e-ISSN 1607-3274 Радіоелектроніка, інформатика, управління. 2022. № 2
p-ISSN 2313-688X Radio Electronics, Computer Science, Control. 2022. № 2

UDC 004.8

DEVELOPMENT OF METHOD FOR IDENTIFICATION THE COMPUTER SYSTEM STATE BASED ON THE DECISION TREE WITH MULTI-DIMENSIONAL NODES

Gavrylenko S. Y. – Dr. Sc., Professor, Professor at Department of Computer Engineering and Programming, National Technical University "Kharkiv Polytechnic Institute", Kharkiv, Ukraine.

Chelak V. V. – Post-graduate Student at Department of Computer Engineering and Programming, National Technical University "Kharkiv Polytechnic Institute", Kharkiv, Ukraine.

Semenov S. G. – Dr. Sc., Professor, Head at Department of Computer Engineering and Programming, National Technical University "Kharkiv Polytechnic Institute", Kharkiv, Ukraine.

ABSTRACT

Context. The problem of identifying the state of a computer system is considered. The object of the research is the process of computer system state identification. The subject of the research is the methods of constructing solutions for computer system state identification.

Objective. The purpose of the work is to develop a method for decision trees learning for computer system state identification.

Method. A new method for constructing a decision tree is proposed, combining the classical model for constructing a decision tree and the density-based spatial clustering method (DBSCAN). The simulation results showed that the proposed method makes it possible to reduce the number of branches in the decision tree, which will increase the efficiency of identifying the state of the computer system. Belonging to hyperspheres is used as a criterion for decision-making, which enables to increase the identification accuracy due to the nonlinearity of the partition plane and to perform a more optimal adjustment of the classifier. The method is especially effective in the presence of initial data with high correlation coefficients, since it combines them into one or more multivariate criteria. An assessment of the accuracy and efficiency of the developed method for identifying the state of a computer system is carried out.

Results. The developed method is implemented in software and researched in solving the problem of identifying the state of the functioning of a computer system.

Conclusions. The carried out experiments have confirmed the efficiency of the proposed method, which makes it possible to recommend it for practical use in order to improve the accuracy of identifying the state of a computer system. Prospects for further research may consist in the development of an ensemble of decision trees.

KEYWORDS: computer system, abnormal state, identification, decision tree, clustering, DBSCAN algorithm, hypersphere.

ABBREVIATIONS

CS is a computer system;

OS is an operating system;

DT is a decision tree;

DBSCAN is a density-based spatial clustering of applications with noise (a data clustering algorithm).

NOMENCLATURE

X is the source data (OS events);

m is a number of the object features;

n is a number of classes in the source subset;

N_i is a number of samples of the i-th class;

N is a total number of samples in the subset;

p_k is a probability of belonging to the k-th class;

w is classifier settings;

I is information gain;

MinPts is a minimum number of neighbors for creating a cluster;

$|C|$ is a number of objects in the largest cluster;

ε is a radius of the neighborhood hypersphere;

d is a distance between objects that are clustered;

xc is a set of coordinates of the cluster center;

η is a radius of the hypersphere that bounds the cluster.

INTRODUCTION

Today, computer technology is an integral part of any state and determines its economic and political role internationally. Despite a number of promising developments in information security, the number of man-made disasters and accidents and attempts to destabilize the functioning of computer systems is increasing [1]. This is due to the imperfection of methods and means of data protection, as well as increased interest in the CS on the part of attackers. That is why the issue of CS state identification in order to spot and localize the destabilizing actions of its functioning in an increasing number of external influences is an urgent task.

The computer system is characterized by a large amount of performance criteria, which leads to difficulties in choosing the most informative criteria and the development of methods for identifying its condition under external influences [2, 3].

Researches of existing methods have revealed a number of limitations in their use [2, 4]. Thus, when the CS operates on the border between normal and abnormal states, modern methods do not always remain effective and require a long time, along with software and hardware resources, which leads to a decrease in efficiency and accuracy of its state identification [5, 6].

In addition, such tasks become especially relevant when the initial data are heterogeneous, absent or insufficient, but there are some observations in the functionality

DOI 10.15588/1607-3274-2022-2-11

of the CS, which is under the condition identification [7]. For this class of tasks, a highly effective tool is DTs and their ensembles [8–10] – a way to represent the rules in a hierarchical structure, where each object corresponds to a single node, which gives the resulting solution. A rule means a logical construction, presented in the form of if-then construct.

The object of the research is the process of computer system state identification.

DT (classification tree, regression tree) is one of the methods of automatic data analysis. DTs allow you to find repetitive patterns in the data, and to perform training to recognize patterns. The fundamental work that gave impetus to the development of this area was the book by E.B. Hunt, J. Marin and P.J. Stone in "Experiments in Induction", which was published in 1966. DT has a number of advantages [4, 5], namely: easy to understand and interpret, able to work with both numerical and categorical data, requires little data preparation, uses a white box model and is easily explained by Boolean logic. The correctness of the model can be verified by statistical tests, which makes it possible to verify its reliability. In addition, during the construction of DT, less informative features will be used to a lesser extent, which makes it possible to either remove them from subsequent runs or use special algorithms for taking into account less informative features [11].

However, DTs have a number of disadvantages [4, 5]. The problem of constructing an optimal DT is NP-complete in terms of some aspects of optimality even for simple tasks. The DT construction algorithm is based on a greedy algorithm, where the only optimal solution is selected locally in each node, which cannot ensure the optimality of the whole tree. DT also has a high sensitivity to noise and changes in the source data, which can lead to the construction of a completely different DT, even with small changes in the source data.

The subject of the research is the methods of constructing solutions for computer system state identification.

There are various methods of constructing DTs, the process of which is a consistent, recursive division of the learning set into subsets using the decision rules in the nodes [12]. The process of partitioning continues until all the nodes at the end of all the branches are declared as leaves. At the same time, when constructing a DT, partitioning in nodes forms rectangular clusters in the feature space, the shape of which may not coincide with the shape of real clusters, which leads to a decrease in the accuracy of decision-making. This is especially important when the functioning of the CS lies on the verge of distinguishing between normal and abnormal states, is characterized by highly correlated data, or is presented by fuzzy data that requires the development of new models of DT construction [6, 13–16].

The purpose of the work is to develop a method for decision trees learning for computer system state identification.

1 PROBLEM STATEMENT

We will assume that the functioning of a CS is characterized by the set of its performance criteria $X = \{x_{i1}, x_{i2}, ..., x_{im}\}$. Input data is the set of marked pairs $\{(x_i, y_i)\}_{i=1}^{N}$, where x_i is the CS state criteria set and y_i is a classifying label. There exists an unknown fitness function – a mapping $\boldsymbol{f : X \rightarrow Y}$ the values of which are only known for a finite set of training samples $(X,Y)=\{(x_1, y_1),\ldots,(x_m,y_m)\}$. The structure of a DT $\boldsymbol{f}$, must be formed, which should be able to classify an arbitrary object $x \in X$ and adjust its parameter w: $F(f(w, x), y) \rightarrow opt$.

2 REVIEW OF THE LITERATURE

Most popular decision tree learning algorithms are based on the divide-and-conquer principle [17].

During the construction of the DT, it is necessary to solve several key problems, each of which is associated with the corresponding step of the learning process:

1) Choice of the partition attribute for a given node.
2) Choice of termination criteria for learning.
3) Choice of decision tree pruning method.
4) Assessment of the accuracy of the constructed tree.

Currently, a significant number of algorithms have been developed for choosing the next partition attribute (DT learning algorithms): ID3, CART, C4.5, C5.0, Newld, ITrule, CHAID, CN2, etc. The most widespread and popular are the following algorithms:

1) ID3 (Iterative Dichotomized) – the algorithm can only use a discrete target variable, so DTs that are built using this algorithm are classifiers [18, 19]. Attribute choice is based on information gain:

$$I = -\sum_{i=1}^{p} \frac{N_i}{N} \log\left(\frac{N_i}{N}\right),$$

or based on Gini impurity:

$$I = \sum_{k=1}^{n} p_k (1 - p_k).$$

For this algorithm the number of children of a tree node is not limited. The algorithm does not support training samples with incomplete data.

2) C4.5 – an improved version of ID3, which adds the ability to work with missing data values. Attribute choice is based on information gain [19,20].

3) CART (Classification and Regression Tree) – a decision tree learning algorithm, which allows the use of both discrete and continuous target variables, i.e. it can solve both classification and regression problems. The algorithm builds trees that have only two children in each node, i.e. it builds a binary DT. Works slowly on large input data with lots of noise [21, 22].

4) Chi-square automatic interaction detection (CHAID). Performs multiway splits during the DT classification calculation DT.

DOI 10.15588/1607-3274-2022-2-11

5) MARS*:* extends DT to improve digital data processing.

No algorithm for constructing a DT can a priori be considered the best or perfect. Feasibility of a particular algorithm should be verified and confirmed experimentally.

Since CS is characterized by a large number of performance criteria, the significance of which is uncertain and which correlate with each other and can characterize the CS state as being in between normal and abnormal states, it is necessary to improve existing or develop new methods of identifying the CS state.

3 MATERIALS AND METHODS

In this study, in accordance with the problem statement, a DT construction method was developed, which differs from the known methods by combining the classical model of DT construction based on the *C4.5* algorithm with the DBSCAN method.

The DBSCAN algorithm was proposed by Martin Ester, Hans-Peter Kriegel, Jörg Sander and Xiaowei Xu in 1996, as a solution to the problem of partitioning data into clusters of arbitrary shape.

The algorithm is based on the idea that inside each cluster the density of objects is significantly higher than outside, and that the density in areas with noise is lower than the density of any of the clusters.

The algorithm requires two parameters: *MinPts* – the minimum number of neighbors for creating a cluster; ε – the radius of the neighborhood hypersphere.

The first step of the algorithm is to compute a matrix of distances *d* between objects that are being clustered, either using the squared Euclidean distances:

$$d(x_i, x_j) = \sum_{k=1}^{m} (x_{ik} - x_{jk})^2,$$

or the Manhattan distances:

$$d(x_i, x_j) = \sum_{k=1}^{m} \left| x_{ik} - x_{jk} \right|.$$

In the next step, a neighbor matrix is computed using the distance matrix, each element in which determines whether the object x_i is a neighbor of x_j:

$$Neighbour_{ij} = \begin{cases} 0, & d(x_i, x_j) > \varepsilon \\ 1, & d(x_i, x_j) \le \varepsilon \end{cases},$$
$$i = \overline{1..N},\ j = \overline{1...N}.$$

Subsequently, clusters are formed based on the neighbor matrix. Initially, all objects are considered undetermined. The clustering procedure is iterative, and starts with an arbitrary object x_i which has not been determined yet. For a given object x_i, a list of neighbors is created, which contains all objects x_j, that have the corresponding element of the *i*-th row of the neighbor matrix set to one. The number of neighbors *K* is counted and compared with *MinPts*. When the count of neighbors is less than *MinPts*, the object is labeled as unclustered, and the next arbitrary undetermined object x_i. is processed. When the count of neighbors is greater or equals to *MinPts*, the current object x_i is considered to be a core object. Objects x_j, which were included in the list are considered reachable in terms of density (also core objects). The current object x_i and its neighbors x_j form a new cluster and are labeled with its number. Next, the iterative process of finding new neighbors is started. Objects that are either undetermined or unclustered are analyzed, and those that are reachable for the x_i objects of the cluster (have the corresponding element of the *j*-th row of the neighbor matrix set to one), are added to the cluster. The iterative process of joining new neighbors is repeated until no more objects can be added to the cluster.

The process of forming new clusters is repeated until all objects are determined. Objects that were labeled as unclustered and were not subsequently placed into a cluster are considered noise and remain unused.

The following procedure of finding the decision parameter η for a multidimensional DT node is developed:

– the cluster with the maximum number of elements *C* is found:

$$C = \max_{A_i \in A}(|A_i|),$$

where $|A_i|$ – is the number of elements in the *i*-th cluster;

– The center of the cluster *xc* is found using each feature of the x_i object:

$$xc_k = \frac{\sum_{i=1}^{|C|} x_{ik}}{|C|}.$$

After obtaining the center of the cluster, η is defined to be the maximum distance from the object to the centers:

$$\eta = \max_{x_i \in C}(d(xc, x_i)).$$

The value of η is the radius of the hypersphere that bounds the cluster and is further used as a decision parameter for the multidimensional DT node.

The process of constructing a DT is as follows. The source data of the DT are the indicators of the functioning of the CS (CPU load, memory load, network traffic, number of read/write operations to disk, intrusion signatures; statistical data based on system events: number of operations with the registry or file system, number of processes, etc.). The source data is divided into clusters using the DBSCAN algorithm. For example, when identifying the CS state, a singular multidimensional criterion can combine features representing the load of individual CPU

DOI 10.15588/1607-3274-2022-2-11

cores. Each of clusters can be further considered as a multidimensional criterion in the construction of the next DT node.

Further process of constructing a DT consists of sequential, iterative division of the learning set into subsets using the decision rules in the nodes. When a given DT node is formed, the feature that gives the best partitioning out of the whole set of features is selected as the partition feature, providing the maximum entropy reduction of the resulting subset relative to the parent. The feature can be either one-dimensional or multidimensional. If the partition feature is one-dimensional, the partition criterion is a comparison with a given threshold value. If the partition feature is multidimensional, the partition criterion belongs to a hypersphere of a given radius η.

Fig. 1 shows a construction of a tree with multidimensional decision nodes. Fig. 2 shows a construction of a decision tree which uses a one-dimensional feature and two features, combined into a single two-dimensional criterion.

Thus, the method of constructing DT can be formulated as follows:

1. To form a training sample of labeled data <x, y>.

2. To divide the source data into clusters using the DBSCAN algorithm

3. Determine the termination criteria of DT construction to avoid overlearning.

To do this, we considered the following approaches:

– Early termination – the algorithm will be aborted as soon as the specified value of a criterion is reached, such as the percentage of correctly recognized samples. The advantage of the approach is the reduction of training time and reduction of variance error, and the disadvantage is the reduction of the accuracy of DT classification;

– Limiting the tree depth – the establishment of the maximum number of partitions in the branches, after which the training stops. This method also leads to a decrease in the accuracy of DT classification;

– Establishment of the minimum admissible number of leaves in a node, which will allow to avoid creation of trivial splits and, consequently, insignificant rules.

4. Determine the information gain Ii of all one-dimensional and multidimensional features in relation to the result value y_i. and select the attribute that will be partitioned in this node.

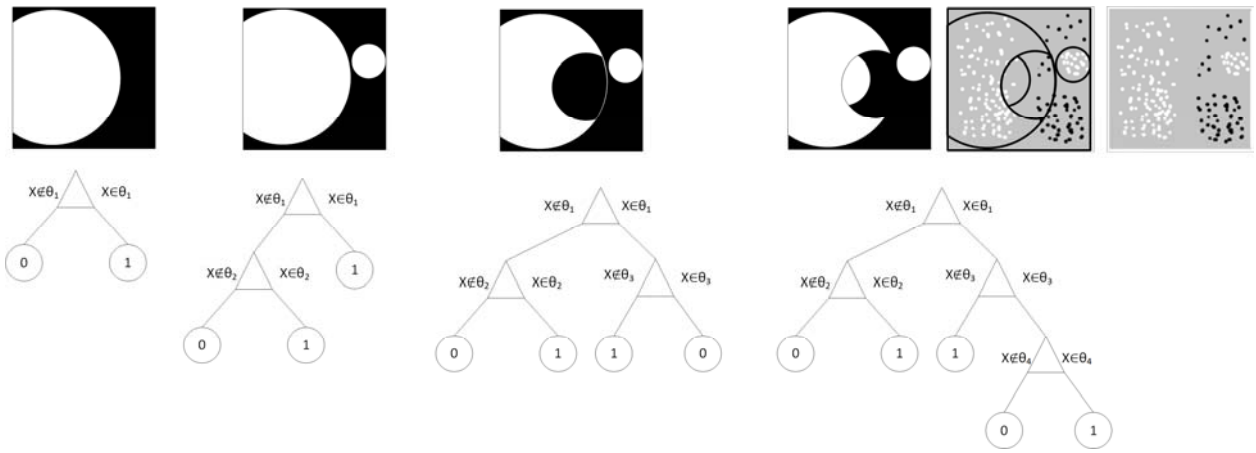

Figure 1 – Example of constructing a tree with multidimensional decision nodes

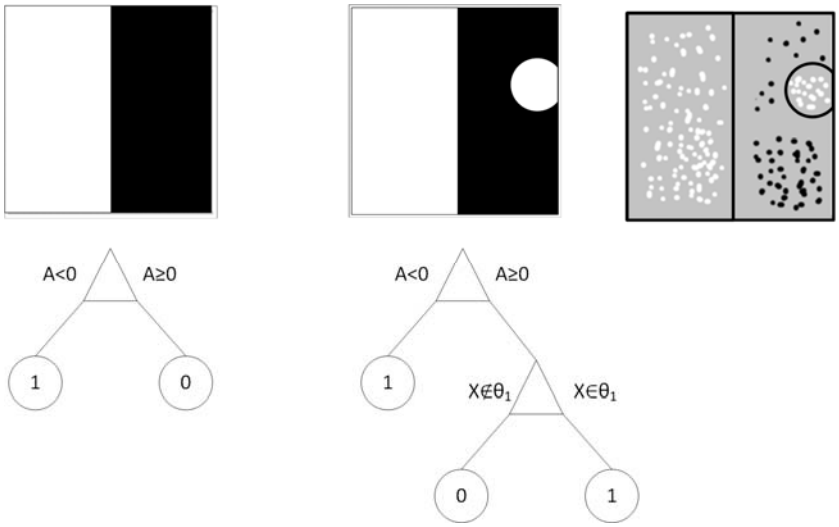

Figure 2 – Example of constructing a tree with multidimensional and one-dimensional nodes

DOI 10.15588/1607-3274-2022-2-11

5. Construct the current DT node based on the selected feature and form leaves with the appropriate set of samples.

6. Check the defined tree termination criteria. Complete the DT construction procedure if at least one of the termination criteria meets the requirements, or return to step 4.

7. If necessary, prune the DT, using the following algorithm:

– Identify two indicators: the relative accuracy of the model (the ratio of the number of correctly recognized samples to the total number of samples) and the absolute error value (the number of incorrectly classified samples);

Remove leaves and nodes from the tree, the cutting of which will not significantly reduce the accuracy of the model or increase the error. Cutting branches, carried out from bottom to top, by successively transforming the nodes into leaves.

4 EXPERIMENTS

According to the proposed algorithm, the DT is constructed (Fig. 3). A comparative analysis of classification accuracy was performed. The following DT construction algorithms have been examined as DT-based classification methods: Fine Tree, Medium Tree, Coarse Tree. Classifiers based on support-vector machines (SVM) and *k*-nearest neighbors (KNN) were also considered. Table 1 shows a comparative estimate of the classification error in the training set (Bias) and the test set (Variance).

As can be seen from Table 1, the method of DT construction, which combines the classical model of DT construction and density-based method of spatial clustering, makes it possible to achieve the accuracy of up to 100% for the training set, while the classification error on the test data set does not exceed 9.1%.

The evaluation of the performance of the classifier based on the proposed method in comparison with previously known methods is shown in Fig. 4. As can be seen from the figure, the proposed method leads to an increase in the efficiency of identification of the state of the CS by 50% compared to the Medium Tree method, which was shown to be the most efficient in previous studies [23].

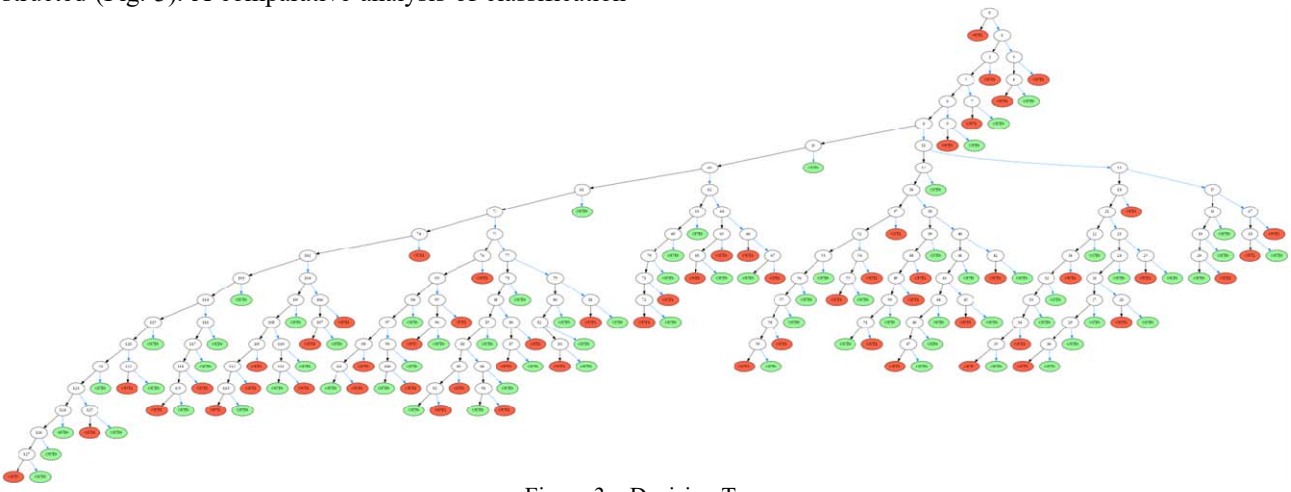

Figure 3 – Decision Tree

Table 1 – Assessment of classification accuracy

Method	Bias, %	Variance %	Method	Bias, %	Variance %
Fine Tree	0.13	31.97	Fine KNN	0	8.63
Medium Tree	0.13	35.03	Medium KNN	0.03	19.57
Coarse Tree	0.23	46.87	Coarse KNN	0.37	45.7
Decision tree with multi-dimensional nodes	0	9.1	Cosine KNN	0.1	28.77
Linear SVM	0.87	33.73	Cubic KNN	0.03	43.73
Quadratic SVM	0.07	48	Weighted KNN	0.03	10.97
Fine Gaussian SVM	0.17	28.87	Subspace Discriminant	3.47	56.43
Medium Gaussian SVM	0.03	42.1	Subspace KNN	0	10.37
Coarse Gaussian SVM	1.87	43.13			

DOI 10.15588/1607-3274-2022-2-11

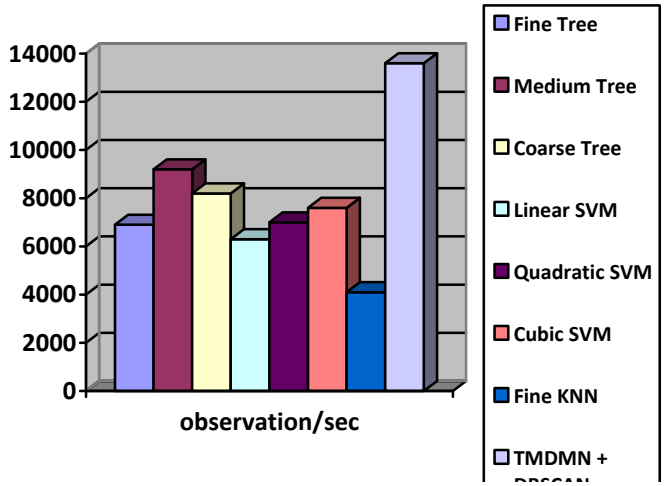

Figure 4 – Performance of CS state identification

5 RESULTS

The simulation results have shown that the proposed method makes it possible to reduce the number of branches in the DT, which leads to an increase in the efficiency of CS state identification by 50%. The use of belonging to the hyperspheres as a decision criterion makes it possible to increase the accuracy of identification (and achieve classification error rates as low as 0% on the training set, and 9.1% on the test data set) due to the nonlinearity of the partitioning plane. Furthermore, a larger set of hyperparameters allows for a more optimal and flexible fine-tuning of the classifier. This method is especially effective when used with source data samples that have high correlation coefficients, as it combines them into one or more multidimensional criteria. The disadvantage of this method is the increase in the training time of the classifier and a slight increase in the amount of resources needed for storage of the obtained models.

6 DISCUSSION

A number of limitations in the use of existing methods have been identified while solving problems related to the identification and protection of computer systems. Thus, anomalies caused by intrusions into the CS with unidentified or fuzzy properties, given the large number of parameters of the functioning of the CS, can not always be identified, which leads to increased damage as a result of cyberattacks.

That is why the conducted research has led us to a method of CS state identification based on DTs. The conducted experiments have allowed us to assess the accuracy and efficiency of the CS state identification, the practical significance and prospects of further research.

CONCLUSIONS

Hence, the problem of increasing the efficiency and accuracy of CS state identification is solved in this work.

The scientific novelty of the obtained results is that for the first time a method of identifying the CS state based on DTs is proposed, which differs from the known methods of DT construction by combining the classical model of tree construction with the DBSCAN method.

The initial data of the DT are CS performance criteria, processed by a special algorithm, namely: criteria that are highly correlated are combined into one or more multidimensional criteria.

A comparative analysis of the accuracy of the proposed algorithm for constructing DTs and the following algorithms: Fine Tree, Medium Tree, Coarse Tree. In addition, classifiers based on SVM and KNN methods were studied.

The simulation results showed that the proposed method makes it possible to reduce the number of branches in the DT, which leads to an increase in the efficiency of CS state identification by 50%. The use of belonging to the hyperspheres as a decision criterion makes it possible to increase the accuracy of identification (and achieve classification error rates as low as 0% on the training set, and 9.1% on the test data set) due to the nonlinearity of the partition plane. In addition, the presence of more hyperparameters allows for a more optimal fine-tuning of the classifier. This method is especially effective when used with source data samples that have high correlation coefficients, as it combines them into one or more multidimensional criteria. The disadvantage of this method is the increase in training time of the classifier. The method also requires more memory.

The practical significance lies in the fact that the developed method is implemented in software and has been researched while solving the problem of CS state identification.

The experiments confirmed the efficiency of the proposed method, which makes it possible to recommend it for practical use as a state identification method.

Prospects for further research may consist of development of an ensemble of trees with multidimensional decision nodes.

DOI 10.15588/1607-3274-2022-2-11

ACKNOWLEDGEMENTS

This work is supported by National Technical University "Kharkiv Polytechnic Institute" in the field of research "Models and methods of processing and protection of information in computer systems" (№60028).

REFERENCES

1. Daniel Schatz, Bashroush Rabih, Wall Julie. Towards a More Representative Definition of Cyber Security, *The Association of Digital Forensics, Security and Law* (ADFSL), 2017, Vol. 12, No. 2, pp. 53–74. DOI: 10.15394/jdfsl. 2017.1476
2. Farooq Anjum and Petros Mouchtaris. Intrusion Detection Systems, *Security for Wireless Ad Hoc Networks*. Wiley, 2007, pp. 120–159. DOI: 10.1002/ 9780470118474.ch5.
3. Kelleher J., Namee B., Archi A. Fundamentals of Machine Learning for Predictive Data Analytics: Algorithms, Worked Examples and Case, Dublin: The MIT Press, 2015, 642 p.
4. Iqbal H. Sarker, Shahriar Badsha, Hamed Alqahtani, Paul Watters, Alex Ng. Cybersecurity data science: an overview from machine learning perspective, *Journal of Big Data*, 2020, Vol. 7 (41) pp. 1–29. DOI: 10.1186/s40537-020-00318-5
5. Xavier Larriva-Novo, Mario Vega-Barbas, Victor A. Villagra, Diego Rivera, Manuel Alvarez-Campana, Julio Berrocal. Efficient Distributed Preprocessing Model for Machine Learning-Based Anomaly Detection over Large-Scale Cybersecurity Datasets, *Applied Sciences*, 2020, Vol. 10, pp. 30–34. DOI: 10.3390/app10103430
6. Gavrylenko S., Semenov S., Sira O., Kuchuk N. Identification of the state of an object under conditions of fuzzy input data, *Eastern-European Journal of Enterprise Technologies,* 2019, Vol. 1, No. 4 (97), pp. 22–29. DOI: 10.15587/1729-4061.2019.157085
7. Alpaydin E. Introduction to Machine learning, London: The MIT Press, 2010, 400 p.
8. Kaminski B., Jakubczyk M., Szufel P. A framework for sensitivity analysis of decision trees, *Central European Journal of Operations Research,* 2018, Vol. 26, pp. 135–159. DOI: 10.1007/s10100-017-0479-6
9. Gavrylenko S., Sheverdin I, Kazarinov M. The ensemble method development of classification of the computer system state based on decision trees, *Advanced Information Systems*, Vol. 4, No. 3, pp. 5–10. DOI:10.20998/2522-9052.2020.3.01
10. SubbotIn S. Podannya y obrobka znan u sistemah shtuchnogo Intelektu ta pidtrimki priynyattya, Zaporizhzhya, ZNTU, 2008, 341 p.
11. Subbotin S. O. Postroenie derevev resheniy dlya sluchaya maloinformativnyih, *Radio Electronics, Computer Science, Contro*l, 2019, No. 1, pp. 122–130. DOI: 10.15588/1607-3274-2019-1-12
12. Mitrofanov S., E. Semenkin. An Approach to Training Decision Trees with the Relearning of Nodes, *International Conference on Information Technologies* (*InfoTech*), 2021, pp. 1–5, DOI: 10.1109/InfoTech52438.2021.9548520
13. Wang S., Wang, L., Jiang, C. Adapting naive Bayes tree classification, *Knowledge and Information system*, 2015, Vol. 44, No. 1, pp. 77–89. DOI: 10.1007/s10115-014-0746-y
14. Kornienko Y., Borisov A. A hybrid algorithm for decision tree generation, *International Scientific Journal of Computing*, 2004, Vol. 3, Issue 3, pp. 51–57. DOI: 10.47839/ijc.3.3.305
15. Irad Ben-Gal, Alexandra Dana, Niv Shkolnik, Gonen Singer. Efficient Construction of Decision Trees by the Dual Information Distance Method, *Quality Technology & Quantitative Management,* 2014, Vol. 11, No. 1, pp. 133–147. DOI: 10.1080/16843703.2014. 11673330
16. Geurts P., Ernst D., Wehenkel L. Extremely randomized trees, *Machine Learning*, 2006, Vol. 63, No. 1, pp. 3–42. DOI: 10.1007/s10994-006-6226-1
17. Kesinee Boonchuay. Krung Sinapiromsaran, Chidchanok Lursinsap Boundary expansion algorithm of a decision tree induction for an imbalanced dataset, *Songklanakarin Journal of Science and Technology* (SJST), 2017, Vol. 39, No. 5, pp. 665–673. DOI: 10.14456/sjst-psu.2017.82
18. Quinlan J. R. Induction of Decision Trees, *Machine Learning*, 1986, No. 1, pp. 81–106.
19. Hssina B.,Merbouha A., Ezzikouri H., Erritali M. comparative study of decision tree ID3 and C4.5, *International Journal of Advanced Computer Science and Applications,* 2014, Vol. 4(2), pp. 13–19. DOI: 10.14569/SpecialIssue.2014.040203
20. Idris Mochamad, Mustafid, Suseno Jatmiko Endro. Implementation of C4.5 Algorithm and Forward Chaining Method for Higher Education Performance Analysis, *The 4th International Conference on Energy, Environment, Epidemiology and Information System,* 2019, Vol. 125. DOI: 10.1051/e3sconf/ 201912521002
21. Painsky A., Rosset S. Cross-Validated Variable Selection in Tree-Based Methods Improves Predictive Performance, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 2017, Vol. 39, No. 11, pp. 2142–2153. DOI: 10.1109/TPAMI.2016.2636831.
22. Maddeh M., Ayouni S., Alyahya S., Hajjej F. Decision tree-based Design Defects Detection, *IEEE Access,* 2021, Vol. 9, pp. 71606–71614. DOI: 10.1109/ACCESS.2021.3078724.
23. Gavrylenko S., Chelak V., Hornostal O. Ensemble approach based on bagging and boosting for Identification the Computer System State, Proceedings of the 31th International Scientific Symposium Metrology and Metrology Assurance.–Sozopol, Bulgaria IEEE Access, 2021. DOI:10.1109/ MMA52675.2021.9610949

Received 03.12.2021.
Accepted 10.12.2021.

УДК 004.8

РОЗРОБКА МЕТОДУ ІДЕНТИФІКАЦІЇ СТАНУ КОМП'ЮТЕРНОЇ СИСТЕМИ НА ОСНОВІ ДЕРЕВА РІШЕНЬ З БАГАТОВИМІРНИМИ ВУЗЛАМИ

Гавриленко С. Ю. – д-р техн. наук, професор, професор кафедри «Обчислювальна техніка та програмування», Національний технічний університет «Харківський політехнічний інститут», Харків, Україна.

Челак В. В. – аспірант кафедри «Обчислювальна техніка та програмування», Національний технічний університет «Харківський політехнічний інститут», Харків, Україна.

DOI 10.15588/1607-3274-2022-2-11

Семенов С. Г. – д-р техн. наук, професор, завідувач кафедри «Обчислювальна техніка та програмування», Національний технічний університет «Харківський політехнічний інститут», Харків, Україна.

АНОТАЦІЯ

Актуальність. Розглянуто задачу ідентифікації стану комп'ютерної системи. Об'єктом дослідження є процес ідентифікації стану комп'ютерної системи. Предметом дослідження є методи побудови дерев рішень для ідентифікації стану КС

Мета. Розробка методу побудови дерев рішень для ідентифікації стану комп'ютерної системи.

Метод. Запропоновано новий метод побудови дерева рішень, який поєднує класичну модель побудови дерева рішень та оснований на щільності метод просторової кластеризації (DBSCAN). Результати моделювання показали, що запропонований метод надає можливість зменшити кількість розгалужень в дереві рішень, що дозволяє підвищити оперативність ідентифікації стану комп'ютерної системи. Використання приналежності до гіперсфер у якості критерію прийняття рішень, надає можливість підвищити точність ідентифікації за рахунок нелінійності площині розбиття та виконати більш оптимальне налаштування класифікатору. Метод є особливо ефективним за наявності вихідних даних, які мають високі кореляційні коефіцієнти, так як поєднує їх в один або декілька багатомірних критеріїв. Проведено оцінку точності та оперативності розробленого методу ідентифікації стану комп'ютерної системи.

Результати. Розроблений метод реалізований програмно і досліджений під час розв'язання задачі ідентифікації стану функціонування комп'ютерної системи.

Висновки. Проведені експерименти підтвердили працездатність запропонованого методу, що надає можливість рекомендувати його для практичного використання з метою підвищення точності ідентифікації стану комп'ютерної системи. Перспективи подальших досліджень можуть полягати в розробці ансамблю дерев рішень.

КЛЮЧОВІ СЛОВА: комп'ютерна система, аномальний стан, ідентифікація, дерево рішень, кластеризація, алгоритм DBSCAN, гіперсфера.

УДК 004.8

РАЗРАБОТКА МЕТОДА ИДЕНТИФИКАЦИИ СОСТОЯНИЯ КОМПЬЮТЕРНОЙ СИСТЕМЫ НА ОСНОВЕ ДЕРЕВА РЕШЕНИЙ С МНОГОМЕРНЫМИ УЗЛАМИ

Гавриленко С. Ю. – д-р техн. наук, профессор, профессор кафедры «Вычислительная техника и программирование», Национальный технический университет «Харьковский политехнический институт», Харьков, Украина.

Челак В. В. – аспирант кафедры «Вычислительная техника и программирование», Национальный технический университет «Харьковский политехнический институт», Харьков, Украина.

Семенов С. Г. – д-р техн. наук, профессор, заведующий кафедры «Вычислительная техника и программирование», Национальный технический университет «Харьковский политехнический институт», Харьков, Украина.

АННОТАЦИЯ

Актуальность. Рассмотрена задача идентификации состояния компьютерной системы. Объектом исследования является процесс идентификации состояния компьютерной системы. Предметом исследования являются методы построения решений для идентификации состояния КС

Цель. Разработка метода построения деревьев решений для идентификации состояния компьютерной системы.

Метод. Предложен новый метод построения дерева решений, сочетающий классическую модель построения дерева решений и основанный на плотности метод пространственной кластеризации (DBSCAN). Результаты моделирования показали, что предложенный метод позволяет уменьшить количество ветвлений в дереве решений, что позволит повысить оперативность идентификации состояния компьютерной системы. Использование принадлежности к гиперсферам в качестве критерия принятия решений позволяет повысить точность идентификации за счет нелинейности плоскости разбиения и выполнить более оптимальную настройку классификатора. Метод особенно эффективен при наличии исходных данных, имеющих высокие корреляционные коэффициенты, так как объединяет их в один или несколько многомерных критериев. Проведена оценка точности и оперативности разработанного метода идентификации состояния компьютерной системы.

Результати. Разработанный метод реализован в виде программного обеспечения и исследован при решении задачи идентификации состояния функционирования компьютерной системы.

Выводы. Проведенные эксперименты подтвердили работоспособность предлагаемого метода, что позволяет рекомендовать его для практического использования с целью повышения точности идентификации состояния компьютерной системы. Перспективы дальнейших исследований могут состоять в разработке ансамбля деревьев решений.

КЛЮЧЕВЫЕ СЛОВА: компьютерная система, аномальное состояние, идентификация, дерево решений, кластеризация, алгоритм DBSCAN, гиперсфера.

ЛІТЕРАТУРА / ЛИТЕРАТУРА

1. Daniel Schatz. Towards a More Representative Definition of Cyber Security / Schatz Daniel, Bashroush Rabih, Wall Julie // The Association of Digital Forensics, Security and Law (ADFSL). – 2017. – Vol. 12, No. 2. – P. 53–74. DOI: 10.15394/jdfsl.2017.1476
2. Farooq Anjum. Intrusion Detection Systems / Farooq Anjum and Petros Mouchtaris // Security for Wireless Ad Hoc Networks. – Wiley, 2007. – P. 120–159. DOI: 10.1002/9780470118474.ch5
3. Kelleher J. Fundamentals of Machine Learning for Predictive Data Analytics: Algorithms, Worked Examples and Case Studies / J. Kelleher, B. Namee, A. Archi. – Dublin : The MIT Press, 2015. – 642 p.
4. Cybersecurity data science: an overview from machine learning perspective / [Iqbal H. Sarker, A. S. M. Kayes,

DOI 10.15588/1607-3274-2022-2-11

Shahriar Badsha et al.] // Journal of Big Data. – 2020. – Vol. 7 (41). – 29 p. DOI: 10.1186/s40537-020-00318-5
5. Xavier Larriva-Novo. Efficient Distributed Preprocessing Model for Machine Learning-Based Anomaly Detection over Large-Scale Cybersecurity Datasets / [Xavier Larriva-Novo, Mario Vega-Barbas, Victor A. Villagra et al.] // Applied Sciences. – 2020. – Vol. 10. – 19 p. DOI: 10.3390/app10103430
6. Identification of the state of an object under conditions of fuzzy input data / [S. Semenov, O. Sira, S. Gavrylenko, N. Kuchuk] // Eastern-European Journal of Enterprise Technologies. – 2019. – Vol. 1, No. 4 (97). – P. 22–29. DOI: 10.15587/1729-4061.2019.157085
7. Alpaydin E. Introduction to Machine learning / E. Alpaydin. – London : The MIT Press, 2010. – 400 p.
8. Bogumil Kaminski. A framework for sensitivity analysis of decision trees / B. Kaminski, M. Jakubczyk, P. Szufel // Central European Journal of Operations Research. – 2018, Vol. 26. – P. 135–159 DOI: 10.1007/s10100-017-0479-6
9. Gavrylenko S. The ensemble method development of classification of the computer system state based on decision trees / S. Gavrylenko, I. Sheverdin, M. Kazarinov // Advanced Information Systems. – 2020. – P. 5–10. DOI:10.20998/2522-9052.2020.3.01
10. Субботін С. О. Подання й обробка знань у системах штучного інтелекту та підтримки прийняття рішень / С. О. Субботін. – Запоріжжя : ЗНТУ, 2008. – 341 с.
11. Субботин С. А. Построение деревьев решений для случая малоинформативных признаков / С. О. Субботин // Радиоэлектроника, информатика, управление. – 2019. – № 1. – С. 122–130. DOI: 10.15588/1607-3274-2019-1-12
12. Sergei Mitrofanov. An Approach to Training Decision Trees with the Relearning of Nodes / S. Mitrofanov and E. Semenkin // 2021 International Conference on Information Technologies (InfoTech). – 2021 .– P. 1–5. DOI: 10.1109/Info Tech52438.2021.9548520
13. Wang S. Adapting naive Bayes tree classification / S. Wang, L. Jiang, C. Li // Knowledge and Information system. – 2015. – Vol. 44, № 1. – P. 77–89. DOI: 10.1007/s10115-014-0746-y
14. Kornienko Y. A hybrid algorithm for decision tree generation / Y. Kornienko, A. Borisov // International Scientific Journal of Computing. – 2004. – Vol. 3, Issue 3. – P. 51–57. DOI: 10.47839 /ijc.3.3.305
15. Efficient Construction of Decision Trees by the Dual Information Distance Method / [Irad Ben-Gal, Alexandra Dana, Niv Shkolnik, Gonen Singer] // Quality Technology & Quantitative Management. – 2014. – Vol. 11, № 1. – P. 133–147. DOI: 10.1080/16843703.2014.11673330
16. Geurts P. Extremely randomized trees / P. Geurts, D. Ernst, L. Wehenkel // Machine Learning. – 2006. – Vol. 63, No. 1. – P. 3–42. DOI:10.1007/s10994-006-6226-1
17. Kesinee Boonchuay. Boundary expansion algorithm of a decision tree induction for an imbalanced dataset / Kesinee Boonchuay, Krung Sinapiromsaran, Chidchanok Lursinsap // Songklanakarin Journal of Science and Technology (SJST). – 2017. – Vol. 39, No. 5. – P. 665–673. DOI: 10.14456/sjst-psu.2017.82
18. Quinlan J. R. Induction of Decision Trees Machine Learning. / J. R. Quinlan // Kluwer Academic Publishers. – 1986. – № 1. – P. 81–106.
19. A comparative study of decision tree ID3 and C4.5 / [B. Hssina, A. Merbouha, H. Ezzikouri, M. Erritali] // International Journal of Advanced Computer Science and Applications. – 2014. – Vol. 4 (2). – P. 13–19. DOI: 10.14569/SpecialIssue. 2014.040203
20. Idris Mochamad. Implementation of C4.5 Algorithm and Forward Chaining Method for Higher Education Performance Analysis / Idris Mochamad, Mustafid, Suseno Jatmiko Endro // The 4th International Conference on Energy, Environment, Epidemiology and Information System (ICENIS 2019). – 2019. – Vol. 125. DOI: 10.1051/e3sconf/201912521002
21. Painsky A. Cross-Validated Variable Selection in Tree-Based Methods Improves Predictive Performance / A. Painsky, S. Rosset // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2017. – Vol. 39, No. 11. – P. 2142–2153. DOI: 10.1109/TPAMI.2016.2636831.
22. Decision tree-based Design Defects Detection / [M. Maddeh, S. Ayouni, S. Alyahya and F. Hajjej] // IEEE Access. – 2021. – Vol. 9. – P. 71606–71614. DOI: 10.1109/ACCESS.2021. 3078724.
23. Gavrylenko S. Ensemble approach based on bagging and boosting for Identification the Computer System State / S. Gavrylenko, V. Chelak, O. Hornostal // Proceedings of the 31th International Scientific Symposium Metrology and Metrology Assurance. – Sozopol, IEEE Access, 2021. DOI:10.1109/MMA52675.2021.961094

DOI 10.15588/1607-3274-2022-2-11