The Method of Classification of User and System Data Based on the Attributes

Igor Zikratov, Igor Pantiukhin, Anna Sizykh
ITMO University
Saint Petersburg, Russia
{zikratov, zevall}@cit.ifmo.ru, anya.sizykh@gmail.com

Abstract—**The paper presents method of classification of user and system data based on the attributes. The method is intended to solve the problem of data classification in the internal audit of computer equipment with a large volume of data stored on the media. The parametric space of informative features for the classification task is defined. To conduct internal audit of computer technologies in order to analyze the information security incidents we need data related directly with the user's action. While OS data are considered a priori as legitimate. With the growing volume of stored and processed information there is a need for accurate classification of user and system data to ensure that expert is analyzing only the information that can attribute itself with the information security incident. Thus, carrying out classification with high precision, it is possible to reduce the probability of the expert error during the internal audit of large amounts of data of operating system and application software. The method consists of several stages. The first stage involves the collecting of samples out of virtual machines running the Windows 10 operating system. Then the attributes of data stored on the hard disk are retrieved. Further, the received set of attributes is written to a "csv" file with a specific structure. On the final stage the informativity of features for further classification of files to user or system is determined. This method has the potential for development and showed its applicability in tasks of data classification to user or system. The feature informativity for accurate classification was identified. The comparison of three classification algorithms was carried out and the Naive Bayes classifier was selected. The method can be used not only in tasks of internal audit of computer equipment under the influence of information security violation threats, but for the purpose of computer forensics, which is investigation of information security incidents.**

I. Introduction

The amount of stored and processed information is growing every day [1]. Hardware is known to be used to commit various offences. In terms of investigation of these offences (incidents) and internal audit of media content, data volume increase becomes a problem. One possible solution is to classify data that relate directly to the operating system and user activity. This will reduce the volume of information explored by experts, examining only the data relevant to user activity and incidents. Thus there is a need for methods of user and system data classification based on the attributes from computer equipment.

In this article user data means data (information) that was operated, saved and changed by the user in the operating system in the computer device. System data is data of operating system and application software. Internal audit of computer technology means the media content investigation aimed at detection of relation of files from storage media to the offense (incident) in the computer device.

It is well known that files stored on storage media have various attributes that can relate either to the user or system attributes. Defining the informativity of each feature, we can classify the file from the media by attributes to the user or system. Getting attributes is possible using the appealing to files on the media with help of special software. There are various types of software to obtain the attributes from the files, which were used in our project to obtain set of attributes. The example of such software is freely redistributable hachoir-metadata [2]. The obtained output of attributes from each file on the media is written to the file in "csv" format.

After obtaining the attribute from data on the media, it is necessary to classify the received attributes in the "csv" file and determine the informativity of each attribute. The informativity of features is determined by the method of Shannon. Shannon's method was selected as the most appropriate because of the need to evaluate information of the features of the weighted average amount of information per different grades of feature. Shannon's method allows determining the feature informativity involved in the recognition of an arbitrary number of object classes. In contrast to the cumulative frequency method and Kullback method that are used to determine the informativity indication, which is involved in the recognition of only two classes of objects.

The experiment using the proposed method was conducted, conclusions about the applicability of this method in the tasks of user and system data classification are made. Parametric space of informative features pointed on further improving of the proposed method is defined.

II. Attributes

A. About Attributes

Since the classification of the object is the process of recognition of its belonging to any class, then the task of classification becomes a mathematical problem of sample recognition.

Usually the task of object classification (recognition) is the following: in consideration of n-dimensional feature space

$\{X_i\}$, where $i = 1,2,\ldots,n$, each j-th $(j = 1,2,\ldots,m)$ is an object in this space is represented by a point with coordinates $x_{1,j}, x_{2,j}, \ldots, x_{n,j}$ and every class of objects is point set. To classify an unknown object, that is to recognize an image, means to determine which class the object belongs to, based on the analysis of the meanings of its features.

Using data stored on the hard disk it is possible to get different attributes (characteristics) that are used in the classification of user and system data. An example of such attributes is given in Table I.

TABLE I. EXAMPLES OF ATTRIBUTES

№	The attribute
1	Full path to file
2	File name
3	File size
4	File extension
5	File type
6	Right to file
7	Author
8	File creation data
9	The last file modification date
10	The last file access date
11	File signature/magic number
12	Hash function
13	Checksum
14	MIME-type

B. The process of obtaining attributes

Each operating system has its own peculiarities of attributes obtaining, and each file type has its own set of attributes. To obtain these attributes were used third-party open-source library Hachoir and its components. Hachoir is a versatile platform for manipulating a binary file, written in Python, operating system independent and having many text/graphic user interfaces. Despite the fact that it contains several functions to edit the files, it is usually designed to study the existing files. Hachoir supports more than sixty file formats. The recognition of the file format is based on the headers and footers in a disk image of the file. It has a fault tolerant parser that is designed to handle truncated or erroneous files. Hachoir concludes the following format [2], [3]:

$ hachoir-metadata video.avi
Common:
- Duration: 1 hour 38 min 4 sec
- Image width: 576
- Image height: 240
- Frame rate: 25.0 fps
- Bit rate: 989.9 Kbit/sec
- Producer: Nandub v1.0rc2
- Comment: Has audio/video index (5.7 MB)
- MIME type: video/x-msvideo
- Endian: Little endian
Video stream:
- Image width: 576
- Image height: 240
- Bits/pixel: 24
- Compression: XviD MPEG-4 (fourcc: "xvid")
- Frame rate: 25.0 fps
Audio stream:
- Channel: stereo
- Sample rate: 44.1 kHz
- Compression: MPEG Layer 3
- Bit rate: 128.0 Kbit/sec

There are cases when Hachoir library cannot parse the attributes of some files. In this case it is necessary to process separately each of these files using the Linux system utilities and regular expressions.

The obtained set of attributes from the media is stored in a "csv" file that contains values for each processed file. The CSV file has a text format, which is designed for convenient presentation of tabular data. Each line in the file is one row in the table. Values of particular columns are separated by a special symbol. It is convenient to use symbol of the semicolon ";" for subsequent analysis.

III. THE DEFINITION OF FEATURES INFORMATIVITY

For the file classification it is essential to obtain and analyze some features of this object (file). Such features are called informative features. The informative feature is information, useful for particular purpose, obtained from the source information. However, these are not always equivalent to achieve a specific purpose, therefore, searching and selection of enough informative features is necessary for the unambiguous files classification. To understand the meaning of "enough informative", we need to introduce the concept of informativity of the feature.

The informativity is how this feature characterizes the relation of the object to the system files class, that is, how it determines the correct classification and recognition result.

In this work the information approach for estimating the informativity is used, whereby the information tag is considered as a significant difference between classes of images in the feature space. If recognition of the object it needs to be categorized into one of several classes then such significant differences can be considered as the difference of the feature distribution probability constructed from samples of the compared classes.

An assessment of informativity is the value of $I(x_j)$, which is the area of the distribution of feature xi, and is not common with the area of another distribution of the same feature.

It is necessary to determine the measurement scale for each feature. For example, for the file type it is the item scale, which presents classes: Application, Archive, Audio, Example, Hidden, Image, Message, Model, Multipart, None, System, Text, Video.

The features for which it is impossible to make a finite number of classes in nominative scale (e.g., hash or date, time) it is essential to apply the ordinal scale.

To estimate the informativity of some feature it is necessary to operate with digital values and, therefore, to count

particulars. For example, for the "File type" attribute in the sample after.csv, we have identified classes of nominative scale and for each class we calculated the relative frequency. The counting result of the particulars for all samples is presented in Table II.

The files with the samples are formed on the basis of the file features. The value collecting is implemented using software code of high-level language "Python", which recursively goes through all elements of the storage device and record the values in the specified file:

- before.csv - file with attributes from freshly installed OS on the storage device without official updates of the system developers.
- after.csv - file with attributes with freshly installed OS with updates from the official developer.
- soft.csv - file with the attributes of user and system data from the storage device.

TABLE II. PARTICULARS FOR ALL SAMPLES OF THE "FILE TYPE" ATTRIBUTE

Type	Frequency (before.csv)	Frequency (after.csv)	Frequency (soft.csv)
Application	21083 (a1)	23573 (a2)	25314 (a3)
Archive	223 (b1)	229 (b2)	239 (b3)
Audio	366 (c1)	375 (c2)	3798 (c3)
Hidden	6 (d1)	6 (d2)	8 (d3)
Image	11200 (f1)	13241 (f2)	14644 (f3)
None	27367 (g1)	31493 (g2)	34751 (g3)
System	35849 (h1)	42710 (h2)	42748 (h3)
Text	3279 (k1)	3588 (k2)	4483 (k3)
Video	56 (l1)	66 (l2)	76 (l3)
Sum	**99429**	**115281**	**126061**

- Gradation for class 1: a1, b1, c1, d1, f1, g1, h1, k1, l1.
- Gradation for class 2: a2, b2, c2, d2, f2, g2, h2, k2, l2.
- Gradation for class 3: a3, b3, c3, d3, f3, g3, h3, k3, l3.

For the feature "File size" the ordinal scale can be used, if the size fits the certain period, then it is assigned the appropriate grade based on the criteria in Table III.

TABLE III. GRADES FOR "FILE SIZE" ATTRIBUTE

X <= 2^5(32)	A (the lowest)
2^5(32) < X <= 2^10(1024)	B
2^10(1024) < X <= 2^15(32768)	C
2^20(32768) < X <= 2^20(1048576)	D
2^25(1048576) < X <= 2^25(33554432)	E
2^25(33554432) < X	F (the highest)

For the "Path" attribute the items scale is used. After processing the data the comparison of the feature with the following classes of path lengths (number of folders) is performed: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, >10.

For the "File signature/magic number" attribute the nominative scale is used. First, the obtained data is processed. If the attribute matches the parameter from the list of the system files, then it is assigned rank 2. If it matches the value from the list [4], then the rank is equal to 1, in all other cases it equals to 0 (0-1-2).

The scale of the "Date" attribute is nominative. While attribute collecting file creation date, the last file access date the file last file modification date are calculated. Each value is saved in the format Tue Mar 31 05:34:42 2014. The obtained data are processed and in accordance with that the following ranking taking place (0-1-2-3):

- If creation date matches the date of the last change, it is assigned rank 3.
- If creation date matches the date of last access, it is assigned rank 2.
- If the date of the last access coincides with the last modification date, it is assigned the rank 1.
- If the values do not match, it is assigned the rank 0.

For "Hash" attribute the nominative scale is used. While collecting attributes the hash sum of the file is calculated. Each value is stored as a hex number of 32 characters. The data obtained are processed, in accordance with which the following ranking is taking place:

- If the value from file before.csv matches the value from file after.csv, it is assigned rank 1.
- If the value from file before.csv does not match the value from file after.csv, it is assigned rank 0.
- If the value from file before.csv matches the value of file soft.csv, it is assigned rank 1.
- If the value from file before.csv does not match the value of file soft.csv, it is assigned rank 0.
- If the value from the file after.csv matches the value of file soft.csv, it is assigned rank 1.
- If the value from the file after.csv does not match the value of soft.csv, it is assigned the rank 0.

One-way of informativity estimating is known from information theory. It is Shannon method, which proposes to estimate the informativity as the weighted average amount of information per different grades of feature. "Information" in information theory is the value of the reduced entropy. Shannon methods operate on the probability, so the sample sizes of features observations in three recognizable classes can be discerned, in contrast to the method of cumulative frequencies, as it operates on the frequencies and the sample size of feature observations must be the same in recognized classes [5].

Thus, the informativity of the j-th feature is calculated by the to equation:

$$I(x_j) = 1 + \sum_{i=1}^{G} \left(P_i \sum_{k=1}^{K} \left(P_{i,k} * \log_k P_{i,k} \right) \right),$$

where G is the number of feature gradations, K is the number of classes.

P_i is the probability of i-th gradations of the feature by the equation:

$$P_i = \frac{\sum_{k=1}^{K} m_{i,k}}{N},$$

where $m_{i,k}$ is the appearance frequency of i-th gradations in that class, N is the total number of observations.

$P_{i,k}$ is the probability of i-th gradations of the feature in k-the class by the equation:

$$P_{i,k} = \frac{m_{i,k}}{\sum_{k=1}^{K} m_{i,k}} .$$

Shannon's method gives the estimation of informativity as a normalized value that varies from 0 to 1. Thus, the informativity of the feature defined by this method can be evaluated in absolute terms: closer to 1 is high; closer to 0 is low [6].

Estimated attribute informativity by the method of Shannon is shown in Table IV.

TABLE IV. THE ESTIMATED INFORMATIVITY

Technology	Feature	Informativity	Scale type
md5	Hash	0,0265103309	Nominative scale
File signature	Magic number	0,0046775854	Nominative scale
Full path (/1/vv/)	Folders amount (2)	0,0124959879	Ordinal scale
Size	Size	0,0062970409	Ordinal scale
Creation date The last modification date The last access date	Date (date match)	0,4814620916	Nominative scale
Extension	File type	0, 0102158879	Item scale

IV. METHOD

The method is divided into several stages. The first stage involves the collecting of samples with virtual machines running the operating system Windows 10 in amount of 4 pieces. The hard disk drive (HDD) memory dump was obtained from this systems: operating system was installed on the virtual machine with the VirtualBox software, the "hard disk" of this machine was stored in the file with "vdi" extension. This "vdi" file was converted using a command line program VBoxManage, which is contained in the standard installation VirtualBox kit, to the file with the "img" extension with the following command:

VBoxManage clonehd PathToVdiFile PathToImgFile -format raw,

where the "*clonehd*" is an option responsible for copying information from the memory dump. "PathToVdiFile" and "PathToImgFile" are the path to the hard disk of the virtual machine with "vdi" extension and the path where it save the copy of the "vdi" file to the file with the "img" extension. Correspondingly, it is followed by creation of the key to the copying process.

The resulting file is bit-wise hard disk image of the virtual machine. This ".img" file mounted as a unit on the personal computer running the Fedora 23 operating system using the "*mount*" command.

In the second stage the set of attributes from the media is retrieved, the result is saved and on this basis the standard database is formed. It consists of attributes collected from freshly installed operating system; for the correctness of processing of the received results it is necessary to disconnect from the Internet. The result is written to the file before.csv. Then the system is updated by means of the official repositories, and attributes from the media are collected. The result is written in file after.csv. Later the attributes from the operating device with the user data are obtained, the generated base stored in soft.csv. All databases should be formatted in accordance with the following requirements:

- The CSV file format.
- Separator " ; " .
- No invalid names in the structure of the table.
- No invalid column values in the structure of the table.

For classification tasks the following attributes of files were selected, namely:

- Hash.
- Magic number.
- Full path.
- Size.
- Creation date.
- The last modification date.
- The last access date.
- File type.

As the hashing algorithm we use a 128-bit md5 algorithm. The hashing means converting the input data array by given algorithm in a bit-string of fixed length. Obtained result of computing is presented in the hexadecimal counting system, which is called the hash sum.

Because the file extension may be absent, for example in the Unix systems, and it can be changed thereby hiding the identity to a specific data type, it is essential to use magic numbers. The magic number, or the file signature, is an integer constant that is used to uniquely identify a resource or data. To have a known signature database makes it possible to compare the obtained value with the existing and on this basis to carry out the classification. However some files can correspond with several extensions, for example, the new Microsoft formats (docx, xlsx) since they essentially are zip archives. That can lead to incorrect results.

The file size in the computer technology is commonly understood as the volume of data on the storage device. The basic measurement unit of this feature is byte.

The third stage is the classification based on the selected attributes and the analysis, which determines which group a given file belongs to. To achieve this the classification algorithm is applied. Two "csv" files containing the attributes are given to input. One of these files is training and includes data after the operating system upgrading (the training sample) and the second includes data from the device, which contains user and system data (test sample). The training sample comes to input of the classification algorithm, then the degree of conformity of files from the device to certain classes is determined.

V. EXPERIMENT DESCRIPTION

RapidMiner software is used for data mining. Each experiment is described in the form of superpositions of arbitrary number of arbitrarily nested operators. Due to visual graphics interface it is easy to show the results [7].

Using RapidMiner the model of algorithm trained on input data from the after.csv file is generated. The file name value is considered as "identifier", and the file type is considered as "label". Testing is conducted on the dataset from soft.csv file. 4 experiments on different data samples were carried out in order to verify the selected method and to determine the accuracy of the classification. The results of the calculations are presented in Table VI.

To identify the best accuracy parameter it was decided to use the following classification methods:

- Decision tree.
- k-NN algorithm.
- The Naive Bayes classifier.

The method of decision trees is one of the most popular methods for solving classification and prediction tasks. Sometimes this method of Data Mining is also called decisive rule trees, classification and regression trees. If the dependent, i.e. the target variable, takes discrete values, using the method of decision tree the problem of classification can be solved. In the simplest form the decision tree is a way of representing rules in a hierarchical, coherent structure. The basis of this structure is the answering "Yes" or "No" to several questions [8].

k-NN algorithm is k Nearest Neighbours. It is one of the simplest metric classifiers based on the assessment of similarity of objects. Classified object belongs to the class that owns the nearest objects of the training sample [9].

Naive Bayes classifier is a simple probabilistic classifier based on applying Bayes theorem with strong (naive) independence assumptions. It assumes that the presence (or absence) of a certain characteristic (attribute) of the class does not correlate with the presence (or absence) of any other feature. Even if the functions depend on each other or on other features, a Naive Bayes classifier considers all of these properties can independently contribute to the probability of the fact that the document belongs to one or another class [10].

The advantage of Naive Bayes classifier is that it requires only a small amount of training data to estimate average values and variables dispersion that are necessary for classification. Because of the fact that independent variables are assumed, the only variables dispersion for each label must be determined but not the entire covariance matrix.

The experiments showed the accuracy calculated by two methods:

- "Accuracy".
- Cohen's kappa statistic.

"Accuracy" is calculated as the percentage of conformity and correctness of something in comparison with a true or absolute value and is determined by the equation:

$$Accuracy = \frac{P}{N} * 100\%,$$

where P is the number of documents for which the classifier made a correct decision, and N is the size of the training sample.

Cohen's kappa statistic is a measure of coherence between two categorical variables X and Y. It is used to assess the consistency between two evaluators, classifying n objects by s categories, as shown in Table V [11], [12].

TABLE V. THE EXAMPLE OF COHEN'S KAPPA STATISTICS CLASSIFICATION

	B_1	B_2	...	B_S	*Total*
A_1	n_{11}	n_{12}	...	n_{1S}	m_1
A_2	n_{21}	n_{22}	...	n_{2S}	m_2
...	...	...	...	...	...
A_S	n_{s1}	n_{S2}	...	n_{SS}	m_S
Total	n_1	n_2	...	n_S	n

The observed coherence between X and Y [13]:

$$P_o = \frac{\sum_{i=1}^{n} n_{i,i}}{n}.$$

The expected probability of random consistency [13]:

$$P_e = \frac{\sum_{i=1}^{n} n_i m_i}{n^2}.$$

Cohen's Kappa statistic is defined as [13]:

$$k = \frac{P_o - P_e}{1 - P_e}.$$

If the evaluators fully agreed, then $k = 1$. If $k > 0,75$, then the coherence is considered as high, if $0,4 < k \le 0,75$ then coherence is good, otherwise coherence is poor [14], [15].

TABLE VI. ACCCURACY AND COHEN'S KAPPA STATISTICS OF EXPERIMENTS

Algorithm	"Accuracy"	Cohen's kappa statistic
k-NN	65,40% + /- 1,76%	0,0534 +/- 0,026
Decision Tree	34,13% +/- 0,00%	0,000 +/- 0,000
Naive Bayes	94,16% + /- 0,06%	0,923 +/- 0,001

As can be seen from Table VI, the Naive Bayes classifier shows the best accuracy value.

The values of "Precision" and "Recall" are convenient to calculate using confusion matrix [16]. With a relatively small number of classes (no more than 100-150), this approach allows to visualize the performance of the classifier.

The confusion matrix is a matrix of size N by N, where N is the number of classes. Columns of the matrix are reserved for expert decisions and lines are reserved for decisions of the classifier [17]. In the process of classification of document from the test sample the number appearing at the intersection of the string class returned by the classifier and the column class, which the document actually belongs to, is incremented [18].

Using such matrix it is simple to calculate the "Precision" and "Recall" for each class. The "Precision" is equal to the ratio of the corresponding diagonal element of the matrix and sum of the entire row of class. "Recall" is the ratio of the diagonal element of the matrix and the sum of column of class [19]. Formally, they are calculated by the equations:

$$Precision_c = \frac{A_{c,c}}{\sum_{i=1}^{n} A_{c,i}}, \qquad Recall_c = \frac{A_{c,c}}{\sum_{i=1}^{n} A_{i,c}}.$$

	true NONE	true Application	true System	true Text	true Image	true Archive	true Audio	true Video	true Hidden	class precision
predicted NONE	**311739**	151	247	15	118	38	9	0	0	99.81%
predicted Application	31	**240033**	429	0	19	0	0	1	0	99.80%
predicted System	21	6	**426125**	1	0	0	0	0	0	99.99%
predicted Text	89	92	8	**38190**	0	0	0	0	0	99.51%
predicted Image	5	2	0	0	**136907**	0	0	0	0	99.99%
predicted Archive	47	55	0	10	20	**2318**	1	1	0	94.54%
predicted Audio	11415	2057	17	0	310	23	**23442**	59	0	62.81%
predicted Video	523	1890	135	14	765	1	0	**699**	0	17.36%
predicted Hidden	19940	7944	519	5730	5861	0	14488	0	**80**	0.15%
class recall	00.67 %	95.16 %	99.68 %	86.87 %	95.07 %	97.39 %	61.79 %	91.97 %	100.00 %	

Fig. 1. The confusion matrix for the Naive Bayes classifier

The accuracy of the system within a class (class precision) is the percentage of documents truly belonging to this class regarding all documents that the system ascribes to this class.

The completeness of the system within a class (class recall) is the proportion of documents belonging to the class regarding to all documents of this class in the test set found by the classifier [20].

As can be seen from Fig. 1, the classifier determines the majority of documents. The diagonal elements of the matrix are explicitly expressed; this means that selected attributes make it possible to accurately classify the data. But nevertheless, in some classes (Hidden, Video) the classifier accuracy is low.

VI. Conclusion

The method of classification of user and system data based on the attributes demonstrated its applicability in tasks of internal audit of computer equipment. The use of described method allows defining the information content of features and on this basis to calculate the accuracy of data classification to user and system. The experiment was conducted by using the proposed method, in which informativity was calculated and the most significant attributes were selected. The results can be used to investigate various information security incidents and to analyse the content of data storage device.

The proposed method is not final version. In continuation of the work the following objectives are outlined: development of an algorithm which takes into account the interdependence of the parameters (attributes) and improving the method of classification of files to system and user. The experimental results show that this method has the potential to develop and can be used not only in classification problems of user and system data when performing an internal audit but also in other tasks of computer forensics.

References

[1] The Economist Newspaper official website, Data, data everywhere, Web: http://www.economist.com/node/15557443.

[2] Inc. Atlassian Bitbucket official website, hachoir/hachoir-metadata, Web: https://bitbucket.org/haypo/hachoir/wiki/hachoir-metadata.

[3] ForensicsWiki website, Hachoir, Web: http://forensicswiki.org/wiki/Hachoir.

[4] G.C.Kessler website, File signatures table, Web: http://www.garykessler.net/library/file_sigs.html.

[5] C.E.Shannon and W.Weaver, *The Mathematical Theory of Communication*. Urbana, IL: University of Illinois Press, 1949.

[6] E.V.Dashutina and E.V.Merkulova, "Design SCS diagnosis of cardiovascular disease", Web: http://www.uran.donetsk.ua/~masters/2012/fknt/dashutina/library/article1.htm.

[7] RapidMiner official website, RapidMiner Studio Manual Web: https://rapidminer.com/wp-content/uploads/2014/10/RapidMiner-v6-user-manual.pdf.

[8] J.R.Quinlan, "Simplifying Decision Trees", *International Journal of Man-machine Studies*, vol.27, no.3, 1987, pp. 221-234.

[9] N.S.Altman, "An introduction to kernel and nearest-neighbor nonparametric regression", *The American Statistician*, vol.46, no.3, 1992, pp. 175–185.

[10] A.McCallum and K.Nigam, "A comparison of event models for naive bayes text classification", *AAAI Workshop on learning for Text Categorization,* 1998, pp. 41-48.

[11] Wikipedia official website, Cohen's kappa, Web: https://en.wikipedia.org/wiki/Cohen%27s_kappa.

[12] J.Cohen, A coefficient of agreement for nominal scales. *Educational & Psychological Measurement,* vol.20, no.1, Apr.1960, pp. 37-46.

[13] J.Cohen, "Weighted kappa: nominal scale agreement with provision for scaled disagreement or partial credit", *Psychological Bulletin*, vol.70, no.4, Oct.1968, pp. 213-220.

[14] A.J.Viera and J.M.Garrett, "Understanding Interobserver Agreement: The Kappa Statistic", *Family Medicine*, vol. 37, no. 5, May 2005, pp. 360-363.

[15] L.V.Grayer and O.A.Arhipova, "Lecture 8. Nonparametric tests of independence. Correlation analysis", Web: https://compscicenter.ru/media/slides/math_stat_2014_spring/2014_03_28_math_stat_2014_spring_1.pdf.

[16] Wikipedia official website, Confusion matrix, Web: https://en.wikipedia.org/wiki/Confusion_matrix.

[17] Kai Ming Ting, "Confusion Matrix", *Encyclopedia of Machine Learning*, 1st Edition, 2011, p. 209.

[18] Václav Hlaváč, "Classifier performance evaluation", Web: http://cmp.felk.cvut.cz/~hlavac/TeachPresEn/31PattRecog/13ClassifierPerformance.pdf.

[19] Zalán Bodó, "Evaluation of text categorization system", Web: http://www.cs.ubbcluj.ro/~zbodo/write/eval.pdf.

[20] Wikipedia official website, Precision and recall, Web: https://en.wikipedia.org/wiki/Precision_and_recall.