The Method of Elf-Files Identification Based on the Metric Classification Algorithms

Igor Zikratov, Igor Pantiukhin, Irina Krivtsova, Nikita Druzhinin
ITMO University
Saint Petersburg, Russia
{zikratov, zevall, ikr}@cit.ifmo.ru, kleverteo@gmail.com

Abstract— **When performing the internal audit of computer equipment an important problem is to identify the elf (executable and linkable format) files stored on the investigated hard drive. To solve this problem, we propose a method of identification of unknown elf-files based on the metric classification algorithms. The method consists of three stages. On the first stage the preparation of the training sample by the disassembly of each file and submitting it in the form of an ordered set of the 118 elements is implemented. Each of these elements is the frequency of occurrences of the 118 most commonly used commands in the assembler code. Each program in the sample is represented by several sets, corresponding to different versions or operating systems in which this software is installed. Then, the Minkowski metric of each sample file and identifiable file is calculated. For the method of potential functions the selection of the reference elements of each set is obtained. On the third stage using the metric classification algorithms we evaluate affiliation of the identifiable file for a particular program from the sample. To approbate proposed method the experiment with the use of this method was conducted; results showing the accuracy of identification of elf-files was equal to 89,60% were obtained. The results indicate that this method is applicable in problems of identification of elf-files while conducting the internal audit of computer equipment. The advantages of the method are the accuracy of program identification regardless of the elf-files versions in the Linux operating systems. The ease of implementation of our method and the identification execution speed can be used not only in tasks of internal audit, but also in other tasks of computer forensics.**

I. INTRODUCTION

In nowadays world the development of computer technology is rapidly developing, especially in the information storing area. The volume of stored and processed information is growing from year to year. Computer equipment is known to be used not only for its intended purpose, but also for committing various crimes. Thus, on the basis of growth of volumes of stored information, there is a need for modern methods of identification of executable files. Identification of executable files allows classifying files its relation to the offence in the computer technology area, thus reducing the probability of the expert error in the study of large amounts of data obtained from media storage. In addition, identification of executable files will solve the problem of classification of the user (created by user in the operating system) and the system (directly related to the operating system) data. In most cases, the purpose of internal audit is to study precisely user data and programs; in this case, these operating systems data are not considered because they are perceived as legitimate.

Internal audit of computer equipment in this article means the realization of the media content research in order to find relation of files from media storage with the offense in the computer technology area.

During recent years various Linux operating systems are gaining increasing popularity. In such operating systems the main format of executable files is elf (executable and linkable format). Previously, the problem of identification of elf-files was solved in the following works: [1], [2], [3], [4]. However, in these studies the identification of elf-files based on the analysis of frequency distributions of machine codes was not performed.

The proposed method compares the frequency distributions of machine codes of identifiable program with many frequency distributions of programs from the template archive based on the three metric classification algorithms:

- Method of potential functions choosing the reference element by applying the equation for calculating the center of mass.
- Method of potential functions choosing the reference element by means of STOLP algorithm and k-nearest neighbors method.
- Method of potential functions choosing the reference element using the k-nearest neighbors method.

Further the comparison of these algorithms is conducted; coefficients and indicators of identification accuracy of executable elf-files are defined.

The experiment using the proposed method is carried out; conclusions regarding the applicability of this method for identifying executable files while internal audit of computer equipment are made.

II. THE ARCHIVE TEMPLATE FORMATION

For the implementation of the identification process of elf-files it is necessary to use a pre-built template archive, which will be compared with the elf-file. For using of metric classification algorithm it is required to submit the training sample, that is plurality of source programs files, in the form of numerical vectors in the Euclidean space.

We present the template of an executable file of each source program in the form of an ordered set of frequencies of machine codes. The construction of template archive is performed in several stages:

1) The formation of the training sample (TS).

2) Disassembling each file and counting frequencies of occurrence of the 118 most common machine codes (add, mov, jmp, etc.).

III. THE FORMATION OF THE TRAINING SAMPLE (TS)

To build the training sample it is necessary to analyze a certain amount of executable files identified with this program, but different in their versions or distributions, on which they are installed. Based on this analysis, templates for different programs forming the archive are created. A training sample is represented in the following form:

$$TS = \{v_1, v_2, \ldots, v_M\},$$

where v_i is the sampling of various programs; M is the number of different programs;

$$v_i = \{f_1, f_2, \ldots, f_n\},$$

where f_j is various versions of the i-th program; n is the number of files in the sample v_i.

Then there is the disassembly of each file f_j and counting of frequencies of occurrence of the most commonly used machine codes.

$$T(f_j) = \{a_1, a_2, \ldots, a_{118}\}, \quad j = 1 \div n \qquad (1)$$

where a_k is the frequency of occurrence of the k-th machine codes in the file.

```
aircrack 6
0 0 0 0 318 4815 1211 1243 0 12 6 5 8 1088 0 0 0 0 0 66 2 0 15 22 251 56 59 8 0 4 86 179
0 0 0 0 256 4180 1195 1242 0 12 6 5 8 1082 0 0 0 0 0 66 2 0 17 22 237 30 57 8 0 4 81 175
0 0 0 0 316 4784 1134 1221 0 17 7 6 10 1046 0 0 0 0 0 31 2 0 16 22 252 33 51 8 0 5 88 163
0 0 0 0 270 5176 1152 1187 0 17 8 5 9 1013 0 0 0 0 0 49 2 0 24 24 253 28 85 8 0 4 86 159
13 5 20 10 322 2740 895 1077 1 10 9 7 9 949 0 0 1 9 10 203 3 0 13 23 257 30 373 11 6 8 65
0 0 0 0 160 3936 834 885 0 13 6 6 8 989 0 0 0 0 0 22 3 0 16 25 214 32 38 10 0 6 50 126 14
calendar 2
0 0 0 0 68 2401 92 256 0 6 0 0 2 166 0 0 0 0 0 6 0 0 2 0 114 5 5 1 0 0 23 31 44 13 0 0 20
0 0 0 0 68 2401 92 256 0 6 0 0 2 166 0 0 0 0 0 6 0 0 2 0 114 5 5 1 0 0 23 31 44 13 0 0 20
gftp-gtk 6
0 0 0 0 629 14344 2109 6464 0 7 119 6 4 1645 0 0 0 0 0 71 6 0 16 3 961 94 78 7 0 10 238 6
8 15 24 25 1107 10271 2120 6784 0 31 197 2 40 1450 0 0 0 19 134 717 20 0 21 7 975 82 1379
0 0 0 0 1591 13952 2700 6632 0 35 124 8 7 1688 0 0 0 0 0 127 5 0 14 6 962 41 106 6 0 10 2
0 0 0 0 1069 13738 2359 6663 0 58 142 4 1 1672 0 0 0 0 0 119 5 0 19 4 977 53 110 13 0 6 2
0 0 0 0 1066 14680 2407 6576 0 35 114 3 2 1621 0 0 0 0 0 116 5 0 23 9 972 46 125 5 0 4 24
30 34 47 42 1293 10024 2284 6917 0 37 231 18 33 1509 0 0 1 32 149 841 23 0 35 9 1074 149
gimp 4
0 0 0 0 30523 235434 42293 111337 0 627 730 273 251 23521 0 0 5 0 0 1478 145 0 758 233 24
0 0 0 0 18958 250305 35335 110647 0 634 820 308 254 21675 0 0 5 0 0 1514 194 0 813 239 24
0 0 0 0 11860 212747 28490 92867 0 701 836 455 253 17944 0 0 5 0 0 1499 138 0 784 227 20
0 0 0 0 29334 237040 41320 114369 0 763 765 275 247 24477 0 0 5 0 0 1505 132 0 761 225 25
```

Fig.1. The example of training sample database

Fig.1 presents an example of training sample. The first line contains the name of the program and the number of files n in the sample for this program. Then n arrays $T(f_j)$ containing 118 elements using the equation (1) is presented. The order of commands in the sample is conditional, but it should be the same for all involved file in the classification. The alphabetical order of commands was selected and conducted in this paper.

IV. THE IDENTIFICATION PROCESS

Identification occurs using the metric classification algorithms: algorithms based on the computation of estimates of the objects similarity. To formalize the notion of similarity in the object space a function of the distance $\rho(x, y)$ is introduced, in most cases it is a metric. Metric classification algorithms are based on comparing the identified object with all objects from the training set following a specific rule.

The training set (TS) is a set of M groups of objects associated with file from training sample of arrays $L(f_i)$, each of it represents a point or vector in the 118-dimensional space. Classification is based on the compactness hypothesis: if the measure of object similarity is introduced successfully, then similar objects are more often surrounded by objects of its class than of another class. In this case, the boundary between classes has a fairly simple form, and classes form a compact localized areas in the object space.

As the array $T(f_i)$ is numerical, we use the Minkowski metric for calculating the distance between identifiable object u and the i-th closest object to u $x_u^{(i)}$.

$$\rho(u, x_u^{(i)}) = \left(\sum_{i=1}^{118} \left| u - x_u^{(i)} \right|^q \right)^{1/q} \qquad (2)$$

where u is the identifiable file, $x_u^{(i)}$ is the file from archive.

Since the choice of adequate metrics is complex and the least studied problem, we investigate how the percentage of correct answers of the classifier depending on the value of the parameter q changes.

V. THE METHOD OF POTENTIAL FUNCTIONS

We consider the classification algorithm based on the method of potential functions. While the classification the object u is checked for proximity to objects from the training set. It is believed that the objects from the training set "loaded" to their class, namely the affiliation of this file to the certain program and the "importance" measure of each depends on its "charge" and distance from the classified object [5].

For the point of 118-dimensional space corresponding the identified object we calculate the potential created by the other points' field. The point belongs to group, which has stronger potential. In the general case, the algorithm based on the method of potential functions [6] is showing by the following equation (3).

$$a(u) = \arg\max_{m \subseteq M} \sum_{i=1}^{N} \left[x_{i,u} = m \right] \gamma\left(x_u^{(i)}\right) K\left(\rho\left(u, x_u^{(i)}\right)\right) \qquad (3)$$

where N is the sample size, m is the subset of the training sample objects joined in a single class, M is class plurality,

$\gamma(x_u^{(i)})$ is a parameter that sets the "charge", that is, the degree of object importance, its value will be equal to 1 as all objects are the same, $K(r)$ – function of calculation of the potential decreasing with growth of argument, $\rho(u, x_u^{(i)})$ is the Minkowski distance between the object u to the i-th nearest to u object $x_u^{(i)}$, calculated by the equation (2).

Next, we need to take only one standard point from each class of training set. Such choice is necessary because the number of points in each group may be different, and, therefore, the contribution of each group in its own capacity depends on the number of points that will have a negative impact on the classification, furthermore, if the objects of the same class closely surrounded by objects of another class, removing them from the training set will not affect the percentage of correct answers of the classifier. Moreover, the result of the each class standard object selection increases the classification quality, reduces the volume of stored and processed data and calculation time.

VI. THE STANDARD POINT

The standard point selection can be obtained in two ways.

A. Using the center of mass of the point group

Assuming that each point that belongs to the class of a specific program and in our 118-dimensional Euclidean space is a point with some mass. Then the program represented by the set of files from the training set is a system of these masses. In this case, the center of mass of the system will be the standard point.

Two ways of the center of mass calculation are available: either the entire mass is concentrated at the tops, or it is concentrated evenly over the area of the polygon with tops at these points.

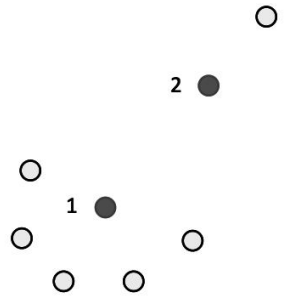

Fig. 2. Two ways of calculation of the center of mass

In Fig.2 point 2 is the center of mass of the polygon with the mass distributed over the area. But for the considered task, this answer is incorrect because the point is not in the place of greatest points grouping. This top point is likely to be noise or emissions and it can badly affect the quality of the classification.

Another result is shown by the method of calculating the center of mass of the points group where the mass is concentrated on each of them and it is point "1" in the Fig.2. The center of mass for this case is calculated by (4).

$$\vec{r_c} = \frac{\sum_i \vec{r_i} m_i}{\sum_i m_i} \quad (4)$$

where r_i is a coordinate vector for each point. The mass m_i for each point is taken as one. In other words, the coordinates of our standard point are the arithmetic mean of the coordinates of the vector of each point of the sample.

Each class received one reference point, thus, the number of reference points is equal to M. To calculate the potential we choose the function decreasing with growth of argument, for example, $K(\rho) = 1/\rho^2$. Therefore, for identifiable file specified by point u and M class reference points the calculated measure of the importance (weight) of the object x_i from the training set is described by the equation:

$$a(u) = \arg\max_{m \subset M} \left[x_{(\ni) i;u} = m \right] \frac{1}{\rho\left(u, x_{(\ni)u}^{(i)}\right)^2},$$

where $i = 1 \ldots M$ and $x_{(\ni)u}^{(i)}$ – the closest to u standard point belonging to the m class.

B. The STOLP algorithm implementation [7]

In the method based on the equation for calculating the center of mass mentioned above, the standard object might not belong to the object space of the training sample. With the help of STOLP algorithm we can find the standard objects among the members of the training sample.

Typically, the training set objects are not equal. Among them typical representatives of class (standards) may exist. If the classified object is close to the standard, then most likely it belongs to the same class. Another one category of objects is peripheral or uninformative objects. They closely surrounded by other objects of the same class. If we drop them from the sample, it will not affect the quality of the classification. Finally, the sample can contain a certain amount of noise emissions that are objects, located in "stratum" of foreign class. Generally, its removal improves the quality of classification.

Consequently, it is essential to exclude from noise and uninformative objects sampling, keeping only the minimum sufficient number of standards. A heuristic algorithm STOLP excludes noise and uninformative objects from sampling.

To estimate the degree of object typicality we introduces the concept of indentation. For classification algorithm $a(u) = \arg\max_{m \in M} \Gamma_m(u)$ indentation is the value calculated by the equation:

$$M(x_i) = \Gamma_{m_i}(x_i) - \max_{m \in M \setminus m_i} \Gamma_m(x_i),$$

which shows the degree of object typicality. The highest indentation is inherent for standard objects, closely surrounded by objects of the same class. So, noises and emissions will have a negative indent and will be cut from the sample.

The distribution of indentation values in the sample gives useful additional information not only about individual objects, but also on the sample as a whole. If the bulk of facilities has positive indents, then the sample division can be considered successful. If the sample has too much negative indents, the hypothesis of compactness fails. If indent values are concentrated near zero, it is unlikely to be possible to conduct reliable classification of objects because too many objects are in the border zone.

The standard set sample Ω is based on a sample of TS and consist of objects with the maximum positive offset only. Trough this an algorithm $a(x_i,\Omega)$ is built.

We denote indentation of training set object x_i on the algorithm by $O(x_i,\Omega)$ regarding the given classification algorithm. A large negative indent indicates that object x_i is surrounded by objects of others classes, therefore, is an outlier. A large positive indent means that the object is surrounded by objects of the same class, that characterizes it as either the standard or peripheral [8].

As a result of the STOLP algorithm execution we extracted from the sample standard elements for each class. The algorithm does not require re-implementation, for any particular set of sample STOLP is calculated only once.

Therefore, the classification algorithm takes the following form:

$$a(u) = \arg\max_{\omega \subseteq \Omega} \left[x_{i;u} = \omega \right] \frac{1}{\rho\left(u, x_u^{(i)}\right)^2},$$

where Ω is the set of standards, $x_u^{(i)}$ is i-th closest to u element of the set Ω, belonging to class ω, $\rho(u,x_u^{(i)})$ – Minkowski metric, as in (2).

VII. K-Nearest Neighbors method

A. About algorithm

Nearest neighbor method is the simplest classification algorithm. Classified object u belongs to class m_i, which the closest training sample x_i object belongs to. For an arbitrary object u we place the objects in the training sample x_i in order of increasing the distance to u:

$$\rho\left(u,x_{1;u}\right) \le \rho\left(u,x_{2;u}\right) \le \ldots \le \rho\left(u,x_{N;u}\right).$$

We refer training set objects to such class that the k-nearest training sample objects belongs to. In general, the algorithm of nearest neighbors is represented by the equation (5).

The algorithm is:

$$a(u) = \arg\max_{m \subseteq M} \sum_{i=1}^{N} \left[x_{i;u} = m \right] w(i;u) \qquad (5)$$

where N is the sample size, and $x_{i;u}$ is i-th closest object to u, $w(i;u)$ is a weighting function that evaluates the degree of importance of the i-th neighbor to classify the object u, and characterizes the method of k-nearest neighbors.

Giving weight function by different ways we can obtain different variants of nearest neighbor method. For the k nearest neighbors $w(i;u) = [i \le k]$.

Thus, during the execution of this algorithm all the training samples are sorted using the selected metric filter out all but the largest k. Then a vote among them is produced: element u is related to such m class that attains the highest number of votes.

B. The choice of number of k-neighbors

Provided $k = 1$ nearest neighbor algorithm is a trivial variant that is sensitive to noise emissions: it gives the erroneous classification not only for the objects-emissions, but also on adjacent objects of other classes. When $k = N$, on the contrary, the algorithm is overly stable and degenerates into a constant. Thus, extreme values of k are undesirable. In the experiment, the values of the parameter k vary from 1 to 4, as in the sample the number of training files is not more than 4 for each class, and, therefore, values that are greater than 4 will not give the increase of correct answers of the classifier.

VIII. The results of the experiment

The proposed method was experimentally tested with 125 programs of training sample (a total of 559 files) and 125 programs of the test sample (125 files). 32- and 64-bit programs were classified separately.

The task consisted in identifying the test sample with the signature archive. The choice of the metric, namely the coefficient q value, plays a crucial role in the objects similarity establishment. So experiments were conducted with different values of q to identify dependency of its value on the number of correct classifier answers. The values of q were taken in range from 0,02 to 4.

Identification of files was carried out in the following sequence.

A. First step: signature archive compilation

For each training sample file is drawn up disassembling the signature of this file in the form of the set of 118 frequency values of each machine code in the format shown in Table I. In addition, Table II shows the frequency distribution of different versions of the same program.

TABLE I. The frequency of the first 28 machine codes for one version of aircrack, gftp and nmap software

Command	Aircrack	Gftp	Nmap
aaa	0	0	0
aad	0	0	0
aam	0	0	0
aas	0	0	0
adc	318	629	7708
add	4815	14344	102198
and	1211	2109	14102
call	1243	6464	15333
cbw	0	0	1
clc	12	7	749
cld	6	119	876

cli	5	6	571
cmc	8	4	796
cmp	1088	1645	13640
cmpsb	0	0	0
cmpsw	0	0	0
cwd	0	0	2
daa	0	0	0
das	0	0	0
dec	66	71	424
div	2	6	68
esc	0	0	0
hlt	15	16	499
idiv	22	3	356
imul	251	961	5343
in	56	94	2545
inc	59	78	584
int	8	7	614

TABLE II. THE FREQUENCY OF THE FIRST 28 MACHINE CODES FOR FOUR DIFFERENT VERSIONS OF AIRCRACK SOFTWARE

Command	Aircrack-0	Aircrack-1	Aircrack-2	Aircrack-3
aaa	0	0	0	0
aad	0	0	0	0
aam	0	0	0	0
aas	0	0	0	0
adc	318	256	316	270
add	4815	4180	4784	5176
and	1211	1195	1134	1152
call	1243	1242	1221	1187
cbw	0	0	0	0
clc	12	12	17	17
cld	6	6	7	8
cli	5	5	6	5
cmc	8	8	10	9
cmp	1088	1082	1046	1013
cmpsb	0	0	0	0
cmpsw	0	0	0	0
cwd	0	0	0	0
daa	0	0	0	0
das	0	0	0	0
dec	66	66	31	49
div	2	2	2	2
esc	0	0	0	0
hlt	15	17	16	24
idiv	22	22	22	24
imul	251	237	252	253
in	56	30	33	28
inc	59	57	51	85
int	8	8	8	8

Then each file is added to the training set in the format shown in Fig. 1.

B. Second stage: calculation of the Minkowski metric

1) For the identified file the signature is formed by the same method as for each of the training file in the first stage (Fig. 3, Fig. 4, Fig. 5).

2) For each chosen from the interval [0,02; 4] q the algorithm described below is applied.

The method of potential functions while selecting the reference object as the center of mass is used for calculation of the following parameters:

- For each class of programs from the archive there is a reference object according to the equation (4) with the use of all the files in the sample for this class. For each machine code the arithmetic mean of values of all the frequencies in the sample is calculated.
- Between each of the resulting standards and identified file the Minkowski metric is calculated for the given value of the parameter q in the equation (2).

The following value are calculated by the method of potential functions using the SLOP algorithm:

- The value of the parameter k from the interval [1; 4] is fixed.
- Between each object from the training set and identifiable object the Minkowski metric for the given value of the parameter q in the equation (2) is computed.
- The array of k highest values of these metrics is composed and the names of those classes of metrics between the object and objects of these classes, which are moved into the array.
- All the training sample feature resulted into the array containing the maximum metric between the objects from training set and identifiable object. These metrics determine the k objects that are the closest to identifiable.

C. The third stage: the file identification

The file identification is realized on the basis of the results of calculations in accordance with the sequence of methods applied in the second stage:

For the method of potential functions while selecting the standard object as the center of mass of the sample and the reference object by the method of STOLP and each set q from the interval [0,02; 4]:

- potentials created by each standard object according to the equation $K(\rho) = 1/\rho^2$, where ρ is the previously calculated metric, are calculated,
- maximum potential is selected,
- the name of the class that owns the maximum potential is defined. This name is the answer of the classifier.

For k-nearest neighbors methods, each set q from the interval [0,02; 4] and k from interval [1; 4]:

1) From the calculated array the names of all classes are retrieved, which metrics identified between the object and objects of these classes were placed in the array.

2) For each extracted class name the value is computed (how many objects belonging to this class out of the total number k are in the deck).

3) The name of the class with the highest value is the answer of the classifier.

In this paper the identification accuracy is the number (%) of correctly classified programs. Fig. 3 and Fig. 4 shows the dependence of the classification accuracy on the value of the parameter q for the method of potential functions, and Fig. 5 shows the dependence for k-nearest neighbors for different values of k.

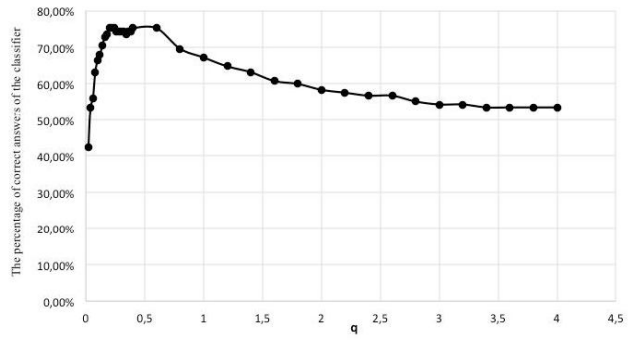

Fig. 3. The accuracy of the method of potential functions while selecting the standard object as the center of mass

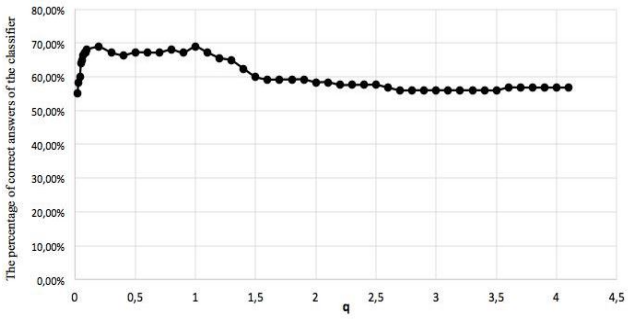

Fig. 4. The accuracy of the method of potential functions while choosing the standard object by the SLOP algorithm

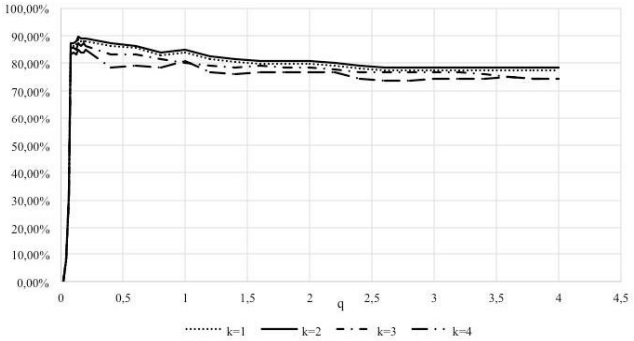

Fig. 5. The accuracy of k-nearest neighbors method with k = 1, 2, 3, 4

Thus, the values of the parameter q experimentally determined, for which the maximum classification accuracy is achieved. According to these dependences:

1) Maximum accuracy of k-nearest neighbors method is 89,60% of correctly classified programs; it was achieved at $q = 0{,}14$ in the case of $k = 2$. So the equation for the calculation of the Minkowski metric in this case is:

$$\rho(u, x_u^{(i)}) = \left(\sum_{i=1}^{118} \left| u - x_u^{(i)} \right|^{\frac{1}{7}} \right)^7 .$$

It should be noted that the identification accuracy decreases with increasing values of k.

2) Maximum accuracy of the method of potential functions with the choice of the standard when using the equation of center of mass is 75,20% of correctly classified programs at $q = q{,}4$ a. In this case, the equation to calculate the Minkowski metric is:

$$\rho(u, x_u^{(i)}) = \left(\sum_{i=1}^{118} \left| u - x_u^{(i)} \right|^{\frac{2}{5}} \right)^{\frac{5}{2}} .$$

3) Maximum accuracy of the method of potential functions with the choice of standard using the STOLP method is 68,80% of correctly classified programs at $q = q{,}q$. Therefore, the equation for the calculation of the Minkowski metric in this case is:

$$\rho(u, x_u^{(i)}) = \left(\sum_{i=1}^{118} \left| u - x_u^{(i)} \right|^{\frac{1}{5}} \right)^5 .$$

IX. Conclusion

The method of identification of elf-files based on the metric of the classification algorithms has shown its applicability in tasks of internal audit of computer equipment. In the process of the experiment it was revealed that the maximum identification accuracy (89,60%) is achieved by using the classification algorithm based on the method of k-nearest neighbors if $k = q$. It should also be noted that the use of occurrence frequencies of machine codes has the advantage over byte-by-byte frequency characteristic in terms of that in order to build the archive template the larger number of elf-files can be used that makes it more unified.

Further modification of the method can be based on the use of other classification algorithms, including metric. Also it is essential to solve the problem of "curse of dimensionality" and selection of the most significant feature of machine codes.

The results indicate that this method can be applied not only to problems of identification when conducting internal audit of computer equipment, but also in other tasks of computer forensics.

References

[1] V.M.Korzhuk, A.P.Kuzmich and G.V.Shved, "The method of an audit of software containing in digital drives", *AICT 2014 (8th IEEE International Conference),* 2014, pp. 128-132.

[2] W.M.Khoo, A.Mycroft and R.Anderson, "Rendezvous: a search engine for binary code", *The 10th Working Conference on Mining Software Repositories*, 2013, pp. 329-338.

[3] S.Moody and R.ErbacherSadi, "Statistical analysis for data type identification", *Systematic Approaches to Digital Forensic Engineering, SADFE '08*, 2008, pp. 41-54.

[4] I.E.Krivtsova, K.I.Salakhutdinov and I.V.Yurin, "Method of executable filts identification by their signatures", *The scientific journal "Vestnik gosudarstvennogo universiteta morskogo i rechnogo flota imeni admirala S.O. Makarova"*, vol.1, no.35, 2016, pp. 215-224.

[5] M.A.Aizerman, E.M.Braverman and L.I.Rozonoer, "Method of potential functions in the theory of machine learning", *Nauka*, 1970, p. 320.

[6] Wiki-resourse of Machine Learning website, Method of potential functions, Web: http://www.machinelearning.ru/wiki/index.php?title=Метод_потенциальных_функций.

[7] N.G.Zagoruiko, *Applied methods of data and knowledge analysis*. Novosibirsk, Russian Federation: Sobolev institute of mathematics, 1999.
[8] K.V.Vorontsov, "Lectures on metric algorithms of classification", Web: http://machinelearning.ru/wiki/images/9/9d/Voron-ML-Metric.pdf.
[9] N.N.Fedotov, *Forensics – computer forensics,* Moscow, Russian Federation: The Legal World, 2007.
[10] GFS-team website, ElF file Structure, Web: http://www. gfs-team.ru/articles/read/149.
[11] M.Sukhanov, "Computer counter forensics: state and prospects", Web: http://www.securitylab.ru/analytics/397811.php.
[12] J.P.Van de Geer, *Some Aspects of Minkowski Distance*. Publisher: Leiden University, Department of Data Theory, 1995.
[13] AccessData official website, Forensic Toolkit (FTK): Recognized around the World as the Standard Digital Forensic Investigation Solution, Web: http://www.accessdata.com/products/digital-forensics/ftk.
[14] Acid-burninfo (in update status) web-blog, Advantages and Disadvantages of FTK and EnCase, Web: http://acid-burninfo.blog spot.ru/ Advantages and Disadvantages of FTK and EnCase.
[15] V.Olifer and N.Olifer, *Computer network. Principles, technologies, protocols*, St. Petersburg, Russian Federation: Piter, 2007.
[16] Inc. Guidance Software official website, EnCase Forensic v7.10: The Fastest, Most Comprehensive Forensic Solution Available, Web: http://www.guidancesoftware.co m/encase-forensic.htm.
[17] D.Hurlbut, "Fuzzy Hashing for Digital Forensic Investigators", Web: https://ad-pdf.s3.amazonaws.com/Fuzzy_Hashing_for_Investi gators.pdf.
[18] L.I.Brylevskoe, I.A.Lapin, L.S.Rachathewa and O.L.Suslina, *The elements of the theory of linear spaces. Tutorial.* St. Petersburg, Russian Federation: ITMO University, 2001.