Creating Test Data for Market Surveillance Systems with Embedded Machine Learning Algorithms

O. Moskaleva <olga.moskaleva@exactprosystems.com>
A. Gromova <anna.gromova@exactprosystems.com>
Exactpro, LSEG,
20A Building 4, 2nd Yuzhnoportovy Proezd, Moscow, 115088, Russia

Abstract. Market surveillance systems, used for monitoring and analysis of all transactions in the financial market, have gained importance since the latest financial crisis. Such systems are designed to detect market abuse behavior and prevent it. The latest approach to the development of such systems is to use machine learning methods that largely improve the accuracy of market abuse predictions. These intelligent market surveillance systems are based on data mining methods, which build their own dependencies between the variables. It makes the application of standard user-logic-based testing methodologies difficult. Therefore, in the context of intelligent surveillance systems, we built our own model for classifying the transactions. To test it, it is important to be able to create a set of test cases that will generate obvious and predictable output. We propose scenarios that allow to test the model more thoroughly, compared to the standard testing methods. These scenarios consist of several types of test cases which are based on the equivalence classes methodology. The division into equivalence classes is performed after the analysis of the real data used by real surveillance systems. We tested the created model and discovered how this approach allows to define its weaknesses. This paper describes our findings from using this method to test a market surveillance system that is based on machine learning techniques**.**

Keywords: test data; equivalence classes; market surveillance systems; machine learning

DOI: 10.15514/ISPRAS-2017-29(4)-18

For citation: Moskaleva O., Gromova A. Creating Test Data for Market Surveillance Systems with Embedded Machine Learning Algorithms. *Trudy ISP RAN/Proc. ISP RAS*, vol. 29, issue 4, 2017, pp. 269-282. DOI: 10.15514/ISPRAS-2017-29(4)-18

1. Introduction

1.1 Market surveillance systems

Electronic trading platforms have become an increasingly important part of the financial market in recent years. They are obligated to take legal responsibilities [1], [2] and correspond to the law and the regulatory requirements. Therefore, all market events in the contemporary electronic trading platforms are monitored and analysed by market surveillance systems.

Such systems are designed to detect market abuse behavior and prevent it. Their main goals are detection and prevention of such market abuse cases as insider trading, intentional and aggressive price positioning, creation of fictitious liquidity, money laundering, marking the close, etc. [3]. Different data mining methods are used for improving the quality of the surveillance systems' work [4], [5], [6], [7], [8], [9].

1.2 Quality assurance for market surveillance

The standard quality assurance (QA) methods and technologies seem to be powerless in regard to machine learning (ML) applications. C. Murphy, G. Kaiser, M. Arias even introduced a concept of "non-testable" applications [10]. From the QA perspective, we do not have to test whether an ML algorithm is learning well, but to ensure that the application uses the algorithm correctly, implements the specification and meets the users' expectations. In this paper, we employ the term "testing" in accordance with the QA theory.

It is clear that a sufficient input data set is needed for high-quality testing coverage. Furthermore, the testing data set should be as close as possible to the real data or should even be real. So, which approach should be used for creating data to verify the implementation of an ML-algorithm more fully?

We can test a market surveillance system in the following ways:

- by creating test cases which are based on the knowledge of the business rules from the specification. Such test data are similar to the real users' behaviour;
- by generating various datasets which contain different combinations of variable.

Both variants are suitable for the surveillance systems that use standard control flows, like loops or choices. For standard systems, there is a set of rules, which allows getting a clear output result for specific input data. When it comes to intelligence systems, it is not normally obvious what will happen as a result of certain input because an ML algorithm builds its own dependencies between the variables and human interpretation of such dependencies is impossible. Because of this, it is important to be able to create a set of test cases that will generate obvious

and predictable output. Therefore, the second approach to generating the test data allows for the creation of output that is easily interpretable.

1.3 Contribution

This paper introduces the following contribution:

- creating a model for classifying the transactions. This model can be used for the detection of market manipulations;
- test cases generation for defining the weaknesses of the created model. The test cases are based on the equivalence classes;
- testing the prototype based on the created model and the analysis of the received results.

2. Related work

2.1 Ongoing problems in the quality assurance of the modern surveillance systems

It is known that the system containing ML algorithms should learn using a dataset that is real or close to real. Obviously, for testing purposes, it is necessary to use a dataset with a similar structure.

Moreover, during the creation of this dataset, it will be helpful to emphasize the variability of values of the attributes included in the sampling. By generating a variety of combinations with different values, we will create test cases to find out the weaknesses in the ML algorithm, which are related to the separation of classes.

2.2 Existing approaches

C. Murphy, G. Kaiser, M. Arias proposed to divide the data into equivalence classes to generate test cases for testing the "non-testable" software [10]. It should be noted that forming equivalence classes is a standard approach in quality assurance [11].

However, C. Murphy, G. Kaiser, M. Arias suggest to follow three steps in testing this kind of software. Firstly, the data should be divided into several classes, taking into account the size of the dataset, the potential ranges of the attributes and the label values, etc. Then, the test cases, that are based on the investigation of the ML algorithm used in this system, should be created. At last, several testing datasets should be generated.

Supposing that a dataset for QA purposes has been defined, based on the knowledge of the method used in the machine learning system. For solving the problem of the dataset sufficiency, C. Murphy, G. Kaiser, M. Arias propose a “parameterized random data generation” methodology. This methodology enables us to generate large datasets and randomly control them [12], [13], [14].

In their paper, J. Zhang et al. suggest to use predictive mutation testing, as it allows to make decisions without executing the costly mutation testing [15].

Thus, it becomes clear that new methodologies for testing ML applications are being developed. However, the contemporary market surveillance systems impose additional requirements on them. These systems should be able to self-detect the incorrect behavior and classify alerts, and further point at the initiator of this behavior. This should always be noted during the process of creation of test scenarios.

3. Definitions and assumptions

3.1 Structure of the transaction log

A transaction is an event that happens on the financial market and changes any financial instrument, for example:

- submitting a buy-order for security S with price P and volume V;
- cancelling an order with id I;
- trading on security S at price P with volume V, etc.

Transaction logs are the data that are used by the ML surveillance system. Each transaction is stored as an object and has a set of input parameters (transaction characteristics) and one output parameter (presence or absence of the alert):

$I = \{i_1, i_2, \dots, i_j, \dots, i_n\}$, where $i = \overline{\{1, n\}}$;

$i_j = \{TID, B, IID, Side, CP, ExP, ExS, TV, S, TInt, Alert\}$;

Where:

$TID \in N$,

$B = \{Broker_1, Broker_2, \dots, Broker_k\}$, where k is the number of brokers that are available in the configuration file,

$IID = \{Instrument_1, Instrument_2, \dots, Instrument_m\}$,where m is the number of instruments that are available in the configuration file,

$Side = \{Buy, Sell\}$,

$CP \in Q$,

$ExP \in Q$ and $ExP \geq 0$,

$ExS \in Z$,

$TV \in Z$,

$S = \{OAC, RT, CAC, ReOAC, ResumeAC, Halt\}$,

$TInt \in N$,

$Alert = \{0,1\}^M$.

For more attribute details, please refer to Table 1.

Table 1. Attribute details

Parameter	Type	Comment

Transaction ID	numerical	Unique identifier of transaction
Broker	categorical	Company to which the trader belongs
Instrument ID	categorical	Identifier of the instrument to be traded
Side	categorical	Side of the order
ChangePrice	numerical	$ChP = LastTradedPrice - CurrentPrice$
Executed Price	numerical	Price of the trade
Executed Size	numerical	Volume of the Trade
Total Volume	numerical	Total volume traded on the instrument
Session	categorical	Current session on the instrument
Time Interval	interval	Time interval between transactions
Alert	boolean	The output: 0 - regular transaction,1 - suspicious transaction

Each transaction can cause several different alerts, that is why every alert should be classified. This type of classification is called a classification with overlapping classes.

3.2 Market abuse alert

The following type of behavior can be considered as abusive: a broker tries to raise or bring down the price on several trades. The alert will be triggered if the price deviation and the traded volume reach the threshold values. If the price goes up, the buy-orders should be checked, if the price goes down – the sell ones.

4. Background

As each transaction is stored as an object and has a set of independent variables and one dependent variable, for testing purposes, these data should be divided into several equivalence classes. All the objects in one equivalence class have the same characteristics of certain attributes [11] and trigger the same system behavior.

It is important to analyze the transactions, the system behavior and the meaningful parameters before forming the classes. We can use the following types of test cases which are based on equivalence classes. These classes are defined after the analysis of real data used by the surveillance systems:

1) **Consistency checks** with consideration of categorical transaction attributes ($Firm$, $Session$, $Action$, etc.):

    a) Let $a = a_1$ and $class = 1$, and $a = a_2$ and $class = 0$. The categorical attribute gets a concrete value for several test cases,

but the other transaction attributes have different values. For such combinations, the $class$ value is set to 1. This is a check for how strongly the $class$ value correlates with the concrete value of the I-th attribute.

b) Let a categorical attribute get any values, always leaving the $class$ set to 1. Let attribute Let $b = b_1$ or $b = b_2$ or $b = b_3$, $class = 1$. It checks that the system can assume that this attribute is inessential.

c) Let $0 \leq c \leq 10$, $class = 1$. The same as for (b), but for numeric variables.

d) Let $c \geq 0$, $class = 1$. The same as for (b), but for numeric variables.

2) **Checks for empty values.** Due to the fact that a surveillance system analyses all the transactions on the financial market and that the considered transaction can be different for each type of alert, it is possible that some values will be empty.

3) **Checks for the presence of noise.** We perform checks for the presence of noise using the equivalence classes for numeric transaction attributes. Some attributes can have values in the concrete range, but mostly they take the average value. For example, the price percentage variable takes the values from 0 to 100 with the mean of 50 and has a low variance. Even if the border values (0 and 100) are included in the acceptable range, they appear so rarely that we treat them as frequency emissions.

5. Approach

5.1 Classification model of market abuse

To prove that the proposed methodology is effective, we have created a model which allows to classify the transactions by one type of abusive behavior. It should be noted that we have to deal with rather specific data, i.e. financial transactions.

It is important to make a thorough analysis of the data related to the business logic, the system requirements and the structure of transaction logs, to be able to define the attributes and their particular qualities. The correct outcomes can only be provided when considering all these factors. Thus, this particular dataset structure, as it is presented in Table 1, was selected for this type of abusive behavior.

We extracted 639 transactions from the messages of one of the electronic trading protocols and manually classified them, using these data to build the classifier. We used a decision tree algorithm for our research. After that, the classifier was trained on a set and its performance was evaluated.

Precision, recall, and F-measure are the evaluation metrics that we used for our calculations. Values of these metrics that we received are represented in Table 2.

Table 2. Values of metrics

Precision	Recall	F-measure
0.615	0.678	0.645

Table 3. Detailed examples of test cases

Scenario 1: Consistency checks 1.a. Instrument		
Test case 1	Test case 2	Test case 3
$B = Broker_1$ $IID = Instrument_1$ $Side = Buy$ $CP = CP_1$ $ExP = P_1$ $ExS = S_1$ $TV = TV_1$ $S = RT$ $TInt = TI_1$ $Alert = 1$	$B = Broker_2$ $IID = Instrument_1$ $Side = Sell$ $CP = CP_1 - 1$ $ExP = P_2$ $ExS = S_2$ $TV = TV_2$ $S = RT$ $TInt = TI_2$ $Alert = 1$	$B = Broker_3$ $IID = Instrument_1$ $Side = Sell$ $CP = CP_1 + 2$ $ExP = P_3$ $ExS = S_3$ $TV = TV_3$ $S = RT$ $TInt = TI_3$ $Alert = 1$
Scenario 2: Consistency checks 1.b. Broker		
Test case 1	Test case 2	Test case 3
$B = Broker_2$ $IID = Instrument_1$ $Side = Buy$ $CP = CP_1$ $ExP = P_1$ $ExS = S_1$ $TV = TV_1$ $S = RT$ $TInt = TI_1$ $Alert = 1$	$B = Broker_3$ $IID = Instrument_1$ $Side = Buy$ $CP = CP_1$ $ExP = P_1$ $ExS = S_1$ $TV = TV_1$ $S = RT$ $TInt = TI_1$ $Alert = 1$	$B = Broker_1$ $IID = Instrument_1$ $Side = Buy$ $CP = CP_1$ $ExP = P_1$ $ExS = S_1$ $TV = TV_1$ $S = RT$ $TInt = TI_1$ $Alert = 1$
Scenario 3: Consistency checks 1.c. Executed Price		
Test case 1	Test case 2	Test case 3
$B = Broker_2$ $IID = Instrument_1$ $Side = Buy$ $CP = CP_1$	$B = Broker_2$ $IID = Instrument_1$ $Side = Buy$ $CP = CP_1$	$B = Broker_2$ $IID = Instrument_1$ $Side = Buy$ $CP = CP_1$

$ExP = P_1$ $ExS = S_1$ $TV = TV_1$ $S = RT$ $TInt = TI_1$ $Alert = 1$	$ExP = P_2$ $ExS = S_1$ $TV = TV_1$ $S = RT$ $TInt = TI_1$ $Alert = 1$	$ExP = P_3$ $ExS = S_1$ $TV = TV_1$ $S = RT$ $TInt = TI_1$ $Alert = 1$
Scenario 4: Consistency checks 1.d. Total Volume		
Test case 1	Test case 2	Test case 3
$B = Broker_2$ $IID = Instrument_1$ $Side = Buy$ $CP = CP_1$ $ExP = P_1$ $ExS = S_1$ $TV = TV_1$ $S = RT$ $TInt = TI_1$ $Alert = 1$	$B = Broker_2$ $IID = Instrument_1$ $Side = Buy$ $CP = CP_1$ $ExP = P_1$ $ExS = S_1$ $TV = TV_2$ $S = RT$ $TInt = TI_1$ $Alert = 1$	$B = Broker_2$ $IID = Instrument_1$ $Side = Buy$ $CP = CP_1$ $ExP = P_1$ $ExS = S_1$ $TV = TV_3$ $S = RT$ $TInt = TI_1$ $Alert = 1$
Scenario 5: Checks for empty values. Executed Price		
Test case 1	Test case 2	Test case 3
$B = Broker_2$ $IID = Instrument_1$ $Side = Buy$ $CP = CP_1$ $ExP = empty$ $ExS = S_1$ $TV = TV_1$ $S = RT$ $TInt = TI_1$ $Alert = 1$	$B = Broker_3$ $IID = Instrument_2$ $Side = Sell$ $CP = CP_1$ $ExP = empty$ $ExS = S_2$ $TV = TV_2$ $S = RT$ $TInt = TI_1$ $Alert = 1$	$B = Broker_1$ $IID = Instrument_3$ $Side = Buy$ $CP = CP_1$ $ExP = empty$ $ExS = S_3$ $TV = TV_4$ $S = RT$ $TInt = TI_1$ $Alert = 1$
Scenario 6: Checks for the presence of noise. Executed Size.		
Test case 1	Test case 2	Test case 3
$B = Broker_2$ $IID = Instrument_1$ $Side = Buy$ $CP = CP_1$ $ExP = P_1$ $ExS = S_1$	$B = Broker_2$ $IID = Instrument_1$ $Side = Buy$ $CP = CP_1$ $ExP = P_1$ $ExS = S_2$	$B = Broker_2$ $IID = Instrument_1$ $Side = Buy$ $CP = CP_1$ $ExP = P_1$ $ExS = S_2$

$TV = TV_1$ $S = RT$ $TInt = TI_1$ $Alert = 1$	$TV = TV_1$ $S = RT$ $TInt = TI_1$ $Alert = 1$	$TV = TV_1$ $S = RT$ $TInt = TI_1$ $Alert = 1$

5.2 Prototype Testing

To test our prototype, we have created a dataset with the same structure as the training dataset and performed all the validations described in Section 4. For each type of the validation, we created the test cases based on the equivalence classes. Please refer to Table 3 for detailed examples.

Figure 1 illustrates the proposed approach. Test Data is the whole set of data that will be used to test ML-based surveillance system. Then these data are divided into equivalence classes, as proposed in Section 4. The generated test cases are used for testing Market Surveillance Systems based on ML.

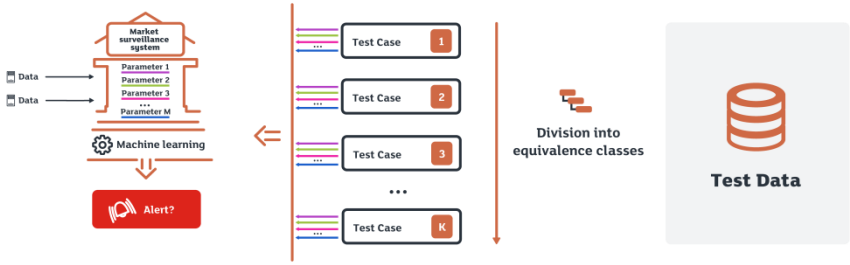

Fig. 1. Prototype Testing

6. Results

After testing our prototype, we received the values of the evaluation metrics that are presented in Table 4.

Table 4. Values of the evaluation metrics

Precision	Recall	F-measure
1.000	0.662	0.797

After a detailed analysis of the instances where the model detected an error, it was observed that errors occurred in:

- different Brokers - Scenario 2: Consistency checks;
- empty values - Scenario 5: Checks for empty values;
- border values - Scenario 6: Checks for the presence of noise.

- Thus, the test that we performed revealed some weaknesses in the ML algorithm:
- An error occurred in one of the categorical attributes during the consistency checks. It may indicate that the model does not take into account the categorical attribute in the algorithm.
- Classification errors occurred in empty values, so the algorithm may miss some abnormal behavior when encountering them. For example, all manipulations at the start of the day may be missed by the surveillance system.
- Some errors were detected in the border values used for the Executed Size variable. According to the general quality assurance theory, we should validate the border values for numeric variables. But in our experiment, unlike what was expected, we saw that the algorithm failed to validate such test cases. The reason is possibly linked with the distribution of the training sample.
- Errors in the tests took place due to thoughtless randomization. After analyzing the results, we can conclude that it is crucial to randomise the dataset with the business logic in mind.

According to these results, it is clear that some details of the dataset and the model need to be considered in the future work.

7. Conclusions and future work

This paper presents the following conclusions:

- The analysis of transactions extracted from an electronic trading protocol allowed us to successfully create a model for the classification of market abuse behavior.
- The proposed scenarios allowed to test the model more thoroughly. The following checks were included: consistency checks, checks for border values, checks for empty values, etc.
- The prototype testing revealed some weaknesses that manifested themselves through unexpected behavior in the case of empty values, in the case of border values (for some of the attributes) and also in the case of different values for one of the categorical attributes.

As focus of our future research, we propose to generate the test cases using the equivalence classes methodology. We advise that the equivalence classes be set according to the types of data (categorical, numeric, etc.) and their border values. Such an approach enables us to parameterize the equivalence classes and to further develop an automatic tool that generates the test data.

References

[1]. FCA (financial conduct authority) (online). Available at: https://handbook.fca.org.uk/
[2]. SEC (Securities and Exchange Commission) (online). Available at: https://www.sec.gov/
[3]. FINMAR Financial Stability and Market Confidence Sourcebook (online publication). Available at: https://handbook.fca.org.uk/handbook/FINMAR/
[4]. Cao L., Ou Y., Yu P.: Detecting Abnormal Coupled Sequences and Sequence Changes in Group-based Manipulative Trading Behaviors. In Proc. of KDD'10, Washington, DC, USA, July 25–28, 2010, pp. 85-93
[5]. Donoho S.: Early Detection of Insider Trading in Option Markets In Proc. of KDD'04, Seattle, Washington, USA, August 22–25, 2004, pp. 420-429
[6]. Luo C., Zhao Y., Cao L., Ou Y., Zhang C.: Exception Mining on Multiple Time Series in Stock Market. In Proc. of International Conference on Web Intelligence and Intelligent Agent Technology, IEEE/WIC/ACM, 2008, pp. 690-693
[7]. Nasdaq and Digital Reasoning Establish Exclusive Alliance to Deliver Holistic Next Generation Surveillance and Monitoring Technology (online publication). Available at: http://www.digitalreasoning.com/buzz/nasdaq-and-digital-reasoning-establish-exclusive-alliance-to-deliver-holistic-next-generation-surveillance-and-monitoring-technology.1884035, 23.02.2016
[8]. Ou Y., Cao L., Luo C., Liu L.: Mining Exceptional Activity Patterns in Microstructure Data. In Proc. of International Conference on Web Intelligence and Intelligent Agent Technology, IEEE/WIC/ACM, 2008, pp. 884-887
[9]. Ou Y., Cao L., Yu T., Zhang C.: Detecting Turning Points of Trading Price and Return Volatility for Market Surveillance Agents. In Proc. of International Conferences on Web Intelligence and Intelligent Agent Technology - Workshops, IEEE/WIC/ACM, 2007, pp. 491-494
[10]. Murphy C., Kaiser G., Arias M.: An Approach to Software Testing of Machine Learning Applications. Proc of the 19th International Conference on Software Engineering and Knowledge Engineering (SEKE), Boston MA, Jul 2007, pp. 167-172
[11]. Nautiyal L, Preeti: A Novel Approach of Equivalence Class Partitioning for Numerical Input. ACM SIGSOFT Software Engineering Notes. Volume 41 Issue 1, 2016, pp. 1-5
[12]. Murphy C., Kaiser G., Arias M.: Parameterizing Random Test Data According to Equivalence Classes. Proc of the 2nd International Workshop on Random Testing (RT'07), Atlanta GA, Nov 2007, pp. 38-41
[13]. Murphy C., Kaiser G., Arias M.: A Framework for Quality Assurance of Machine Learning Applications. Columbia University Computer Science Technical Reports, New York, 2006
[14]. Murphy C., Kaiser G., Hu L., Wu L.: Properties of Machine Learning Applications for Use in Metamorphic Testing. Proc of the 20th International Conference on Software Engineering and Knowledge Engineering (SEKE), Redwood City CA, Jul 2008, pp. 867-872.
[15]. Zhang J., Wang Z., Zhang L., Hao D., Zang L., Cheng S., Zhang Lu.: Predictive Mutation Testing. In Proc. of ISSTA'16, Saarbrücken, Germany, July 18–20, 2016, pp. 342-353

Создание тестовых данных для систем контроля и мониторинга рынка, содержащих встроенные алгоритмы машинного обучения

О. Москалёва <olga.moskaleva@exactprosystems.com>
А. Громова <anna.gromova@exactprosystems.com>
Exactpro, LSEG,
115088, Россия, Москва, 2-й Южнопортовый проезд, 20А, с4

Аннотация. Для правильной обработки информации о возможных сделках, проходящих через торговые платформы, выявления и предупреждения финансовых манипуляций, биржи устанавливают системы контроля и мониторинга данных. Эти системы получили широкое распространение за последние годы. Объемы и темпы торговли постоянно возрастают, увеличивается число разновидностей ценных бумаг. Финансовые регуляторы предъявляют все новые требования к торговым платформам. Из вышесказанного следует, что современные финансовые информационные системы в ближайшие годы будут продолжать совершенствоваться и активно использовать средства машинного обучения. Поэтому в последние несколько лет системы контроля и мониторинга рынка начинают внедрять модули интеллектуального анализа транзакций. Таким образом, интеллектуальные системы контроля и мониторинга рынка требуют усовершенствования подходов к их тестированию. Это связано с тем, что методы интеллектуального анализа данных формируют свои собственные зависимости между переменными. Тестовые сценарии должны разрабатываться таким образом, чтобы ожидаемый результат был понятен и предсказуем. Очевидно, что стандартные методы тестирования требуют модернизации. В представленной статье рассмотрены особенности современных интеллектуальных информационных систем, а также особенности их тестирования. В данном исследовании был разработан прототип модуля классификации финансовых манипуляций. Также предложены тестовые сценарии, позволяющие тестировать разработанный прототип. Данные сценарии состоят из нескольких типов, основанных на методологии классов эквивалентности. Разделение на классы эквивалентности было выполнено после анализа реальных данных. Прототип был протестирован с помощью выше обозначенных сценариев. Предложенный метод позволил выявить недостатки модуля классификации финансовых манипуляций.

Ключевые слова: тестовые данные; классы эквивалентности; системы контроля и мониторинга рынка; машинное обучение.

DOI: 10.15514/ISPRAS-2017-29(4)-18

Для цитирования: Москалёва О., Громова А. Создание тестовых данных для систем контроля и мониторинга рынка, содержащих встроенные алгоритмы машинного обучения. *Труды ИСП РАН*, том 29, вып. 4, 2017 г., стр. 269-282 (на английском языке). DOI: 10.15514/ISPRAS-2017-29(4)-18

Список литературы

[1]. FCA (financial conduct authority) (online). Доступно по ссылке: https://handbook.fca.org.uk/

[2]. SEC (Securities and Exchange Commission) (online). Доступно по ссылке: https://www.sec.gov/

[3]. FINMAR Financial Stability and Market Confidence Sourcebook (online publication). Доступно по ссылке: https://handbook.fca.org.uk/handbook/FINMAR/

[4]. Cao L., Ou Y., Yu P.: Detecting Abnormal Coupled Sequences and Sequence Changes in Group-based Manipulative Trading Behaviors. In Proc. of KDD'10, Washington, DC, USA, July 25–28, 2010, pp. 85-93

[5]. Donoho S.: Early Detection of Insider Trading in Option Markets In Proc. of KDD'04, Seattle, Washington, USA, August 22–25, 2004, pp. 420-429

[6]. Luo C., Zhao Y., Cao L., Ou Y., Zhang C.: Exception Mining on Multiple Time Series in Stock Market. In Proc. of International Conference on Web Intelligence and Intelligent Agent Technology, IEEE/WIC/ACM, 2008, pp. 690-693

[7]. Nasdaq and Digital Reasoning Establish Exclusive Alliance to Deliver Holistic Next Generation Surveillance and Monitoring Technology (online publication). Доступно по ссылке: http://www.digitalreasoning.com/buzz/nasdaq-and-digital-reasoning-establish-exclusive-alliance-to-deliver-holistic-next-generation-surveillance-and-monitoring-technology.1884035, 23.02.2016

[8]. Ou Y., Cao L., Luo C., Liu L.:Mining Exceptional Activity Patterns in Microstructure Data. In Proc. of International Conference on Web Intelligence and Intelligent Agent Technology, IEEE/WIC/ACM, 2008, pp. 884-887

[9]. Ou Y., Cao L., Yu T., Zhang C.:Detecting Turning Points of Trading Price and Return Volatility for Market Surveillance Agents. In Proc. of International Conferences on Web Intelligence and Intelligent Agent Technology - Workshops, IEEE/WIC/ACM, 2007, pp. 491-494

[10]. Murphy C., Kaiser G., Arias M.: An Approach to Software Testing of Machine Learning Applications. Proc of the 19th International Conference on Software Engineering and Knowledge Engineering (SEKE), Boston MA, Jul 2007, pp. 167-172

[11]. Nautiyal L, Preeti: A Novel Approach of Equivalence Class Partitioning for Numerical Input. ACM SIGSOFT Software Engineering Notes. Volume 41 Issue 1, 2016, pp. 1-5

[12]. Murphy C., Kaiser G., Arias M.: Parameterizing Random Test Data According to Equivalence Classes. Proc of the 2nd International Workshop on Random Testing (RT'07), Atlanta GA, Nov 2007, pp. 38-41

[13]. Murphy C., Kaiser G., Arias M.: A Framework for Quality Assurance of Machine Learning Applications. Columbia University Computer Science Technical Reports, New York, 2006

[14]. Murphy C., Kaiser G., Hu L., Wu L.: Properties of Machine Learning Applications for Use in Metamorphic Testing. Proc of the 20th International Conference on Software Engineering and Knowledge Engineering (SEKE), Redwood City CA, Jul 2008, pp. 867-872.

[15]. Zhang J., Wang Z., Zhang L., Hao D., Zang L., Cheng S., Zhang Lu.: Predictive Mutation Testing. In Proc. of ISSTA'16, Saarbrücken, Germany, July 18–20, 2016, pp. 342-353