Problems of identification in information systems

UDC 004.9:519.2 **doi: 10.20998/2522-9052.2020.3.01**

Svitlana Gavrylenko[1], Illia Sheverdin[1], Michael Kazarinov[2]

[1] National Technical University «Kharkiv Polytechnic Institute», Kharkiv, Ukraine
[2] Northeastern Illinois University, Chicago, USA

THE ENSEMBLE METHOD DEVELOPMENT OF CLASSIFICATION OF THE COMPUTER SYSTEM STATE BASED ON DECISIONS TREES

Abstract. The subject of this article is exploration of methods for identifying the status of a computer system. The purpose of the article is development of a method for classifying a computer system anomalous state based on ensemble methods. Task: To investigate the usage of algorithms for building decision trees: REPTree, Random Tree, J48, HoeffdingTree, DecisionStump and bagging and boosting decision tree ensembles to identify a computer system anomalous state by analyzing operating system events. The methods used are artificial intelligence, machine learning and ensemble classification methods. The following results were obtained: the methods of identifying the computer systems anomalous state based on ensemble methods were investigated, namely, bagging, boosting, and classifiers: REPTree, Random Tree, J48, HoeffdingTree, DecisionStump to identify a computer system anomalous state. The different classifiers set and classifiers ensembles were developed. Training and cross-validation on each algorithm was performed. The developed classifiers performance has been evaluated. The research suggests an ensemble method of a computer system state classifying based on the J48 decision tree algorithm. **Conclusions.** The scientific novelty of the obtained results consists in creating an ensemble method for classifying the state of a computer system based on a decision tree, which makes it possible to increase the reliability and speed of classification.

Keywords: computer system; decision trees; ensemble methods; boosting; bagging; operating system events; anomalous state.

Introduction

An ordinary feature of the modern country is the use of computer systems in almost all sectors of the national economy. The secure function of such systems is a priority and determines the role of the country in the world. This is why the study of identifying the state of computer systems methods and means is an urgent task.

The computer system is characterized by a huge number of functioning indicators. One of the most common methods of analysing large amount of data (data mining) is machine learning. Machine learning algorithms are designed to work directly with huge amounts of information.

Problem analysis and scientific publications. Issues related to problems encountered when dealing with large amounts of information, especially when used to evaluate the state of a computer system, are discussed in [1-5]. Complex mathematical algorithms based on machine learning methods are used for data analysis and classification: classical methods [6], reinforcement learning methods [7], decision trees and ensemble methods [8,9], neural networks and deep learning [10] and so on.

Different approaches to solving the problems proposed in the work [11-13]. One of the classification effective method is ensemble, which are based on a basic classifier set, which results are combined and operate the aggregated classifier prediction [14].

In [15-17], a comparative investigation of various building ensemble methods was performed. Nevertheless, this work does not include the effectiveness of using different decision trees methods in conjunction with different ensemble decision-making methods.

Algorithms analysis for building decision trees and decision tree ensembles

One of the main tasks of machine learning is to learn from use cases, according to which the object M is studying. Objects with M are described by a set of attributes $\{x_1, ..., x_n\}$. Each object $S \in M$ is represented by a vector of the length n, where the coordinate j is equal to the attribute value x_j for the object S. It was set a set of 'answers' and a selection of objects (precedents) $T = \{S_1, ..., S_m\}$ of M so that for each object $S_i \in T$ there is an 'answer' y_i, $y_i \in Y$. By the selection T, we need to build an algorithm $AT: M \to Y$, which puts in correspondence to each object the value y with Y.

One of the main types of case-based learning is the classification problem for which the 'response' y for an object S from M is a class label, including a binary classification where $Y = \{-1, +1\}$.

One of the well-known tools for solving case-based learning problems is decision trees. The procedure of building a classical decision tree (DT) is iterative. As a rule, to build the next vertex of the tree, the attribute that best meets branching criteria is selected. The values of this attribute are used for branching. Then the specified procedure is repeated for each leaf.

One of the first algorithms for building a decision tree is the ID3 algorithm [18]. The idea of the ID3 algorithm is to divide the selection into two parts until each part contains objects of only one class. An improved version of the ID3 algorithm is the C4.5 algorithm, which has been added the pruning procedure, the ability to work with numeric attributes, and the ability to build a tree with an incomplete training

sample that lacks the values of some attributes. The C4.5 algorithm selects an attribute based on the normalized information gain or information entropy (Gain Ratio):

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

where n – number of classes in the original subset, N_i – the number of samples of i class, N – the total number of samples in the subset.

The best attribute for splitting A_j is the one that provides the maximum reduction in the entropy of the obtained subsets relative to the parent set. However, in practice, it is more convenient to use the concept of information, which is the reverse of entropy. Then the best partition attribute is the result, which provides the maximum increase in information of the resulting node relative to the original one:

$$\text{Gain}(A) = \text{Info}(S) - \text{Info}(S_A),$$

where Info (S) – information related to a subset of S before splitting, Info (S_A) – information associated with a subset of S after splitting by attribute A.

The J48 algorithm is the analog of the C4.5 algorithm, which is implemented in Java in the Weka application.

The Decision Stump algorithm is a single-level tree with a statistical branching criterion [19]. This tree has a root vertex that is connected by an edge to each branch of the tree. Decision Stump considers each attribute of x sequentially and builds a separate tree for this attribute. Possible option: 1) for each value of attribute x, one leaf is built; 2) the number a (threshold) is selected and two leaves are built, in one of which $x < \alpha$, and in the second $x \geq \alpha$; 3) the set of values of attribute x is divided into intervals and a tree is built with the number of leaves equal to the number of these intervals.

To select the optimal rule, we use the partition quality estimation function, which is formalized in the Gini index. If the data set S contains data from n classes, then the Gini index is defined as:

$$Gini(S) = 1 - \sum_{i=1}^{n} p_i^2 \ ,$$

where p_i – probability (relative frequency) of class i in S.

If the set M is split into two parts S_1 and S_2 with the number of samples in each N1 and N2, respectively, then the partition quality indicator will be equal to:

$$Gini_{split}(S) = \frac{N_1}{N_2} \cdot Gini(S_1) + \frac{N_2}{N1} \cdot Gini(S_2).$$

The best partition is where $\text{Gini}_{\text{split}}(S)$ is minimal.

The REPTree (Reduced Error Pruning Tree) algorithm builds binary trees for classification and regression problems using entropic or statistical branching criteria, respectively. This algorithm was first proposed by Quinlan in 1987 [20]

HoeffdingTree is an incremental decision tree induction algorithm that can learn from data flows, assuming that the distribution generation examples do not change over time [21]. Hoeffding trees take advantage of the fact that a small sample is often enough to select the optimal splitting attribute. This idea is supported mathematically by the Hoeffding estimate, which quantifies the number of observations needed to evaluate some statistical data of a given accuracy.

RandomTree is a tree building algorithm that considers k randomly selected attributes for each node [22]. The algorithm makes it possible to estimate the probability value of a class based on a set selection by combining multidimensional linear regression and one-dimensional smoothing. Thus, a nonlinear problem is reduced to solving a sequence of linear problems.

To improve the efficiency of various algorithms for building decision trees are used ensembles, in which several models (weak students or basic models) are learning to solve the same problem and are combined to improve performance. Today, the most popular are meta-algorithms that aimed to combining weak students such as begging, boosting.

Begging is an ensemble of homogeneous weak classifiers that learn in parallel and independently from each other on different random samples from the source data using the same decision-making algorithm. The results are then combined, following some deterministic averaging process. At the same time, classifiers do not correct each other's mistakes, but compensate for them when voting [23]. The main idea of Begging is to select several independent models and find the optimal average value to get the model with the smallest spread. The most common begging algorithm is the Random Forest algorithm.

Boosting combines homogeneous weak classifiers, training them consistently in an adaptive way (the weak classifier depends on the previous ones) and combines them, following a deterministic strategy [24]. By reducing the variance, accuracy increases and the number of matching and training operations decreases. The most common boosting algorithms are AdaBoost, LogitBoost, and Gradient boosting.

Development of an ensemble classification method

In the Windows operating systems all events can be divided into 4 main types: process communication events, file system interaction events, Internet connection events, and operating system registry interaction events.

The developed software made it possible to collect changes in the state of the computer system. Weka software [25] was chosen for an analysis of the system state and quality assessment of methods for building decision trees and classification algorithms. It provides a set of virtualization tools and components for data mining and solving forecasting problems. The functionality of the program allows to perform the task of data analysis, clustering, regression analysis etc.

For further analysis of the computer system state were used classifiers based on boosting and begging. Each of the classifiers was investigated using different types of the above decision trees.

For evaluating the quality of classifiers, the following criteria were selected as the main:

1. The number of correctly classified objects in absolute and percentage value (Correctly Classified Instances).

2. The number of incorrectly classified objects in absolute and percentage value (Incorrectly Classified Instances).

3. The Kappa coefficient (Kappa statistic). The Kappa-Cohen coefficient is a metric that compares the accuracy of an observation with the expected accuracy. Kappa statistics are used not only to evaluate a single classifier, but also to evaluate classifiers among themselves. A value greater than 0 means that your classifier works better than the probability that it will function properly:

$$\kappa = \frac{p_e - p_0}{1 - p_0} = 1 - \frac{1 - p_0}{1 - p_e}$$

where p_0 – is the relative observed agreement among raters and p_e – is the hypothetical probability of chance agreement.

4. Mean absolute error (MAE). MAE-a value used to measure how close forecasts are to possible results:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|f_i - y_i| = \frac{1}{n}\sum_{i=1}^{n}|e_i|$$

where f_i –is the prediction and y_i – is the true value.

5. Root mean squared error (RMSE) is the standard deviation for sampling differences between predicted and observed values:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\widehat{\theta}_i - \theta_i\right)^2}$$

where the vector of observed values of the variable being predicted – θ_i, with $\widehat{\theta}_i$ – being the predicted values.

6. Relative absolute error (RAE) – the absolute difference between the measurement result and the true value of the measured value, which is a sign of the quality of the measurement:

$$RAE = \sum_{i=0}^{n}(P - T_i) \Big/ \sum_{i=0}^{n}(T_i - \overline{T_i}),$$

where P is the predicted value; T_i is the target value for case i; and is given by the formula:

$$\overline{T} = \frac{1}{n}\sum_{j=1}^{n} T_j.$$

7. Root relative squared error (RRSE). The RRSE takes the General squared error and normalizes it by dividing by the General squared error of a simple predicate. By taking the square root of the relative square error, we can reduce the error to the same size as the predicted value:

$$RRSE = \sqrt{\sum_{i=0}^{n}(P - T_i)^2 \Big/ \sum_{i=0}^{n}(T_i - \overline{T_i})^2}.$$

8. Total Number of Instances.

9. Testing time.

10. Learning time.

Studies of various classifiers of ensemble methods in conjunction with various algorithms for building decision trees allow us to obtain the following results.

The results of the Boosting algorithm in conjunction with various algorithms for building decision trees are presented on the Table 1.

As can be seen from the Table 1, the meta-algorithm of machine learning is an effective ensemble method for evaluating the state of CS. The best results are obtained when using decision trees: J48, REPTree, and Random Tree.

The results of the Bagging algorithm in conjunction with various algorithms for building decision trees are shown in Table 2.

As can be seen from the Table.2 Bagging is also an effective ensemble method for assessing the state of the CS. The best results are obtained when using decision trees: J48, REPTree, and Random Tree.

It is shown on the Fig. 1 and 2 histograms of the absolute error of classification and time of CS testing when using decision trees based on algorithms: J48, REPTree, and Random Tree.

As can be seen from the histogram on Fig. 1, at the learning stage, the accuracy of the classification of the ensemble classifier based on boosting is slightly lower compared to the accuracy of the classification of the ensemble classifier based on Begging. However, the testing time for this classification is slightly better for an ensemble classifier based on boosting.

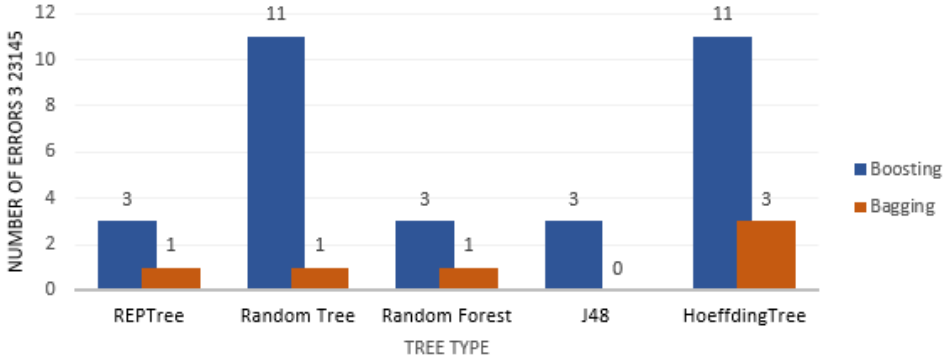

Fig. 1. Absolute error classification histogram

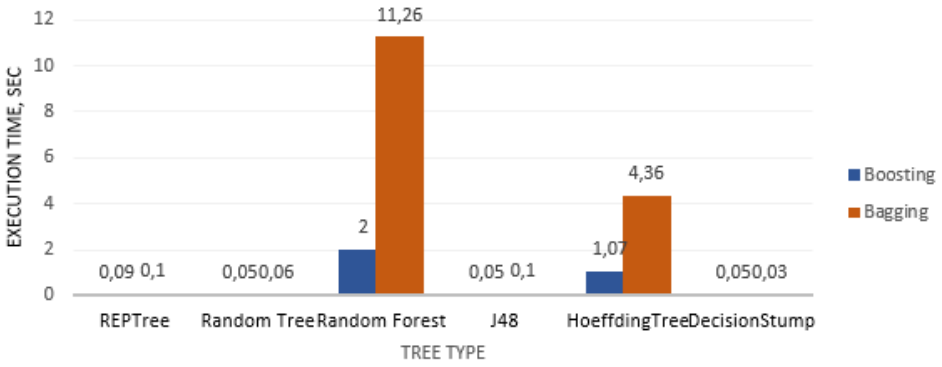

Fig. 2. Testing time histogram

In order to make sure that the decision-making algorithm is stable and accurate, Cross Validation was performed and the results were checked on the test data. Cross Validation was performed as follows. The original data was divided into 10 parts. Then each of the parts was used for testing and training: once for testing and 9 times for training at a time. Ten obtained results were averaged. Later, after performing a 10-fold cross-check and calculating the evaluation results, the training algorithm was obtained for the last 11th time on the entire data set. Afterwards, the resulting model was built.

Table 1 – **Boosting algorithm results**

Boosting	REPTree	Random Tree	J48	HoeffdingTree	Decision Stump
Correctly Classified Instances	23142	23134	23142	23134	20827
Incorrectly Classified Instances	3	11	3	11	2318
Correctly %	99,99	99,9525	99,987	99,9525	89,9849
Incorrectly %	0,019	0,0475	0,0131	0,0475	10,0151
Kappa statistic	0,9997	0,9988	0,9997	0,9988	0,7313
Mean absolute error	0	0,0001	0	0	0,0575
Root mean squared error	0,0032	0,0061	0,0029	0,0069	0,1372
Relative absolute error	0,0339	0,1491	0,0339	0,1242	150,2427
Root relative squared error	2,3429	4,3693	2,1164	4,9879	99,1314
Total Number of Instances	23145	23145	23145	23145	23145
Learning Time	0,1401	0,0601	0,1301	1,5103	0,1801
Testing Time	0,0902	0,0502	0,0502	1,0702	0,0503

Table 2 – **Bagging algorithm results**

Bagging	REPTree	Random Tree	J48	HoeffdingTree	Decision Stump
Correctly Classified Instances	23144	23144	23145	23142	20831
Incorrectly Classified Instances	1	1	0	3	2314
Correctly %	99,9957	99,9957	100	99,987	90,0022
Incorrectly %	0,0043	0,0043	0	0,0131	9,9978
Kappa statistic	0,9999	0,9999	1	0,9997	0,7317
Mean absolute error	0	0	0	0	0,0142
Root mean squared error	0,0019	0,0023	0,0017	0,0028	0,0841
Relative absolute error	0,0328	0,0853	0,0328	0,0508	37,0307
Root relative squared error	1,3592	1,647	1,2477	2,0455	60,882
Total Number of Instances	23145	23145	23145	23145	23145
Learning Time	0,6502	0,21	0,37	5,41	0,18
Testing Time	0,1	0,06	0,1	4,36	0,03

Thus, the results of Cross Validation also confirmed the high quality of decision trees based on algorithms: J48, REPTree.

The classifier based on decision trees of the Random Tree algorithm has slightly worse results. But, as shown by the results of the cross validation (Fig. 3) the classification accuracy of ensemble classifier on the basis of bagging is a little lower compared with the classification accuracy of ensemble classifier based on boosting.

Conclusion

This article discusses methods of identifying the abnormal state of computer systems based on ensemble methods.

The following events were used as outgoing data for evaluating the state of the computer system: process communication events, file system interaction events, Internet connection events, operating system registry interaction events.

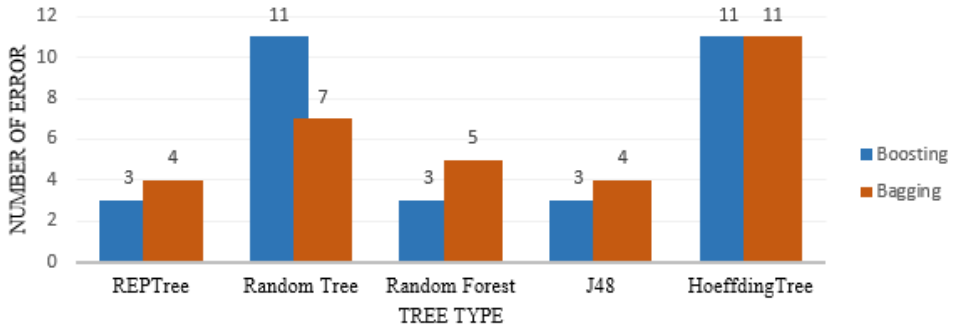

Fig. 3. Absolute error of classification after Cross Validation histogram

Ensemble meta-algorithms for machine begging and boosting that combine classifiers based on decision trees: REPTree, Random Tree, J48, HoeffdingTree, and DecisionStump were evaluated. Meta-algorithms were trained and cross-qualified. The effectiveness of the developed classifiers was evaluated according to the following criteria: the number of correctly and incorrectly classified objects in absolute and percentage values, Kappa coefficient, absolute error, average absolute error, standard error, relative squared error, training and testing time.

It was found that the ensemble meta-algorithms of machine learning begging and boosting are effective tools for evaluating the state of a computer system. The best results were obtained by combining classifiers based on algorithms for building decision trees: J48, REPTree, and Random Tree.

The most effective, both for boosting and begging ensembles, is a classifier based on the J48 decision tree construction algorithm.

The classification accuracy and testing time of computer systems based on it is almost the same for both ensemble classifiers.

Thus, to identify the state of a computer system based on research results, a method based on the j48 classifier was proposed, which is an effective tool for ensemble meta-algorithms of begging and boosting.

Further studies of computer system state identification technologies can be performed in computer intrusion prevention systems.

References

1. Korchenko, A.A. (2012), “Sistema vyyavleniya anomalnogo sostoyaniya v kompyuternykh setyakh”, *Bezpeka informacziyi*, Kyiv, Vol. 2 (18), pp. 80-84.
2. Chowdhury, M. (2017), “Malware Analysis and Detection Using Data Mining and Machine Learning Classification”, *International Conference on Applications and Techniques in Cyber Security and Intelligence*, ATCI, pp. 266-274.
3. Gavrilenko, S.Yu. (2019), “Metodika vidboru sistemi pokaznikiv dlya identifikacziyi stanu komp’yuternoyi sistemi kritichnogo zastosuvannya”, *Radioelektronni i komp’yuterni sistemi*, Vol. 2 (90), pp. 127-135, DOI: https://doi.org/10.32620.reks.2019.2.12
4. Krivenko, M.P. and Vasilev, V.G. (2013), *Metody klassifikaczii dannykh bolshoj razmernosti*, IPI RAN, Moscow, 204 p.
5. Bargesyan, A.A. (2007), *Tekhnologii analiza dannykh: Data Mining, Visual Mining, Text Mining, OLAP*, BKhV-Peterburg, Sankt-Peterburg, 384 p.
6. Vipin, Kumar (2009), *The Top Ten Algorithms in DataMining*, Taylor & Francis Group, LLC, 2006 p.
7. Satton, Richard and Barto, Endryu G. (2020), *Obuchenie s podkrepleniem = Reinforcement Learning*, DMK press, Moscow, 552 p.
8. Kaftannikov I.L. and Parasich, A.V. (2015), “Osobennosti primeneniya derevev reshenij v zadachakh klassifikaczii”, *Vestn. YuUrGU. Ser. «Kompyuternye tekhnologii, upravlenie, radioelektronika»*, Vol. 15, No. 3, pp. 26-32.
9. Cha, Zhang (2012), *Ensemble Machine Learning. Methods and Applications*, Springer, London, 329 p.
10. Tarkhov, D.A. (2014), *Nejrosetevye modeli i algoritmy*, Radiotekhnika, Moscow, 352 p.
11. Vyugin, V.V. (2013), *Matematicheskie osnovy mashinnogo obucheniya i prognozirovaniya*, MCzNMO, Moscow, 304 p.
12. Marchenko, O.O. and Rossada, T.V. (2017), *Aktualni problemi Data Mining*, Kyiv, 150 p.
13. Bolshakov, A.S. and Gubankova, E.V. (2020), “Obnaruzhenie anomalij v kompyuternykh setyakh s ispolzovaniem metodov mashinnogo obucheniya”, *Telekommunikaczionnye ustrojstva i sistemy*, Vol. 20 (1), pp. 37-42.
14. Joseph, Rocca and Baptiste, Rocca (2020), “Ensemble methods: bagging, boosting and stacking”, available at: https://towardsdatascience.com/ensemble-methods-bagging-boosting-and-stacking-c9214a10a205.
15. Bauer, E. (1999), “An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants”, *Machine Learning*, Springer Link, pp. 105–139.
16. Kristína, Machová, Miroslav, Puszta, František, Barčák, and Peter, Bednár (2006), “A Comparison of the Bagging and the Boosting Methods Using the Decision Trees Classifiers”, *Computer Science and Information Systems*, Vol. 3(2), pp.57-72.
17. (2020), *Metody postroeniya derevev reshenij v zadachakh klassifikaczii v Data Mining*, available at: https://ami.nstu.ru/~vms/lecture/data_mining/trees.htm.
18. Mitchell, N. and Tom, Michael (1997), *Machine Learning*, McGraw-Hill, New York, 432 p.
19. Iba, Wayne and Langley, Pat. (1992), “Induction of One-Level Decision Trees”, *ML92 – Proceedings of the Ninth International Conference on Machine Learning*, Morgan Kaufmann, San Francisco, pp. 233-240.

20. Zontul, M., Aydin, F., Dogan, G., Sener, S and Kaynar, O. (2013), “Wind speed forecasting using REPtree and bagging methods in kirklareli-turkey”, *Journal of Theoretical and Applied Information Technology*, Vol. 56(1), pp.17-29.
21. (2020), *Class HoeffdingTree*, available at: https://weka.sourceforge.io/doc.dev/weka/classifiers/trees/HoeffdingTree.html.
22. (2020), *Class RandomTree*. available at: http://weka.sourceforge.net/doc.dev/weka/classifiers/trees/RandomTree.html.
23. Krivenko, M.P. and Vasilev V.G. (2013), *Metody klassifikac dannykh bolshoj razmernosti*, IPI RAN, 2013, Moskow, 204 p.
24. Myuller, A. (2017), *Vvedenie v mashinnoe obuchenie s pomoshhyu Python. Rukovodstvo dlya speczialistov po rabote s dannymi*, Alfa-kniga, Moskow, 480 p.
25. (2020), *The workbench for machine learning*, available at: https://www.cs.waikato.ac. nz/ml/weka

Received (Надійшла) 17.06.2020
Accepted for publication (Прийнята до друку) 09.09.2020

ABOUT THE AUTHORS / ВІДОМОСТІ ПРО АВТОРІВ

Гавриленко Світлана Юріївна – доктор технічних наук, доцент, професор кафедри обчислювальної техніки та програмування, Національний технічний університет “Харківський політехнічний інститут”, Харків, Україна;
Svitlana Gavrylenko – PhD (Technical), Associate Professor, Professor of Department of "Computer Engineering and Programming", National Technical University «Kharkiv Polytechnic Institute», Kharkiv, Ukraine;
e-mail: gavrilenko08@gmail.com; ORCID ID: https://orcid.org/0000-0002-6919-0055.

Шевердін Ілля Валентинович – аспірант, кафедра обчислювальної техніки та програмування, Національний технічний університет “Харківський політехнічний інститут”, Харків, Україна;
Illia Sheverdin – PhD Student of Department of "Computer Engineering and Programming", National Technical University «Kharkiv Polytechnic Institute», Kharkiv, Ukraine;
e-mail: illia.sheverdin@gmail.com, ORCID ID: https://orcid.org/0000-0002-7881-0658.

Казарінов Міхаел – викладач, кафедра інформатики, Північно-східний Іллінойський університет, Чикаго, штат Іллінойс, Сполучені Штати Америки;
Michael Kazarinov – teacher, Computer Science Department, Northeastern Illinois University, Chicago, IL US;
e-mail: kazarinov@gmail.com; ORCID ID: https://orcid.org/0000-0002-0790-5262

Розробка ансамблевого методу класифікації стану комп’ютерної системи на основі дерев рішень

С. Ю. Гавриленко, І. В. Шевердін, М. Казарінов

Анотація. Предметом статті є дослідження методів ідентифікації стану комп’ютерної системи. **Метою** статті є розробка методу класифікації аномального стану комп’ютерної системи на основі ансамблевих методів. **Завдання**: дослідити використання алгоритмів побудови дерев рішень: REPTree, Random Tree, J48, HoeffdingTree, DecisionStump та ансамблів дерев рішень на основі беггінгу та бустінгу для ідентифікації аномального стану комп’ютерної системи на основі аналізу подій операційної системи. Використовуваними **методами** є: методи штучного інтелекту, машинного навчання та ансамблеві методи класифікації. Отримано такі **результати**: досліджено методи ідентифікації аномального стану комп’ютерних систем на базі ансамблевих методів, а саме беггінгу, бустінгу та класифікаторів: REPTree, Random Tree, J48, HoeffdingTree, DecisionStump для ідентифікації аномального стану комп’ютерної системи. Розроблено набір різних класифікаторів та ансамблів класифікаторів, виконано їх навчання та кросвалідацію. Виконано оцінку ефективності розроблених класифікаторів. За результатами досліджень запропоновано ансамблевий метод класифікації стану комп’ютерної системи на основі алгоритму побудови дерева рішень J48. **Висновки. Наукова новизна** отриманих результатів полягає в створенні ансамблевого методу класифікації стану комп’ютерної системи на основі дерева рішень, що надає можливість підвищити надійність та швидкість класифікації.

Ключові слова: комп’ютерна система; дерева рішень; ансамблеві методи; бустинг; беггінг; події операційної системи; аномальний стан.

Разработка ансамблевого метода классификации состояния компьютерной системы на основе деревьев решений

С. Ю. Гавриленко, И. В. Шевердин, М. Казаринов

Аннотация. Предметом статьи является исследование методов идентификации состояния компьютерной системы. **Целью** статьи является разработка метода классификации аномального состояния компьютерной системы на основе ансамблевых методов. **Задача**: исследовать использование алгоритмов построения деревьев решений: REPTree, Random Tree, J48, HoeffdingTree, DecisionStump и ансамблей деревьев решений на основе беггинга и бустинга для идентификации аномального состояния компьютерной системы на основе анализа событий операционной системы. Используемыми **методами** являются: методы искусственного интеллекта, машинного обучения и ансамблевые методы классификации. Получены следующие **результаты**: исследованы методы идентификации аномального состояния компьютерных систем на базе ансамблевых методов, а именно беггинга, бустинга и классификаторов: REPTree, Random Tree, J48, HoeffdingTree, DecisionStump для идентификации аномального состояния компьютерной системы. Разработан набор различных классификаторов и ансамблей классификаторов, выполнены их обучение и кросвалидация. Выполнена оценка эффективности разработанных классификаторов. По результатам исследований предложено ансамблевый метод классификации состояния компьютерной системы на основе алгоритма построения дерева решений J48. **Выводы**. **Научная новизна** полученных результатов заключается в создании ансамблевого метода классификации состояния компьютерной системы на основе дерева решений, дает возможность повысить надежность и скорость классификации.

Ключевые слова: компьютерная система; деревья решений; ансамблевые методы; бустинг; беггинг; события операционной системы; аномальное состояние.