Intellectualization of Decision Support Systems Based on Cloud Computing

Viktor Krasnoproshin
Faculty of Applied Mathematics and Computer Science
Belarusian State University
Minsk, Belarus
Email: krasnoproshin@bsu.by

Vadim Rodchenko and Anna Karkanitsa
Faculty of Mathematics and Informatics
Yanka Kupala State University of Grodno
Grodno, Belarus
Email: rovar@grsu.by, a.karkanica@grsu.by

***Abstract*—The implementation of artificial intelligence is a priority direction for the development of information management systems. Decision support systems (DSS) are information processing systems that emerged from the integration of automated management systems and database management systems. Their purpose is to assist users in making decisions on complex, unstructured, or weakly structured problems in various fields of human activity. Various methods can be used for analysis and generating recommendations by the system, including information retrieval, simulation modeling, case-based reasoning, situational analysis, cognitive modeling, machine learning, artificial neural networks, data mining, knowledge discovery in databases, genetic algorithms, and others. This article proposes a method for the intellectualization of decision support systems based on the use of OSTIS Technology, intelligent agents, and cloud computing. The implementation of autonomous agents for discovering hidden interpretable patterns in data and subsequent classifier construction is described. The effectiveness of the method is confirmed by the results of a numerical experiment based on model data.**

***Keywords*—decision support systems, cloud computing, knowledge discovery, data mining, instance-based learning**

I. Introduction

Solving problems related to improving the efficiency of using data warehouses and databases is one of the priority areas of computer science. This is a science that deals with the development of methods for processing, analyzing, and evaluating information, ensuring its effective use for decision-making [1], [2]. The development and implementation of artificial intelligence technologies are the most important factors in the progress of creating and using information systems [3]–[6]. Currently, the drivers of intellectualization in these systems are machine learning, artificial neural networks, and data mining [6]–[12].

The development of new approaches and technologies in machine learning largely determines progress in the field of artificial intelligence [13], [14]. At its core, machine learning is associated with identifying empirical patterns in data based on observed instances. Sets of positive and negative examples related by an unknown pattern are analyzed, and a classification algorithm is developed to separate the examples into classes [15], [16].

In machine learning, a situation has arisen where the construction of a classification algorithm follows a standard scheme: a description space of objects is given, in which a separating surface needs to be constructed. Learning occurs during the solution of an optimization problem and involves the construction of decision rules that provide an extremum for a certain criterion. A class of decision rules is predefined up to parameters, and learning is reduced to finding parameter values that provide an extremum for the selected criterion [17], [18].

The learning process involves multiple executions of the "training + testing" scenario. Within this process, through positive and negative reinforcement, the necessary patterns of classifier behavior are formed. However, learning can only be conducted in an automated mode, and the constructed classification algorithm represents a "black box" [19], [20].

An alternative method of learning by precedents can be implemented based on the properties generality principle. The main assumption of this principle is that instances of the same class share common properties reflecting their similarity. The learning process is aimed at investigating the properties of feature combinations from an a priori dictionary to construct feature spaces where class patterns do not intersect.

This article proposes a method for the intellectualization of decision support systems based on the use of OSTIS Technology, intelligent agents, and cloud computing. An original method for implementing autonomous agents to discover hidden interpretable patterns within training data and to construct a classifier is presented. The effectiveness of the method is confirmed by the results of a numerical experiment.

II. Decision Support System and Knowledge Discovery Classification

A decision support system (DSS) is an information system designed to help specialists quickly solve managerial and organizational problems that may arise unexpectedly and require prompt responses. DSS emerged from the integration of management information systems and database management systems (Fig. 1).

Figure 1. Decision Support System.

Various methods can be used for analysis and generating recommendations by the system, including information retrieval, simulation modeling, case-based reasoning, situational analysis, cognitive modeling, machine learning, artificial neural networks, data mining, knowledge discovery in databases, genetic algorithms, and others.

The implementation of DSS increases the efficiency of planning and managing processes in organizations. They facilitate the prompt resolution of emerging problems, allowing for more informed decisions. This leads to time savings and a reduction in the number of employees required, as well as mitigating the human factor in risk assessment.

In the late 1970s, intelligent decision support systems (IDSS) emerged, built on the application of artificial intelligence technologies [21], [22]. Ideally, an IDSS should act as a consultant-specialist to support decision-makers. The system should provide information collection and analysis, identify and diagnose problems, and propose and evaluate possible courses of action.

In the initial stages of implementation, many intelligent decision support systems were based on approaches and technologies developed for expert systems [23], [24]. Currently, the construction of IDSS is based on the use of machine learning methods, artificial neural networks, and data mining.

For the intellectualization of decision support systems, it is proposed to use cloud computing technologies and multi-agent systems (Fig. 2) [25], [26].

Intelligent agents are complex specialized objects that can interact with each other and possess the following properties:

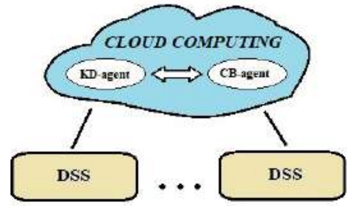

Figure 2. Multi-agent System and Decision Support Systems.

- autonomy or semi-autonomy – functioning without external intervention and ensuring self-control over their actions and internal states;
- social ability – interacting with other agents through message exchange using communication means;
- reactivity – the ability to perceive the state of the external environment;
- pro-activity – agents respond not only to stimuli from the environment but also proactively initiate actions.

The Knowledge Discovery Agent (KD-agent) automatically identifies feature combinations that distinguish classes based on the analysis of training data. The Classifier Builder Agent (CB-agent) constructs a classification algorithm by receiving a set of distinguishing feature combinations from the KD-agent and automatically builds a classifier.

The task of intelligent agents is to implement the procedure of learning from instances. In the classical formulation, the mathematical statement of the problem is as follows: *Let X be a set of object descriptions and Y be a set of valid classification responses. There is an unknown target dependency $y^* : X \to Y$ the values of which $X^m = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ are known only for the objects of the training set. It is required to construct an algorithm $a : X \to Y$ that approximates this target dependency not only on the objects of the finite set but also on the entire set X.*

Traditionally, to solve the problem, a parametric family of algorithms is first selected, in which the initial parameter values are fixed. Then, based on the training data, the final tuning of the algorithm is performed. In fact, learning is conducted within the framework of solving an optimization problem, which is reduced to constructing algorithms *(decision rules)* that provide an extremum for a certain criterion *(for example, the value of the average risk in a special class of decision rules)*.

The main drawback of the described instance-based learning method is the lack of the possibility of its implementation in an automated mode.

Automated instance-based learning is proposed to be conducted based on the properties generality principle. Suppose that $K = \{K_1, \ldots, K_l\}$ is the alphabet of classes, $F = \{f_1, \ldots, f_n\}$ is the a priori dictionary of features, and X is the training set. Let $V = \{v_1, \ldots, v_q\}$

(where $q = 2^n - 1$) be the set of all non-empty subsets formed by all possible combinations of features from F. The goal of learning is to identify such feature combinations $v_i \in V$ that ensure class distinction.

It is proposed to obtain an estimate of the distinguishing properties of combinations $v_i \in V$ in the corresponding feature space by constructing class patterns and comparing their mutual placement. A class pattern should be constructed based on all its objects in the training set and will essentially represent the domain of the class. Feature combinations $v_i \in V$ that ensure class distinction are those for which, in the corresponding feature space, the patterns of different classes do not intersect.

The corresponding task is called *Knowledge Discovery Classification (KDC task)*. The problem statement is as follows: *Let two sets $\{I_0\}$, $\{I(S)\}$ be given - admissible training information $I_0(K_1, \ldots, K_l)$ and descriptions $I(S)$ of admissible objects $S \in M$ respectively. It is required, based on the analysis of the set of information $I_0(K_1, \ldots, K_l)$, to find a set of distinguishing qualities of classes $Q(K_1, \ldots, K_l)$ such that $K_i \cap K_j = \emptyset, \forall i \neq j$ (where $i, j = 1, 2, \ldots, l$) and then, using the set $Q(S)$, compute the values of predicates $P_j(S), j = 1, 2, \ldots, l$.*

III. OSTIS Technology and Intelligent Agents

The process of representing the formal ontology of a subject domain and designing intelligent agents can be implemented based on the comprehensive component design technology of intelligent systems - *OSTIS Technology (Open Semantic Technology for Intelligent Systems)* [27].

The operation of intelligent agents is based on training data. The procedure for its formation begins with the construction of an alphabet of classes and an a priori dictionary of features. The use of OSTIS Technology allows, through *SC-code (sc-models)*, to get the semantic representation of classes, the formalized representation of features of the a priori dictionary, and to construct a training set model.

The result of the KD-agent's work is a set of feature combinations $V^* = \{v_1^*, \ldots, v_t^*\}$, which possess the property of class distinction. Each combination $v_i^* \in V^*$ defines a pattern of the form *"in the space of feature combinations v_i^*, the classes do not intersect"* and is formally described using the unified representation language – *SC-code*.

The input information for the KD-agent is the a priori dictionary of features $F = \{f_1, \ldots, f_n\}$ and the training set X, and the result of its work is a set of feature combinations $V^* = \{v_1^*, \ldots, v_t^*\}$ (where $0 \leq t \leq 2^n - 1$), ensuring class distinction (Fig. 3). The mathematical statement of the problem for the KD-agent is as follows: *Let $F = \{f_1, \ldots, f_n\}$ be the a priori dictionary of features, X be the training set. It is required to find feature spaces in which class patterns do not intersect.*

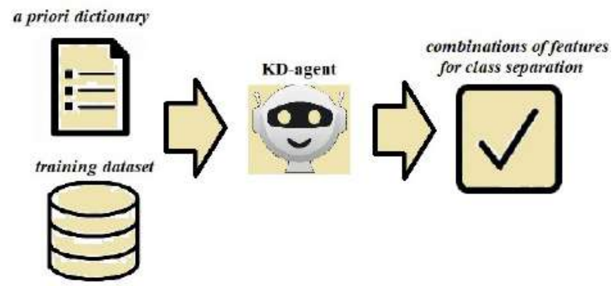

Figure 3. KD-agent Operation Scheme.

Let $V = \{v_1, \ldots, v_q\}$ (where $q = 2^n - 1$) be the set of all non-empty subsets formed by all possible combinations of features from F. The algorithm for searching on the set V for feature combinations that ensure class distinction is as follows:

Step 1. Select a subset $V^+ = \{v_1^+, \ldots, v_+^i\}$ of V, where v_i^+ contains only one feature.

Step 2. For each v_i^+ we construct class patterns (class definition areas) and compare their mutual placement [28].

Step 3. If class patterns do not intersect, then feature v_i^+ is included in the set $V^* = \{v_1^*, \ldots, v_k^*\}$.

Step 4. Exclude from the set $V = \{v_1, \ldots, v_q\}$ the subset $V^+ = \{v_1^+, \ldots, v_n^+\}$ and get $V^\Delta = \{v_1^\Delta, \ldots, v_p^\Delta\}$.

Step 5. Exclude from V^Δ all combinations of v_i^Δ, that contain any combination from $V^* = \{v_1^*, \ldots, v_k^*\}$.

Step 6. Take the next combination v_i^Δ from V^Δ and build a feature subspace based on it.

Step 7. In this feature subspace, we construct class patterns and compare their mutual placement.

Step 8. If the class patterns do not intersect, we include the combination of features v_i^Δ in the set V^*, and exclude from V^Δ all combinations that contain v_i^Δ.

Step 9. Repeat the process until V^Δ is empty.

As a result of the algorithm execution, the set $V^* = \{v_1^*, \ldots, v_t^*\}$ is formed, where $0 \leq t \leq q$.

The input information for the CB-agent is the training set X and the set of feature combinations $V^* = \{v_1^*, \ldots, v_t^*\}$, ensuring class distinction, and the result of its work is a classification algorithm.

The mathematical statement of the problem for the CB-agent is as follows: *Let X be a set of object descriptions and Y be a set of valid classification responses. There is an unknown target dependency $y^* : X \to Y$, the values of which $X^m = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ are known only for the objects of the training set. It is required, based on the set $V^* = \{v_1^*, \ldots, v_t^*\}$ of feature combinations ensuring class distinction, to construct an algorithm $a : X \to Y$ that approximates this target dependency not only on the objects of the finite set but also on the entire set X*(Fig. 4).

Feature combinations $v_i^* \in V^*$ define decision spaces in which class patterns do not intersect. In such spaces,

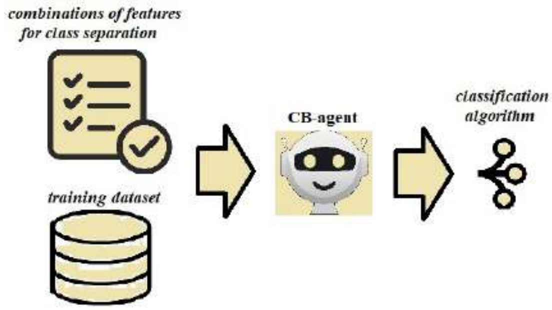

Figure 4. CB-agent Operation Scheme.

classification is conducted in terms of the *KDC task* according to the rule:

- for each feature combination $v_i^* \in V^*$ (where $i = 1, 2, \ldots, t$), based on the training data, cluster structures $P_1^i, \ldots, P_l^i$ – class patterns $K_1, \ldots, K_l$ are constructed;
- the object under study $S \in K_m$ if $S \in P_m^i \forall i = 1, 2, \ldots, t$.

IV. Example of Supervised Learning Based on Model Data

Let the following be given:

- class **A** (four-digit decimal numbers with digits in the *Units* place ranging from 0 to 4 inclusive), class **B** (four-digit decimal numbers with digits in the *Units* place ranging from 5 to 9 inclusive, and in the two higher places *Hundreds* and *Thousands* a pair of digits of the form *even-even* or *odd-odd*), class **C** (five-digit decimal numbers with digits in the *Units* place ranging from 5 to 9 inclusive, and in the two higher places *Hundreds* and *Thousands* a pair of digits of the form *even-odd* or *odd-even*);
- a priori dictionary of features $F = \{Units, Tens, Hundreds, Thousands\}$;
- the content of the training set is presented in Table I.

Table I
Training Set

n/n	Class A	Class B	Class C
1	7854	2479	4319
2	9723	7327	7205
3	5481	8256	3856
4	4270	6825	9248
5	3422	1798	3037
6	1351	3756	2965
7	2650	4439	2389
8	5964	5917	7827
9	9293	9385	5408
10	8762	3148	8746

Table II presents the results of the study of the intersection of patterns of class **A** and the class formed by combining instances of classes **B** and **C**. From Table II, it can be seen that:

Table II
Results for classes **A** and **B+C**

		A	**B+C**			**A**	**B+C**
Units	0	2	0	Hundreds	0	0	1
	1	2	0		1	0	1
	2	2	0		2	2	3
	3	2	0		3	1	4
	4	2	0		4	2	3
	5	0	4		5	0	0
	6	0	4		6	1	0
	7	0	4		7	2	3
	8	0	4		8	1	3
	9	0	4		9	1	2
Tens	0	0	2	Thousands	0	0	0
	1	0	2		1	1	1
	2	2	3		2	1	3
	3	0	2		3	1	4
	4	0	3		4	1	2
	5	3	3		5	2	2
	6	2	1		6	0	1
	7	1	1		7	1	3
	8	1	2		8	1	2
	9	1	1		9	2	2

Table III
Results for classes **B** and **A+C**

		B	**A+C**			**B**	**A+C**
Units	0	0	2	Hundreds	0	0	1
	1	0	2		1	1	0
	2	0	2		2	1	4
	3	0	2		3	2	3
	4	0	2		4	2	3
	5	2	2		5	0	0
	6	2	2		6	0	1
	7	2	2		7	2	3
	8	2	2		8	1	3
	9	2	2		9	1	2
Tens	0	0	2	Thousands	0	0	0
	1	1	1		1	1	1
	2	2	3		2	1	3
	3	1	1		3	2	3
	4	1	2		4	1	2
	5	2	4		5	1	3
	6	0	3		6	1	0
	7	1	1		7	1	3
	8	1	2		8	1	2
	9	1	1		9	1	3

- values 0, 1, 2, 3, 4 of the *Units* feature are present only in representatives of class **A**, and values 5, 6, 7, 8, 9 only in representatives of the combined class **B+C**, which means that the *Units* feature ensures the distinction between classes **A** and the combined **B+C**;
- features *Tens*, *Hundreds* and *Thousands* do not possess the property of distinguishing classes **A** and the combined **B+C**.

Table III presents the results of the study of the intersection of patterns of class **B** and the class formed by combining representatives of classes **A** and **C**. From Table III, it can be seen that none of the features *Units, Tens, Hundreds* or *Thousands* ensures the distinction between classes **B** and **A+C**.

Table IV presents the results of the study of the intersection of patterns of class **C** and the class formed by combining instances of classes **A** and **B**.

Table IV
Results for classes **C** and **A+B**

		C	A+B			C	A+B
Units	0	0	2	Hudreds	0	1	0
	1	0	2		1	0	1
	2	0	2		2	2	3
	3	0	2		3	2	3
	4	0	2		4	1	4
	5	2	2		5	0	0
	6	2	2		6	0	1
	7	2	2		7	1	4
	8	2	2		8	2	2
	9	2	2		9	1	2
Tens	0	2	0	Thousands	0	0	0
	1	1	1		1	0	2
	2	1	4		2	2	2
	3	1	1		3	2	3
	4	2	1		4	1	2
	5	1	5		5	1	3
	6	1	2		6	0	1
	7	0	2		7	2	2
	8	1	2		8	1	2
	9	0	2		9	1	3

From Table IV, it can be seen that none of the features *Units, Tens, Hundreds* or *Thousands* ensures the distinction between classes **C** and **A+B**.

The results of the study of the properties of the features *Units, Tens, Hundreds* and *Thousands* demonstrated that the properties of the *Units* feature are unique for class **A**, and the following intermediate classification rule can be formulated:

IF ($0 \leq Units \leq 4$) ***THEN Class A***
ELSE (Class B or Class C)

. Next, to finalize the classifier, it is necessary to find feature combinations that ensure the distinction between classes **B** and **C**.

Let's proceed to analyze the properties of feature combinations *(Units, Tens)*, *(Units, Hundreds)*, *(Units, Thousands)*, *(Tens, Hundreds)*, *(Tens, Thousands)*, *(Hundreds, Thousands)*.

Table V presents the results of the study of the mutual placement of patterns of classes **B** and **C** in the space formed by the combination of features *(Units, Tens)*.

From Table V, it can be seen that there are intersections of patterns of classes **B** and **C** (pairs 27, 48 and 56), and the combination of features *(Units, Tens)* does not ensure the distinction between classes **B** and **C**.

Table VI presents the results of the study of the mutual placement of patterns of classes **B** and **C** in the space formed by the combination of features *(Hundreds, Thousands)*. From Table 6, it can be seen that the patterns of classes **B** and **C** do not intersect. For representatives of class **B**, the combinations of values of features *(Hundreds, Thousands)* are either *(even, even)* or *(odd, odd)*, while for representatives of class **C**, the combinations of values of features *(Hundreds, Thousands)* are either *(odd, even)* or *(even, odd)*.

Table V
Results of placement of patterns **B** and **C** in the space *(Units, Tens)*

Units Tens	0	1	2	3	4	5	6	7	8	9
0						C		B	C	
1										C
2						B		B,C		
3								C		B
4							C		B,C	
5							B,B,C			
6						C				
7										B
8						B				C
9									B	

Table VI
Results of placement of patterns **B** and **C** in the space *(Hundreds, Thousands)*

Hundreds Thousands	0	1	2	3	4	5	6	7	8	9
0										
1								B		
2				C	B					C
3	C	B						B	C	
4				C	B					
5					C					B
6									B	
7			C	B					C	
8			B					C		
9			C	B						

The results of the study of the properties of feature combinations *(Units, Tens)* and *(Hundreds, Thousands)* demonstrated that the classification rule for representatives of the two classes **C** and **B** is as follows:

IF ((Hundreds, Thousands) = (even, even) or
(Hundreds, Thousands) = (odd, odd))
THEN Class B ELSE Class C

Combining the two local rules identified during the study, for the three classes **A**, **B**, **C**, we obtain the following general rule for constructing the classifier:

IF ($0 \leq Units \leq 4$) ***THEN Class A***
ELSE IF ((Hundreds, Thousands) = (even, even) or
(Hundreds, Thousands) = (odd, odd)) THEN Class B
ELSE Class C

V. Conclusion

The article proposes a method for the intellectualization of decision support systems based on the joint use of OSTIS Technology, autonomous agents, and cloud computing. An original method for implementing intelligent agents based on the properties generality principle is presented. The input is a training set, and by interacting with each other, the agents automatically implement supervised learning.

An algorithm for learning from instances is described to identify feature combinations that possess the property of class distinction. In an automated mode, the training data is analyzed, and a classifier is constructed. A model example demonstrates the results of applying an intelligent agent for the automatic construction of a classification algorithm based on the proposed method and algorithm for learning from instances.

References

[1] V. Glushkov Osnovy bezbumazhnoj informatiki [Fundamentals of paperless computer science], Moscow, Nauka, 1987, 552 p.
[2] Computer science. Available at: https://en.wikipedia.org/wiki/Computer_science (accessed 2025, March)
[3] G. Luger Iskusstvennyj intellekt: strategii i metody resheniya slozhnykh problem [Artificial Intelligence: Structures and Strategies for Complex Problem Solving], Moscow, Williams Publishing House, 2003, 864 p.
[4] S. Russel, P. Norving Iskusstvennyj intellekt: sovremennyj podkhod [Artificial intelligence: a modern approach], Moscow, Williams Publishing House, 2006, 1408 p.
[5] Yu. Nechaev, A. Degtyarev Intellektualnyje sistemy: kontseptsii i prilozheniya [Intelligent systems: concepts and applications], Saint-Petersburg, Publishing SPb. University, 2011, 269 p.
[6] Artificial intelligence. Available at: https://en.wikipedia.org/wiki/Artificial_intelligence (accessed 2025, March)
[7] P. Flach Mashinnoe obuchenie. Nauka i iskusstvo postroeniya algoritmov, kotorye izvlekayut znaniya iz dannykh [Machine learning. Science and art of constructing algorithms that extract knowledge from data], Moscow, DMK Press, 2015, 400 p.
[8] Machine learning. Available at: https://en.wikipedia.org/wiki/Machine_learning (accessed 2025, March)
[9] A. Pandya, R. Macy Pattern Recognition with Neural Networks in C++, Boca Raton, CRC Press, 1995, 426 p.
[10] Neural network (machine learning). Available at: https://en.wikipedia.org/wiki/Neural_network_(machine_learning) (accessed 2025, March)
[11] A. Barsegyan, M. Kupriyanov, V. Stepanenko Tekhnologii analiza dannykh: Data Mining, Visual Mining, Text Mining, OLAP [Data Analysis Technologies: Data Mining, Visual Mining, Text Mining, OLAP], Saint-Petersburg, BKhV-Peterburg, 2007, 384 p.
[12] Data mining. Available at: https://en.wikipedia.org/wiki/Data_mining (accessed 2025, March)
[13] V. Ruchkin, V. Fulin Universalnuj iskusstvennuj intellect i ekspertnue sistemu [Universal artificial intelligence and expert systems], Saint-Petersburg, BKhV-Peterburg, 2009, 240 p.
[14] V. Krasnoproshin, V. Rodchenko, A. Karkanitsa The Properties Generality Principle and Knowledge Discovery Classification, Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system [Open semantic technologies for intelligent systems], 2024, pp.15–120.
[15] S. Raschka Paython i mashinnoe obuchenie [Python and machine learning], Moscow, DMK Press, 2017, 418 p.
[16] V. Krasnoproshin, V. Rodchenko Obuchenie po pretsedentam na osnove analiza svojstv priznakov [Learning by precedents based on the analysis of the features], Doklady BGUIR [BSUIR Reports], 2017, no 6, pp.35-41.
[17] A. Osrtoukh Intellektualnye sistemy: monografiya [Intelligent systems: monograph], Krasnoyarsk, Scientific information center, 2020, 316 p.
[18] V. Rodchenko Pattern Recognition: Supervised Learning on the Bases of Cluster Structures, Pattern Recognition and Information Processing, 2017, pp. 106-113.
[19] L. Rutkovskij Metody i tekhnologii iskusstvennogo intellekta [Methods and technologies of artificial intelligence], Moscow, Publisher Telekom, 2010, 520 p.
[20] V. V. Krasnoproshin, V.G. Rodchenko Klassifikatsiya na osnove prostranstv reshenii [Classification based on decision spaces], Doklady BGUIR [BSUIR Reports], 2019, no 6, pp.20-25.
[21] J. Giarratano, G. Riley Ekspertnye sistemy: printsipy razrabotki i programmirovanie [Expert systems: principles and programming], Moscow, Williams Publishing House, 2007, 1152 p.
[22] T. Gavrilova, D. Kudryavtsev, D. Muromtsev Ingeneriya znanij. Modeli i metody [Knowledge engineering. Models and methods], Saint-Petersburg, Publishing House Lan, 2016, 324 p.
[23] C. Naylor Kak postroit' svoju ekspertnuju sistemy [Build your own expert system], Moscow, Energoatomizdat, 1991, 286 p.
[24] P. Jackson Vvedenie v ekspertnyje sistemu [Introduction to expert systems], Moscow, Williams Publishing House, 2001, 624 p.
[25] V. Krasnoproshin, V. Rodchenko, A. Karkanitsa Specialezed KD-agent for knowledge ecosystems, Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system [Open semantic technologies for intelligent systems], 2021, pp.59–62.
[26] V. Krasnoproshin, V. Rodchenko, A. Karkanitsa Automatic construction of classifiers by knowledge ecosystem agents, Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system [Open semantic technologies for intelligent systems], 2022, pp.195–198.
[27] V. Golenkov, N. Guliakina, I. Davydenko, D. Shunkevich, A. Eremeev Ontologicheskoe proektirovanie gibridnykh semanticheskikh system na osnove smyslovogo predstavleniya znanij [Ontological design of hybrid semantically compatible intelligent systems based on sense representation of knowledge], Ontology of designing, 2019, no 1, pp.132-151.
[28] V. Krasnoproshin, V. Rodchenko Klasternye struktury i ikh primenenie v intellektualnom analize dannykh [Cluster structures and their applications in data mining], Informatika [Computer Science], 2016, no 2, pp.71-77.

ИНТЕЛЛЕКТУАЛИЗАЦИЯ СИСТЕМ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ НА ОСНОВЕ ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ

Краснопрошин В. В., Родченко В. Г., Карканица А. В.

Системы поддержки принятия решений представляют собой системы обработки информации, которые возникли в результате слияния автоматизированных систем управления и систем управления базами данных. Их назначение – помогать пользователям принимать решения по сложным не- или слабоструктурированным проблемам. Для анализа и выработки системой предложений могут быть использованы различные методы – информационный поиск, имитационное моделирование, рассуждение на основе прецедентов, ситуационный анализ, когнитивное моделирование, машинное обучение, искусственные нейронные сети, интеллектуальный анализ данных, поиск знаний в базах данных, генетические алгоритмы и др. В статье предлагается метод интеллектуализации систем поддержки принятия решений на основе использования технологии OSTIS, интеллектуальных агентов и облачных технологий. Описывается реализация автономных агентов для обнаружения в данных скрытых интерпретируемых закономерностей и последующего построения классификатора. Эффективность метода подтверждается результатами численного эксперимента на основе модельных данных.

Received 24.03.2025