for fraud detection in heterogeneous data arrays when installing mobile applications. In contrast to existing methods, it should enable defining of complete fraudster formats by means of intelligent data analysis and the proposed model, scales and scaling algorithms to materially improve effectiveness of the procedure for detecting new fraudsters;

– to analyze results obtained in the study of the proposed model, algorithms and the method of fraud detection in heterogeneous data and evaluate accuracy of fraudster detection in a given representative sample of users.

4. Classification of heterogeneous data when installing mobile applications

When choosing a method necessary for solving the problem from a multitude of considered methods, it is important to analyze input data with which the considered methods would work and the known data for achievement of the objective.

In order to classify data and determine the set of input data necessary for deciding on the presence of fraudulent situation, it is necessary to consider in detail the whole process of emergence and change of data. In the process of installing a mobile application, behavior of each user taking part in this process is characterized by a set of events, for example: installation and confirmation of installation, opening of the application and its registration, etc. sequential set of possible input data is presented in Fig. 1.

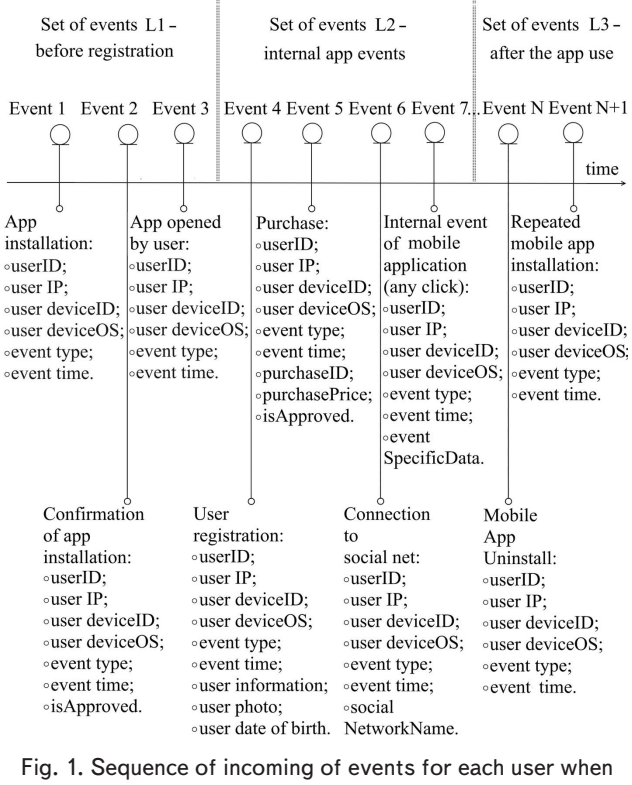

Fig. 1. Sequence of incoming of events for each user when installing mobile applications

Sequence of occurrence of events for each user divided into $L1$, $L2$, $L3$ sets depending on the event type is shown in Fig. 1. Note that common features of each event are as follows:

– information about the current user, namely unique identifier (*userID*);

– information about the current device, namely its unique identifier (*user deviceID*), IP address (*user IP*), information on the device operating system (*user deviceOS*);

– information on the current event, namely the event type and time.

Also, each of the events may contain features belonging solely to it, for example:

– every internal event of mobile application contains a tuple of information specific to it (eventSpecificData);

– the event of confirmation of app installation has an isApproved check box indicating whether the installation is confirmed or not;

– the user registration event contains user information, photo and date of birth;

– the purchase event contains a unique identifier for the purchased product (offerID) and the purchase proper (purchaseID), price of the purchase (purchasePrice) and an isApproved check box for purchase confirmation or non-confirmation.

In the process of this study, the whole set of input data required for fraud detection when installing mobile applications was divided into several subsets:

– M1 set: the user data and information during installation;

– M2 set: the user information and his actions after installation;

– M3 set: the user data during the uninstall process.

However, determination of amount of most important input data and characteristics of these data has appeared to be a rather difficult task in this study process. Therefore, an expert poll was carried out to identify a complete array of input data and characteristics of these data in which 25 experts experienced in fraud detection from leading IT companies in Ukraine, Switzerland and the USA have taken part.

The expert poll was aimed at determining the set of input data and data characteristics that can be used to determine whether a user is a fraudster or not. For this purpose, poll was conducted in two stages. At the first stage, experts were provided with a set of all possible input data to be ranked or supplemented with other input data. At the second stage, experts determined limit values for the defined set of input data. For example, the following was established:

– limit values for the number of clicks from one IP address:

$$Tip\_act\_min \text{ and } Tip\_act\_max;$$

– limit values for time between the events of app installation and the user registration:

$$Tinst\_min \text{ and } Tinst\_max;$$

– minimum and maximum number of friends in social nets:

$$Cf\_min \text{ and } Cf\_max.$$

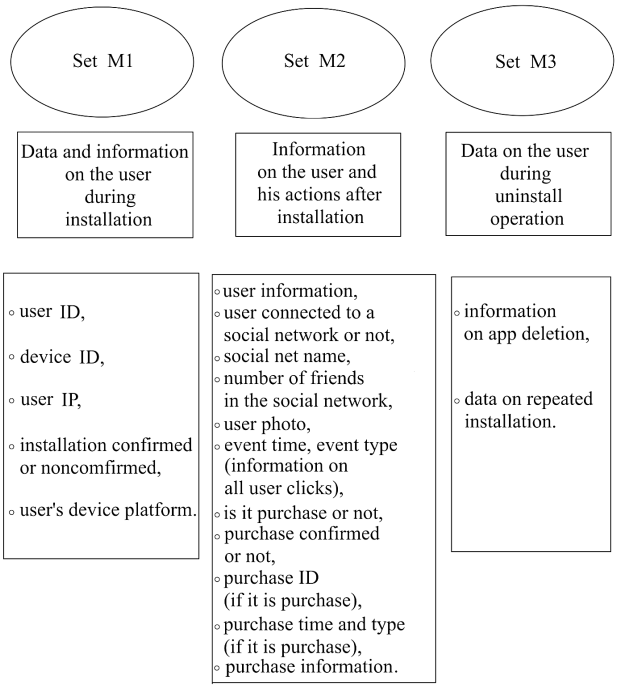

Fig. 2. Input data divided into three distinct sets

The results obtained at the first stage of the expert poll are presented in Fig. 2. As seen from the input data established by the expert poll, the data are heterogeneous. For example, input data contain both qualitative and numerical indicators and take values from different ranges.

The results obtained at the second stage of the expert poll are presented using the set theory, namely in the component form:

$$O=(D,X), \tag{1}$$

where D is the set of events for each user; X is the data characteristics defined by the experts and presented in the form of properties of the data set D. These properties should be inherent to all organic users. Let us consider these properties:

$$P_1(IP)=IP\notin FRAUD\_IP, \tag{2}$$

where $FRAUD\_IP$ is the set of known fraudulent IP addresses, IP is the IP address that the user used. This property (2) will check whether the IP address is fraudulent or not. So, if one looks at this property from the opposite side, then it can be noted that if $IP\in FRAUD\_IP$, then the user having this IP address is definitely a fraudster;

$$P_2(ID)=ID\notin FRAUD\_ID, \tag{3}$$

where $FRAUD\_ID$ is the set of identifiers of known fraudsters, ID is the unique user identifier. Similar to the previous property, it can be noted that if $ID\in FRAUD\_ID$, then the given user is a fraudster;

$$P_2(D\_ID)=D\_ID\notin FRAUD\_D\_ID, \tag{4}$$

where $FRAUD\_D\_ID$ is the set of identifiers of devices on which fraudsters were spotted; D_ID is the unique device (mobile phone) identifier. Similar to the previous properties, it can be noted that if $D\_ID\in FRAUD\_D\_ID$, then the given user is a fraudster;

$$P_3(P_{id})=(P_{id}.isAproved=true), \tag{5}$$

where P_{id} is the unique identifier of the purchase made by the user, the $isAproved$ check box indicates whether the purchase was confirmed by corresponding store or not. It can be noted that if the user has at least one unconfirmed purchase, then the given user is definitely a fraudster;

$$P_4(T)=T\in AVAILABLE\_TYPES, \tag{6}$$

where $AVAILABLE\_TYPES$ is the set of event types available to the user at this stage, T is the type of the current user event. Similar to the previous properties, it can be noted that if $T\notin AVAILABLE$, then the given user is a fraudster;

$$P_4(D\_A_{cnt})=D\_A_{cnt}<=5, \tag{7}$$

where D_A_{cnt} is the number of accounts on one device. If this property is not met, then users who use the device with the current identifier $deviceID$, are marked as fraudsters;

$$P_4(IP\_A_{cnt})=IP\_A_{cnt}<=5, \tag{8}$$

where IP_A_{cnt} is the number of accounts from one IP address. Similar to the previous property, if this condition is not met, then the users with this IP address are considered fraudsters.

To describe further properties, select from the D set the subsets of events for each of the user's devices (by $deviceID$) $E_1,E_2,\ldots,E_d$, where d is the number of devices from which the user used the mobile application, i. e.

$$E_1\cup E_2\cup\ldots\cup E_d=E.$$

It is worth noting that $E_1\subseteq E, E_2\subseteq E,\ldots, E_d\subseteq E$ and

$$E_1\cap E_2\cap\ldots\cap E_d=\varnothing.$$

In turn, each of the selected subsets can be divided into subsets of user actions every minute, namely:

$$\begin{aligned}&E_{1,1}\cup E_{1,2}\cup\ldots\cup E_{1,m1}=E_1, E_{1,1}\cap E_{1,2}\cap\ldots\cap E_{1,m1}=\varnothing;\\&E_{2,1}\cup E_{2,2}\cup\ldots\cup E_{2,m2}=E_2, E_{2,1}\cap E_{2,2}\cap\ldots\cap E_{2,m2}=\varnothing;\\&\ldots,\\&E_{d,1}\cup E_{d,2}\cup\ldots\cup E_{d,md}=E_d, E_{d,1}\cap E_{d,2}\cap\ldots\cap E_{d,md}=\varnothing,\end{aligned} \tag{9}$$

where $m1,m2,\ldots,md$ is the number of minutes spent in the application from the 1, 2, …, d device, respectively.

Then each of the $E_{1,1},\ldots, E_{1,m1}, E_{2,1},\ldots, E_{2,m2}, E_{d,1},\ldots, E_{d,md}$ subsets can be specified as follows:

$$\{e\,|\,P_{e1}(c),P_{e1}(c)\}=c<=50, \tag{10}$$

where c is the number of elements in this subset. This property indicates that an organic user can do no more than 50 clicks (events) per minute. If the number of events is much larger, then the user can be considered suspicious. In order to assuredly determine if the user is a fraudster judging by

the time of his events, it is necessary to check the following properties. To this end, represent the $E_1, E_2, \ldots, E_d$ subsets in the following form:

$$\{d \mid P_{d1}(t_{di}, t_{di+1}), P_{d2}(t), P_{d3}(n), P_{d4}(t_{din}, t_{do})\}, \quad (11)$$

$$P_{d1}(t_{di}, t_{di+1}) = t_{di+1} - t_{di} >= 60000 \text{ ms},$$

where t_{di}, t_{di+1} is the time between neighbor events. This property checks time between events. If the time is greater than the specified one, the user is considered suspicious and further properties are checked;

$P_{d2}(t) = E_i \notin F$, where t is the time of events $i \in [1, \ldots, d]$, F is the set of completely known distribution laws. This property checks affiliation of the set of the user event time to some well-known distribution law. This is necessary because of the fact that fraudulent scripts usually use random functions to select time between events. However, any random function is constructed on the basis of a certain known distribution law, usually a normal distribution law;

$P_{d3}(n) = E_i \notin F$, where n is the type (name) of events. This property is similar to the previous one but checks distribution of the event types. Necessity of this check is also explained by the fact that fraudulent scripts that perform as many as billions of installations per day/hour choose the type of the event in a way a "user" (i. e. script) does it with the use of a random function based on a specific distribution law;

$$P_{d4}(t_{din}, t_{do}) = t_{din} - t_{do} >= 120000 \text{ ms},$$

where $t_{din} - t_{do}$ is the time between adjacent events.

Similar properties will have subsets divided not according to *deviceID*, but according to *userIP*.

Certainly, main data characteristics were established based on the second stage of the expert poll, however, new fraudulent methods for installing applications may appear in each particular period of time. This is why data mining will also be applied in the method development. In fact, according to Geron Aurelien's definition [5], data mining is application of machine learning techniques to study large amounts of data (Big Data) for further detection of the fingerprints that were not immediately spotted.

5. Method of fraud detection in heterogeneous data arrays

Thus, to analyze such data, it is necessary to apply a method that works with heterogeneous data. However, if we consider one of the most popular current approaches to solving similar problems, i. e. neural and similar to neural networks [26], these approaches work only with numerical data, too. There is a way of converting categorical data to numerical (one-hot encoding [5]) but it is necessary to know the whole set of possible values of the categorical feature to apply this approach. However, it is impossible to know in advance all possible variants of such a feature as an IP address. Even types of events will be constantly added in the process of improvement of a mobile application which will lead to ever-growing number of input features and their data and continuous retraining of the system. In the case of IP addresses, there will be a very rapid growth of incoming categories which will require huge storage resources for each user. In general, most current algorithms practically do not support work with categorical data.

It should be noted that one more significant problem for the method being developed in this study is inability to use neural networks to overcome heterogeneous data. The reason is that it is impossible today to clearly substantiate decisions made by neural networks. It is important in consideration of litigations in which clear arguments have to be provided. For example, this is one of the most important reasons why unmanned cars are not yet available for sale: because such substantiation is obligatory in the studied area.

There is also a method that works both with numerical and qualitative data: a generalized discriminant analysis [10, 27]. However, this method also requires a set of possible values of a categorical variable similar to one-hot encoding. In our case, not all categorical input data have a discrete set of categories.

If we consider other existing methods of data normalization, they also do not work with numerical and qualitative data and there may be worsening of accuracy because of information loss and the mistakes made. Among the methods discussed, we should mention the methods of converting heterogeneous data to homogeneous (for example, multidimensional scaling) or methods for identifying the most important features (for example, principal component analysis [7]).

Most modern methods work with homogeneous data, so the need to develop a method that uses heterogeneous data obtained in the process of fraud detection is an important task.

Heterogeneity in most of the methods under consideration and in this study is understood as data of various value types (numerical and qualitative data) found in different ranges and dimensions that cannot be equivalently compared with each other. It should also be noted that the vector of input data can contain not only numerical or qualitative data but also arrays of numerical or qualitative data.

To overcome data heterogeneity, the authors used scaling (normalization) of data which means converting all data (qualitative and numerical) to a single scale from 0 to 1. We will assume that the zero value in the scale will mean that the user is a fraudster by this feature, and the value of 1 will mean that the user is organic. This is a feature of the proposed scale. For example:

– IP address (qualitative data whose entire set of values is unknown) is converted into a coefficient 0 or 1. This coefficient is determined in the following way: if the IP address belongs to the set of fraudulent IP addresses specified by the experts and is supplemented by the developed system in the process of its study, then the coefficient is equal to 0. If not, then it is equal to 1 (Fig. 3);

– event type: qualitative data in which, unlike the previous feature, all types of events are known in advance. However, when adding new types of events and using existing methods to overcome heterogeneity, it is necessary to retrain the entire system by adding new categories first. There are several properties of the set of input data with this feature. For example, if there are event types inaccessible for the user (for example, some functions of mobile applications can only be available after their registration) among the types of actions performed by the user, then the coefficient takes zero value;

– presence of an unconfirmed purchase is a qualitative feature, i.e. dichotomous. In the case of an unconfirmed purchase, this coefficient is 0 which indicates that the user is a fraudster. Otherwise, this coefficient is equal to 1;

– the set of the user event time is converted to a value from 0 to 1 where 0 means that the set belongs entirely to a particular distribution law, and 1 means the opposite (Fig. 4);

– the number of friends in the social network: it is numerical data which is also converted into a coefficient with value from 0 to 1. If the number of friends is within the limit values defined at the second stage of expert evaluation, then the coefficient value is 1, if otherwise, 0. Example of numerical feature is given in Fig. 5.

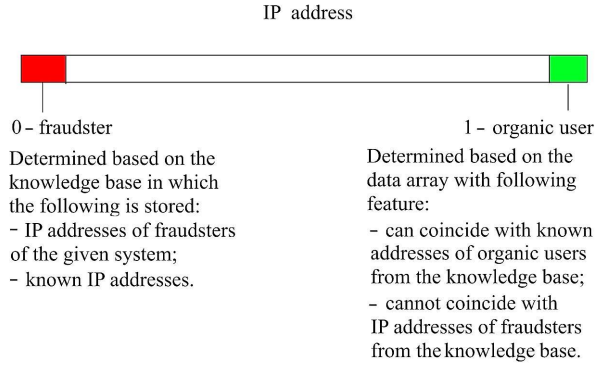

Fig. 3. The scale of defining the user class by a feature of IP address

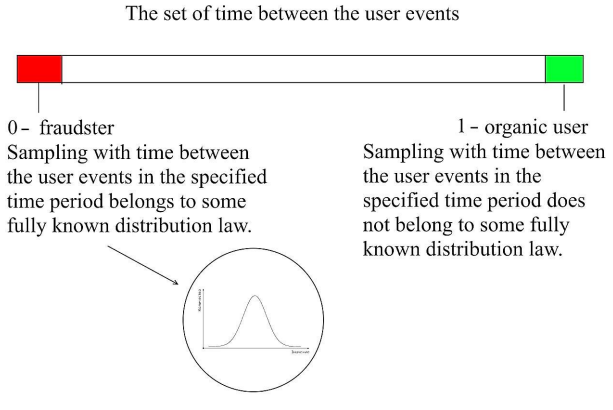

Fig. 4. The scale of defining the user class by feature 'set of time between the user events'

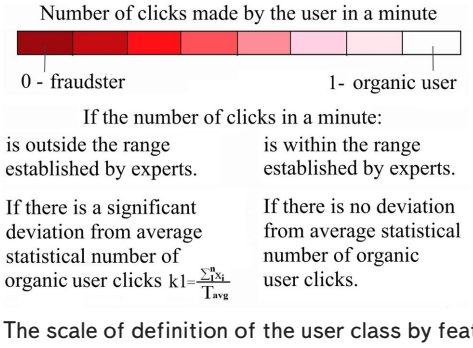

Fig. 5. The scale of definition of the user class by feature of number of clicks in a minute

As can be seen from the examples shown in Fig. 3–5, transition from heterogeneous to homogeneous data presented in the form of coefficients, was performed. Such a conversion of heterogeneous data into a homogeneous subsequently enables the use of classification methods to detect a fraudster.

In the process of scaling and on the basis of expert evaluation, authors have identified 17 coefficients that influence decision making in fraud detection. Analysis of these coefficients has made it possible to group the coefficients as follows:

– the first group covers the coefficients enabling preliminary analysis of data, namely uniquely identify fraudulent, organic and suspicious users in the set of users. It enables initial sampling based on the decision tree. For example, there are 13 coefficients in the first group. Let us consider some of them:

k_2 (purchased/not purchased): if the number of purchases made by the user $\geq K_{2\_min}$, then k_2 is tentatively equal to 1 and means that the user is not a fraudster. If the number of purchases $< K_{2\_min}$, then k_2 is tentatively equal to 0.5 which means that the user is suspicious. Coefficient K_{2_min} was determined in this study on the basis of expert poll. If a user has made a purchase that is not available to him, then k_2 is tentatively equal to 0 and means that the user is a fraudster;

k_3 (confirmed/nonconfirmed purchase): if a user has made an attempt to send a non-existent identifier of purchase that was not confirmed by the store of the corresponding mobile platform, then this user can automatically be considered a fraudster;

k'_4 (time between user events): if a sample with time between user events belongs to some well-known distribution law, then that user will be identified as a fraudster. Affiliation of the sample to the distribution law can be determined using the Kolmogorov-Smirnov criterion [28];

k_5, k_6 (user event type, frequency of each user event type): interrelated indicators. When $k_5 \in \left[K_{5\_min\_cnt}; K_{5\_max\_cnt}\right]$ a user is definitely considered a fraudster. $K_{5\_min\_cnt}$ and $K_{5\_max\_cnt}$ are determined on the basis of an expert poll;

k_8 (number of friends in a social net): if a user is connected to a social net and the number of his friends in this net belongs to $\left[K_{8\_min} * K_{8\_opt}; K_{8\_max} * K_{8\_opt}\right]$ and interval k_8 in this case equals to $\left[K_{8\_min}; K_{8\_max}\right]$, the given user is a fraudster. Coefficients K_{8_min}, K_{8_opt}, K_{8_max} are determined on the basis of an expert poll;

– the second group covers coefficients that do not enable a preliminary analysis based on the decision tree. However, based on the samples of fraudsters, organic and suspicious users formed on the basis of coefficients from the first group, one can find coefficients of similarity of all users with coefficients found according to each of characteristics of the second group. For example, 6 coefficients belong to the second group (some of them coincide with those from the first group). There are some of them:

k_1 (the number of clicks per minute from one device): an important coefficient in fraud detection, however, a user having just this coefficient cannot unequivocally identify a fraudster (anomaly),

k_4 (time between user events): similar to k_1 if actions do not fit a specific fingerprint and the time between events is not specified by distribution;

k_8 (number of friends in a social net): if a user is connected to a social net and has enough friends, namely $(K_{8\_max} * K_{8\_opt}; \infty)$, then it is possible to check names (whether the friends are real) and the friends' photos (similar to definition of k_{15} coefficient);

k_{10}, k_{11} (number of installations from one device, time between installations on one device): make it possible to determine whether the user is organic based on the installation data. If the frequency of installation from one device set by

k_{11} coefficient indicates that the time between installation requests is not less than K_{11_min}, then the user will be considered organic according to these coefficients. However, when determining this coefficient, all other indicators need to be checked. Coefficient K_{11_min} is established on the basis of an expert poll.

Thus, when scaling input data according to the scales proposed in the study, a mathematical model of the scaling process (12) is obtained. It contains coefficients of a definite metric.

$$\vec{I}\begin{pmatrix} U_1(i_1,i_2,...,i_{s1}) \\ U_2(i_1,i_2,...,i_{s2}) \\ ... \\ U_n(i_1,i_2,...,i_{sn}) \end{pmatrix} \rightarrow$$

$$\rightarrow \begin{cases} \vec{G}_1\begin{pmatrix} U_1(g_{11},...,g_{1r}) \\ U_2(g_{11},...,g_{1r}) \\ ... \\ U_n(g_{11},...,g_{1r}) \end{pmatrix} \rightarrow \\ \vec{G}_2\begin{pmatrix} U_1(g_{21},...,g_{2l}) \\ U_2(g_{21},...,g_{2l}) \\ ... \\ U_n(g_{21},...,g_{2l}) \end{pmatrix} \rightarrow \end{cases} \begin{cases} \vec{X}\begin{pmatrix} U_1(x_1,...,x_n) \\ U_2(x_1,...,x_n) \\ ... \\ U_n(x_1,...,x_n) \end{pmatrix} \rightarrow F_1(\vec{X}) \rightarrow \vec{X}_1\begin{pmatrix} U_n(k_{01}) \\ U_n(k_{02}) \\ ... \\ U_n(k_{0n}) \end{pmatrix} \\ \vec{B}\begin{pmatrix} U_1(b_1,...,b_k) \\ U_2(b_1,...,b_k) \\ ... \\ U_n(b_1,...,b_k) \end{pmatrix} \rightarrow F_2(\vec{B}) \rightarrow \vec{B}_1\begin{pmatrix} U_n(k_{11}) \\ U_n(k_{12}) \\ ... \\ U_n(k_{1n}) \end{pmatrix} \rightarrow \\ ... \\ \vec{W}\begin{pmatrix} U_1(w_1,...,w_m) \\ U_2(w_1,...,w_m) \\ ... \\ U_n(w_1,...,w_m) \end{pmatrix} \rightarrow F_3(\vec{W}) \rightarrow \vec{W}_1\begin{pmatrix} U_n(k_{21}) \\ U_n(k_{22}) \\ ... \\ U_n(k_{2n}) \end{pmatrix} \end{cases}$$

$$\rightarrow F_4(\vec{X}_1,\vec{B}_1,\vec{W}_1) \rightarrow \vec{D}\begin{pmatrix} U_1(k_{01},k_{11},...,k_{21}) \\ U_1(k_{02},k_{12},...,k_{22}) \\ ... \\ U_n(k_{0n},k_{1n},...,k_{2n}) \end{pmatrix} \rightarrow F_5(\vec{D}) \rightarrow \vec{R}\begin{pmatrix} U_1(C_1) \\ U_2(C_0) \\ ... \\ U_n(C_1) \end{pmatrix}, \quad (12)$$

where

$$\vec{I}\begin{pmatrix} U_1(i_1,i_2,...,i_{s1}) \\ U_2(i_1,i_2,...,i_{s2}) \\ ... \\ U_n(i_1,i_2,...,i_{sn}) \end{pmatrix}$$

is information from the database for each user, namely the vector containing vectors with all defined features for each user ($U_1,U_2,...,U_n$);

$\vec{G}_1$ and $\vec{G}_2$ are vectors of heterogeneous input data divided into two groups, namely, as established on the basis of an expert poll;

$\vec{X},\vec{B},...,\vec{W}$ are vectors of homogeneous data grouped by types;

$F_1(\vec{X})$, $F_2(\vec{B})$,..., $F_3(\vec{W})$ are corresponding functions of conversion of homogeneous data by a certain feature to a criterion with value from 0 to 1. In the output, $\vec{X}_1$, $\vec{B}_1$, $\vec{W}_1$, vectors will be obtained containing users with the criterion value according to a corresponding feature;

$F_4(\vec{X}_1,\vec{B}_1,...,\vec{W}_1)$ is the function of combining all criteria by users in a $\vec{D}$ vector;

$F_5(\vec{D})$ is the function of user classification in clusters C_0 (fraudsters) and C_1 (organic users). The result of this classification will be presented as a $\vec{R}$, vector of users, and each user in it will have a corresponding class as a parameter.

Two algorithms for scaling diverse data arrays have been developed on the basis of the proposed mathematical model of scaling (12). Let us consider these algorithms in more detail.

Algorithm 1. The algorithm of scaling heterogeneous data arrays:

1. Analysis of input data.

1. 1. Group data in the $\vec{G}_1$ vector.

1. 2. Group data in the $\vec{G}_2$ vector.

2. Create $\vec{X}$, $\vec{B}$,..., $\vec{W}$. vectors of homogeneous data.

3. Scaling data according to a certain feature in a criterion.

3. 1. Scale $\vec{X}$, vector data using $F_1(\vec{X})$ function.

3. 2. Scale $\vec{W}$, vector data using $F_3(\vec{W})$ function.

3. 3. Scale $\vec{W}$, vector data using $F_3(\vec{W})$ function.

4. Identifying uniquely known fraudsters and organic users and building a knowledge base.

4. 1. Identify definitely fraudster users based on the criteria obtained and record such users into the knowledge base.

4. 2. Identify definitely organic users based on the criteria obtained and record their data into the knowledge base.

4. 3. Identify suspicious users based on the criteria obtained and record their data into the knowledge base.

5. Combine criteria by users using function $F_4(\vec{X}_1,\vec{B}_1,...,\vec{W}_1)$ to obtain a set of homogeneous data for each user, U_i, combined into the

$$\vec{D}\begin{pmatrix} U_1(k_{01},k_{11},...,k_{21}) \\ U_1(k_{02},k_{12},...,k_{22}) \\ ... \\ U_n(k_{0n},k_{1n},...,k_{2n}) \end{pmatrix} \text{ vector.}$$

6. Detect fraud with the help of user classification based on homogeneous $\vec{D}$ vector data.

7. Record data on the users that were referred by the algorithm to the C_0 class (fraudster) or C_1 class (organic user) in the knowledge base.

8. Record results of the first algorithm into the knowledge base used for constructing a training procedure used in the algorithm 2.

As a result of operation of Algorithm 1, all input data are scaled and made homogeneous.

Algorithm 2 makes it possible to detect abnormalities in data (fraud) using coefficients of similarity between users. The proposed algorithm works with both data and user data arrays. Let us consider it.

Algorithm 2. The algorithm of detecting fraud when installing mobile applications:

1. Connect to the user database.

2. Connect to the knowledge base obtained by means of the algorithm 1.

3. Provide feature engineering.

4. Convert the (data) features into coefficients with forming rules of coefficient definition.

5. Group the coefficients, namely, into $\vec{G}_1$ and $\vec{G}_2$ vectors.

6. Determine values of the coefficients from the first group according to the characteristics of the data defined at the second stage of expert evaluation. For example:

6. 1. If the data are numerical, then value of the coefficient usually depends on the limit values.

6. 2. If the data are qualitative, then the coefficient value usually depends on affiliation of the data to a set of qualitative data predefined by means of the expert poll.

6. 3. If this is a set of data, then the coefficient value usually depends on affiliation of the given set to a certain known distribution law or on whether the amount of data (of a particular type) from the given set belongs to the limit data.

7. Define the sets of fraudsters, organic and suspicious users based on the coefficients of the first group, i. e. the coefficients obtained on the basis of the $\vec{G}_1$ vector using formula (12).

8. Select the set of undefined users.

9. Determine values of the coefficients from the second group using formulas for determining similarity of users. To do this, the following has to be done for each of the determined coefficients from the second group:

9. 1. Determining coefficients of similarity of users with fraudsters forming a set of coefficients having values ranging from 0 to 1.

9. 2. Determining coefficients of similarity of users with organic users.

9. 3. Determining coefficients of similarity of users with suspicious users.

10. Combine the obtained sets of coefficient values into one set of homogeneous values.

11. Direct homogeneous values to the classification model.

11. 1. Training the model for the data noticed in p. 6 using cross-validation to avoid re-training of the model.

11. 2. Running the model for undefined users to obtain class of each user.

12. Augment training rules and update the knowledge base.

13. Interpret the data obtained from the knowledge base.

Based on the proposed algorithms, a scheme for detecting fraud in installation of mobile applications was developed (Fig. 6).

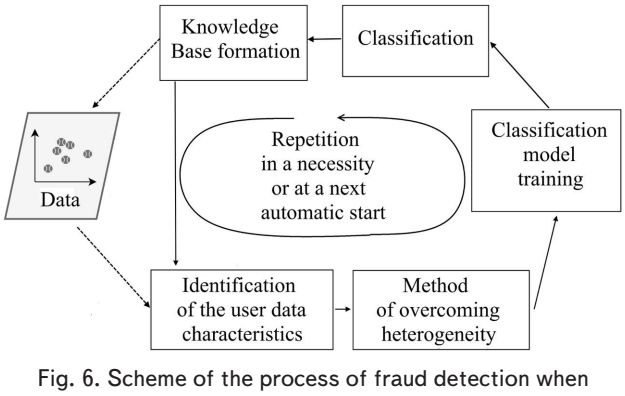

Fig. 6. Scheme of the process of fraud detection when installing mobile applications

Fig. 6 shows the sequence of conversion of large arrays of heterogeneous data into fraudster fingerprints based on the proposed method of fraud detection in heterogeneous data, the mathematical model of the scaling process and the algorithms developed in this study.

6. Analysis of results obtained in the study of the proposed model, algorithms and method of fraud detection in heterogeneous data

To test the proposed model, algorithms and method, an experimental study was conducted [29, 30]. A representative sample of data from a mobile application containing 284807 aggregated user actions was taken as a test set for verification. An example of a user event is shown in Fig. 7. Note that general user data, such as the user IP address, device ID which are personal data as well and stored in the session collection. Relevant information for each event can be found by session_id. This is done to optimize storage resources as such data will not change within the session.

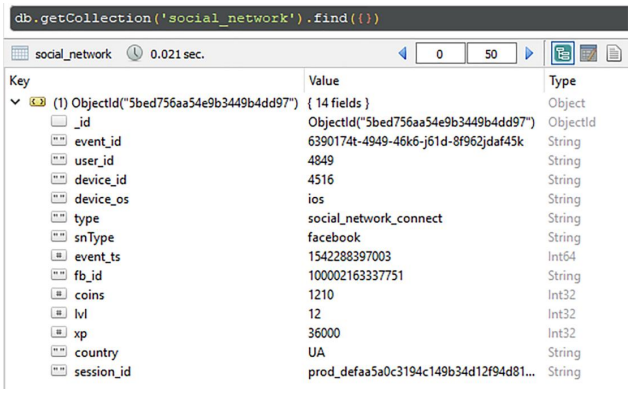

Fig. 7. Example of one of the events from the test set

Combine the event data for each user into the vector of input data

$$\vec{I}\begin{pmatrix} U_1(i_1,i_2,...,i_{s1}) \\ U_2(i_1,i_2,...,i_{s2}) \\ ... \\ U_n(i_1,i_2,...,i_{sn}) \end{pmatrix}.$$

Consequently, the $\vec{I}$ vector for each of the existing users will contain a plurality of event types. An example of one of them was discussed above.

Let us form two groups of the input data, $\vec{G}_1$ and $\vec{G}_2$ from the $\vec{I}$ vector selected by means of an expert poll.

Using the proposed method of overcoming heterogeneity, convert each of the features into a criterion having value ranging from 0 to 1 (the previously mentioned functions $F_1(\vec{X})$, $F_2(\vec{B})$,..., $F_3(\vec{W})$)) and combine the criteria by users (earlier mentioned function $F_4(\vec{X}_1,\vec{B}_1,\vec{W}_1)$) in a $\vec{D}$ vector. Data for the first five users represented by the $\vec{D}$ vector were demonstrated by the use of the Python programming language in Fig. 8 where the record number is the user ID in a sequence; *V*1, *V*2,... are the criteria for each user; *Time* is the system shift.

As can be seen from Fig. 8, all data are homogeneous, namely, there is no qualitative data and all data are in the same range (from 0 to 1).

```
appActions.head()
```

Out[2]:

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...
0	0.0	0.359807	0.072781	0.536347	0.378155	0.338321	0.462388	0.239599	0.098698	0.363787	...
1	0.0	0.191857	0.266151	0.166480	0.448154	0.060018	0.082361	0.078803	0.085102	0.255425	...
2	1.0	0.358354	0.340163	0.773209	0.379780	0.503198	0.800499	0.791461	0.247676	0.514654	...
3	1.0	0.966272	0.185226	0.792993	0.863291	0.010309	0.247203	0.237609	0.377436	0.387024	...
4	2.0	0.158233	0.877737	0.548718	0.403034	0.407193	0.095921	0.592941	0.270533	0.817739	...

Fig. 8. Homogeneous data for each user obtained as a result of use of the method of overcoming heterogeneity

After completion of classification for obtained homogeneous data, a resultant

$$\vec{R}\begin{pmatrix} U_1(C_1) \\ U_2(C_0) \\ \dots \\ U_n(C_1) \end{pmatrix}$$

vector was obtained at the output. For visualization of the result, distribution of users by classes is shown in Fig. 9. Classes 0 and 1 include fraudsters and organic users, respectively.

```
print(appActions.groupby('Class').size())
```

```
Class
1    284315
0       492
dtype: int64
```

Fig. 9. Distribution of users in two classes

For a better understanding of the results obtained and visual tracking of the trend in the data, input data of one user assigned by the algorithm to the class 0 can be represented as histograms (Fig. 10). As the figure shows, the sets of data, *V*13 and *V*14, belong to the normal distribution law. Consequently, the given user applied a script which has set time of his events according to the normal distribution law.

According to the results of experimental studies carried out using the developed computer programs based on the algorithms proposed in the study (for which author's certificates were issued [29, 30]), accuracy of fraudster detection for a given representative sample was 99.14 %. The software was implemented using the Python programming language and TensorFlow, Pandas and Numpy libraries.

7. Discussion of results obtained in the study of the proposed model, algorithms and method of fraud detection in heterogeneous data

Advantage of the proposed method of fraud detection over existing methods discussed in Section 2 [1, 5–25] consists in that it allows users to work with heterogeneous input data when installing mobile applications with application of algorithms of data scaling. One more advantage is creation of a knowledge base and fingerprints for fraudsters and organic users at the stage of executing the algorithm 2 of fraud detection when installing mobile applications. Availability of the knowledge base accelerates fraud detection for new sets of input data and detection of user fingerprints makes it possible to identify fraudsters that cannot be identified by analogous systems.

The developed method can be used in automatic detection of fraudsters by using intelligent data analysis. Its limitation consists in that it can be used only in the field of installing mobile applications. To adapt this method to other areas, it is necessary to classify the input data for their use by the proposed scaling algorithms.

Further development of this study consists in paralleling the proposed processes and raising computation power for future experiments. This will enable speeding up fraud detection when installing mobile applications and determining minimum amount of resources required.

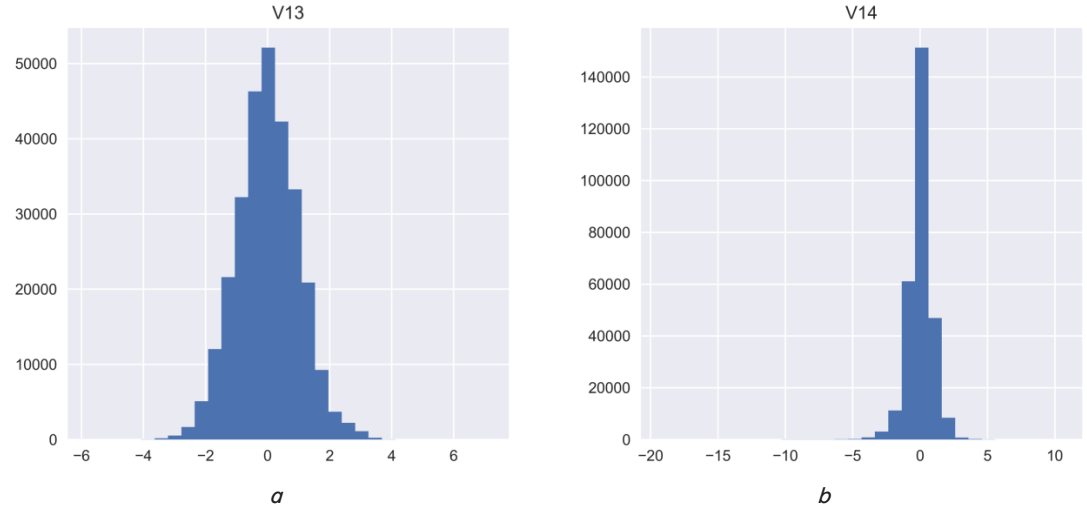

Fig. 10. Demonstration of scaled characteristics, V13, V14, of the first group, $\vec{G}_1$ of one of the fraudsters in the form of histograms: distribution of time between the user events in the chosen time interval (a); time distribution between an internal event of "preference" type (*b*)

Thus, this paper proposes a method for detecting anomalies (fraud) in large arrays of heterogeneous data, a mathematical model of the process of scaling large arrays of heterogeneous data and scaling algorithms. The developed method, mathematical model and algorithms can be used in automatic detection of fraudsters (anomalies in data) when installing mobile applications.

7. Conclusions

1. Heterogeneous data used during installation of mobile applications have been classified which has enabled further development of the method of fraud detection when installing mobile applications.

2. A method for fraud detection in heterogeneous data arrays during installation of mobile applications was proposed. In contrast to existing methods, it makes it possible to define complete formats of fraudsters using intelligent data analysis. In the course of development of the method, an algorithm of scaling heterogeneous data sets was proposed based on the proposed scales and a mathematical scaling model which makes it possible to reduce the whole set of data to two homogeneous groups. Also, algorithms for processing the obtained groups of homogeneous data and detection of fraudsters with a full set of fraudster characteristics were proposed. These algorithms and the method were used in the intelligent automatic fraudster detection system intended for conducting experimental studies. As it was shown by the results of experimental study, application of the method and algorithms as well as definition of complete formats of fraudsters with the help of intelligent data analysis can improve accuracy of solution of the set task up to 99.14 %.

3. The results obtained in the experimental study have shown effectiveness of automatic fraud detection and the possibility of expanding formats and characteristics of fraudulent users based on intellectual analysis and knowledge bases. The study was carried out using software developed on the basis of the model, method and algorithms proposed in this paper with the help of the Python programming language and TensorFlow, Pandas and Numpy libraries in the PyCharm development environment. To implement the classification block, a fully-connected deep neural network with 3 hidden layers was used.

References

1. Chandola V., Banerjee A., Kumar V. Anomaly detection // ACM Computing Surveys. 2009. Vol. 41, Issue 3. P. 1–58. doi: https://doi.org/10.1145/1541880.1541882
2. Conditional Anomaly Detection / Song X., Wu M., Jermaine C., Ranka S. // IEEE Transactions on Knowledge and Data Engineering. 2007. Vol. 19, Issue 5. P. 631–645. doi: https://doi.org/10.1109/tkde.2007.1009
3. Gricenko A. V. Tipy anomaliy v videoizobrazheniyah // Tekhnicheskie nauki – ot teorii k praktike: sbornik statey po materialam VII mezhdunarodnoy nauchno-prakticheskoy konferencii. Chast' I. Novosibirsk: SibAK, 2012. URL: https://sibac.info/conf/tech/vii/26730
4. Prado-Romero M. A., Gago-Alonso A. Detecting contextual collective anomalies at a Glance // 2016 23rd International Conference on Pattern Recognition (ICPR). 2016. doi: https://doi.org/10.1109/icpr.2016.7900017
5. Géron A. Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. Aurélien Géron, O'Reilly Media, 2017. 574 p.
6. Cielen D., Meysman A. D. B., Ali M. Introducing Data Science: Big data, machine learning, and more, using Python tools. Manning, 2016. 320 p.
7. Guido S., Müller A. Introduction to Machine Learning with Python: A Guide for Data Scientists. O'Reilly Media, 2016. 400 p.
8. Chollet F. Deep Learning with Python. Manning, 2017. 384 p.
9. Agrawal R., Srikant R. Mining sequential patterns // Proceedings of the Eleventh International Conference on Data Engineering. 1995. doi: https://doi.org/10.1109/icde.1995.380415
10. Agarwal D. An Empirical Bayes Approach to Detect Anomalies in Dynamic Multidimensional Arrays // Fifth IEEE International Conference on Data Mining (ICDM'05). 2005. doi: https://doi.org/10.1109/icdm.2005.22
11. Siaterlis C., Maglaris B. Towards multisensor data fusion for DoS detection // Proceedings of the 2004 ACM symposium on Applied computing – SAC '04. 2004. doi: https://doi.org/10.1145/967900.967992
12. Agarwal D. Detecting anomalies in cross-classified streams: a Bayesian approach // Knowledge and Information Systems. 2006. Vol. 11, Issue 1. P. 29–44. doi: https://doi.org/10.1007/s10115-006-0036-4
13. MachineLearning.ru. Professional'nyy informacionno-analiticheskiy resurs, posvyashchennyy mashinnomu obucheniyu, raspoznavaniyu obrazov i intellektual'nomu analizu dannyah. URL: http://www.machinelearning.ru
14. Polhul T. D., Yarovyi A. A. Vyznachennia shakhraiskykh operatsiy pry vstanovlenni mobilnykh dodatkiv z vykorystanniam intelektualnoho analizu danykh // Suchasni tendentsiyi rozvytku systemnoho prohramuvannia. Tezy dopovidei. Kyiv, 2016. P. 55–56. URL: http://ccs.nau.edu.ua/wp-content/uploads/2017/12/%D0%A1%D0%A2%D0%A0%D0%A1%D0%9F_2016_07.pdf
15. Polhul T. D., Yarovyi A. A. Vyznachennia shakhraiskykh operatsiy pry instaliatsiyi mobilnykh dodatkiv z vykorystanniam intelektualnoho analizu danykh // Materialy XLVI naukovo-tekhnichnoi konferentsiyi pidrozdiliv VNTU. Vinnytsia, 2017. URL: http://ir.lib.vntu.edu.ua/bitstream/handle/123456789/17200/2158.pdf?sequence=3
16. Program applications install fraud detection using data mining / Yarovyi A. A., Romanyuk O. N., Arsenyuk I. R., Polhul T. D. // Naukovi pratsi Donetskoho natsionalnoho tekhnichnoho universytetu. Seriya: "Informatyka, kibernetyka ta obchysliuvalna tekhnika". 2017. Issue 2 (25). P. 126–131. URL: http://science.donntu.edu.ua/wp-content/uploads/2018/03/ikvt_2017_2_site-1.pdf
17. Yarovyi A., Polhul T., Krylyk L. Rozrobka metodu vyiavlennia shakhraistva pry instaliuvanni mobilnykh dodatkiv z vykorystanniam intelektualnoho analizu danykh // Materialy konferentsiyi «XLVII Naukovo-tekhnichna konferentsiya pidrozdiliv Vin-

nytskoho natsionalnoho tekhnichnoho universytetu (2018)». Vinnytsia, 2018. URL: http://ir.lib.vntu.edu.ua/bitstream/handle/123456789/22722/079.pdf?sequence=1

18. Kiulian A. H., Polhul T. D., Khazin M. B. Matematychna model rekomendatsiynoho servisu na osnovi metodu kolaboratyvnoi filtratsiyi // Kompiuterni tekhnolohiyi ta Internet v informatsiynomu suspilstvi. 2012. P. 226–227. URL: http://ir.lib.vntu.edu.ua/bitstream/handle/123456789/7911/226-227.pdf?sequence=1&isAllowed=y
19. Segaran T. Programming Collective Intelligence. Building Smart Web 2.0 Applications. O'Reilly Media, 2008. 368 p.
20. Yeung D.-Y., Chow C. Parzen-window network intrusion detectors // Object recognition supported by user interaction for service robots. 2002. doi: https://doi.org/10.1109/icpr.2002.1047476
21. Hodge V. J., Austin J. A Survey of Outlier Detection Methodologies // Artificial Intelligence Review. 2004. Vol. 22, Issue 2. P. 85–126. doi: https://doi.org/10.1007/s10462-004-4304-y
22. Agyemang M., Barker K., Alhajj R. A comprehensive survey of numeric and symbolic outlier mining techniques // Intelligent Data Analysis. 2006. Vol. 10, Issue 6. P. 521–538. doi: https://doi.org/10.3233/ida-2006-10604
23. Keogh E., Lin J., Fu A. HOT SAX: Efficiently Finding the Most Unusual Time Series Subsequence // Fifth IEEE International Conference on Data Mining (ICDM'05). 2005. doi: https://doi.org/10.1109/icdm.2005.79
24. Finding the most unusual time series subsequence: algorithms and applications / Keogh E., Lin J., Lee S.-H., Herle H. V. // Knowledge and Information Systems. 2006. Vol. 11, Issue 1. P. 1–27. doi: https://doi.org/10.1007/s10115-006-0034-6
25. Donoho S. Early detection of insider trading in option markets // Proceedings of the 2004 ACM SIGKDD international conference on Knowledge discovery and data mining – KDD '04. 2004. doi: https://doi.org/10.1145/1014052.1014100
26. Finding Time Series Discords Based on Haar Transform / Fu A. W., Leung O. T.-W., Keogh E., Lin J. // Lecture Notes in Computer Science. 2006. P. 31–41. doi: https://doi.org/10.1007/11811305_3
27. Yarovyi A. A., Polhul T. D. Pidvyshchennia produktyvnosti obchysliuvalnykh protsesiv v paralelno-ierarkhichniy merezhi za dopomohoiu Framework Benchmark Akka // Zbirnyk tez dopovidi VII Mizhnarodnoi naukovo-tekhnichnoi konferentsiyi «Fotonika ODS-2015». Vinnytsia, 2015. P. 9.
28. Baudat G., Anouar F. Generalized Discriminant Analysis Using a Kernel Approach // Neural Computation. 2000. Vol. 12, Issue 10. P. 2385–2404. doi: https://doi.org/10.1162/089976600300014980
29. Yarovyi A. A., Polhul T. D. Kompiuterna prohrama «Prohramnyi modul zboru danykh informatsiynoi tekhnolohiyi» vyiavlennia shakhraistva pry instaliuvanni prohramnykh dodatkiv. Cvidotstvo pro reiestratsiu avtorskoho prava na tvir No. 76348. Kyiv: Ministerstvo ekonomichnoho rozvytku i torhivli Ukrainy, 2018.
30. Yarovyi A. A., Polhul T. D. Kompiuterna prohrama «Prohramnyi modul vyznachennia skhozhosti korystuvachiv informatsiynoi tekhnolohiyi vyiavlennia shakhraistva pry instaliuvanni prohramnykh dodatkiv». Cvidotstvo pro reiestratsiu avtorskoho prava na tvir No. 76347. Kyiv: Ministerstvo ekonomichnoho rozvytku i torhivli Ukrainy, 2018.