UDC 004.93; 004.8; 004.942

V. Tigariev[1], PhD, Assoc. Prof.,
O. Lopakov[1],
V. Kosmachevskiy[1],
A. Liushenko[2]
[1] Odessa Polytechnic National University, Shevchenko Ave. 1, Odesa, Ukraine, 65044, e-mail: kedrodessa9@gmail.com
[2] Odesa I. I. Mechnikov National University, Dvoryanska St. 2, Odesa, 65082, Ukraine

DEVELOPMENT OF THE BASIC MODULE OF THE FUZZY ARTMAP ALGORITHM IN OPERATING SYSTEMS FOR DECISION MAKING AND INTELLIGENT DATA ANALYSIS

В. Тігарєв, О. Лопаков, В. Космачевський, А. Люшенко. **Розробка базового модуля алгоритму нечіткої логіки fuzzy artmap в операційних системах прийняття рішень та інтелектуального аналізу даних.** У системах підтримки прийняття рішень при управлінні організаціями на практиці часто застосовуються прості і зрозумілі моделі, як, наприклад, правила прийняття рішень на основі відомих методів нечіткої логіки, лінійна або логістична регресія, метод дерев класифікації і регресії. Цінність і практична значимість подібних алгоритмів полягає у важливій здатності цих алгоритмів розуміти і роз'яснювати їх внутрішню логіку прийняття рішення, але недоліком є їх невисока точність. Більш точні нейромережеві алгоритми, як правило, не мають властивості інтерпретованості. Однак пропонований в даному дослідженні алгоритм поєднує в собі досить високу точність при аналізі моніторингової інформації і, разом з тим, він добре пояснює отримані рішення. Для розробки методів і алгоритмів інтелектуальної підтримки прийняття управлінських рішень при обробці моніторингової інформації найкраще підходять нейронні мережі сімейства ART, оскільки вони відрізняються стабільною і швидкою атрибуцією даних, і разом з тим пластичністю для запам'ятовування нової інформації. На основі використання мереж сімейства ART запропоновано загальний підхід до вирішення завдань кластеризації моніторингових даних. Оскільки загальновідомим недоліком мереж сімейства ART є залежність від початкової ініціалізації гіперпараметрів, досліджено вид і характер даної залежності в завданнях кластеризації моніторингових даних. Запропоновано генетичний алгоритм для автоматичного налаштування гіперпараметрів мережі Fuzzy ARTMAP з метою подолання зазначеного недоліку. Такий алгоритм дозволяє вдосконалювати методи отримання та обробки інформації для завдань управління організаційними системами. Оскільки моніторингова інформація може породжувати потоки даних великого обсягу, запропоновано алгоритм використання ансамблю мереж Fuzzy ARTMAP для паралельної обробки та структурування потокових даних. Технологія паралельних і високопродуктивних обчислень розвивається, а обчислювальні ресурси стають все більш доступними, з'являються нові можливості для розпаралелювання моделей нейронних мереж, з точки зору кращої обробки обчислень і підвищення інтенсивності обробки даних, а, отже, і підвищення швидкості прийняття управлінських рішень на основі даних, які оперативно надходять , що особливо важливо в завданнях управління в організаційних системах. У статті розроблено та досліджено алгоритм навчання мережі Fuzzy ARTMAP для вирішення задачі класифікації в умовах пересічних класів. Така задача часто виникає при аналізі моніторингової інформації в системах підтримки прийняття управлінських рішень, оскільки при зборі оперативних даних часто зустрічаються шуми і помилки, що розмиває межі між класами, на які розбиваються значення вхідних моніторингових показників. Для цього алгоритму запропоновано модифіковану функцію вибору, що забезпечує подібну класифікацію, математично обгрунтовано її властивості.

Ключові слова: машинне навчання, алгоритми класифікації та кластеризації, нейро-нечіткі мережі (ANFIS), операційні системи (ОС), нормалізація вхідних векторів, препроцесинг, постпроцесинг, швидкість навчання мережі, метод комплементації, критерій подібності, фазіфікація даних

V. Tigarev, A. Lopakov, V. Kosmachevskiy, A. Liushenko. **Development of the basic module of the fuzzy artmap algorithm in operating systems for decision making and intelligent data analysis.** In decision support systems for organizational management, simple and understandable models are often used in practice, such as decision-making rules based on well-known fuzzy logic methods, linear or logistic regression, and classification and regression tree methods. The value and practical significance of such algorithms lie in their important ability to understand and explain their internal decision-making logic, but their disadvantage is their low accuracy. More accurate neural network algorithms, as a rule, do not have the property of interpretability. However, the algorithm proposed in this study combines a sufficiently high accuracy in the analysis of monitoring information and, at the same time, it explains the resulting decisions well. ART neural networks are best suited for developing methods and algorithms for intelligent support of management decision-making when processing monitoring information, as they are characterized by stable and fast data attribution, and at the same time are flexible for storing new information. Based on the use of ART family networks, a general approach to solving monitoring data clustering problems has been proposed. Since a well-known disadvantage of ART family networks is their dependence on the initial initialization of hyperparameters, the type and nature of this dependence in monitoring data clustering problems have been investigated. A genetic algorithm has been proposed for the automatic tuning of the hyperparameters of the Fuzzy ARTMAP network in order to overcome this drawback. Such an algorithm allows for the improvement of methods for obtaining and processing information for organizational system management tasks. Since monitoring information can generate large data flows, an algorithm for using an ensemble of Fuzzy ARTMAP networks for parallel processing and structuring of stream data is proposed. Parallel and high-performance computing technology is developing, and computing resources are becoming increasingly accessible, new opportunities are emerging for parallelizing neural network models in terms of better processing of calculations and increasing the intensity of data processing, and, consequently, increasing the speed of management decisions based on operational data, which is especially important in management tasks in organizational systems. This article develops and investigates an algorithm for training a Fuzzy ARTMAP network to solve classification problems in conditions of overlapping classes. Such a task often

DOI: 10.15276/opu.2.72.2025.15

arises when analyzing monitoring information in management decision support systems, since noise and errors often occur when collecting operational data, which blurs the boundaries between the classes into which the values of the input monitoring indicators are divided. A modified selection function that provides such classification is proposed for this algorithm, and its properties are mathematically justified.

Keywords: machine learning, classification and clustering algorithms, artificial neural fuzzy inverse systems (ANFIS), operating systems (OS), input vector normalization, preprocessing, postprocessing, network learning speed, complementation method, similarity criterion, data fuzzification

Introduction

Improving management processes in operating systems at the present stage requires the introduction of new information technologies, including methods of decision support based on operational monitoring data. In today's world, monitoring has become an integral part of various fields of activity. For example, it is actively used in ecology to control environmental pollution; in medicine to help monitor the health of patients; and in education to assess the effectiveness of educational institutions. The use of monitoring data to support management decision-making in operating systems has a number of advantages. First, monitoring provides continuous information about the status of objects and processes, which makes it possible to respond quickly to emerging problems when making management decisions. Second, monitoring contributes to more efficient use of resources, as processes can be optimized and losses reduced based on up-to-date data. However, despite all its advantages, monitoring also has its limitations. Existing methods of processing monitoring data are generally insufficiently effective due to the inability to quickly take into account large amounts of incoming information (including noisy data and data containing missing values collected from diverse sources). Incorrect processing and analysis of monitoring data often leads to wrong decisions. In general, the use of monitoring data in management requires not only its correct analysis, but also an understanding of how this data should influence changes in the organization. Machine learning technologies that have emerged in recent years, based on neural network and neuro-fuzzy approaches, allow management systems to process real-time data, including data containing missing, erroneous, or inaccurate values, and automatically generate examples of management decisions. A study of existing neural network and neuro-fuzzy architectures has revealed the feasibility of using adaptive resonance networks (ART) for processing monitoring data in decision support tasks. ART networks are stable, which means they can retain accumulated knowledge throughout the entire operating time of the system. In addition, they provide flexibility through the use of an incremental learning mechanism. Incremental learning allows you to take into account current information about the state of objects and respond quickly to changes in the situation. Cascade ARTMAP neuro-fuzzy models are good at processing noisy data and allow the development of a system for the automated construction of decision rules to support management decision-making based on monitoring data.

Analysis of recent publications and problem statement

Problems related to the development of management methods and mechanisms based on monitoring data were considered in the works of V.N. Burkov, D.A. Novikov, D.V. Gaskarov, A.V. Shchepkina, Ya. E. Lvovich, V.A. Irikov, V.D. Kondratiev, G.A. Ugolnitsky, and others. Modern neural network and neuro-fuzzy technologies used in the development of information support for control systems are discussed in the works of K. Broyden, D. Goldfarb, E. Mamdani, G.S. Pospelov, S. Haikin, D. Shanno, G. Carpenter, S. Grossberg, Y. Lecune, P. Flach, and Y. Goodfellow. However, issues related to the implementation of control methods based on real-time monitoring data using machine learning algorithms in the practice of decision support in control systems have not yet been sufficiently addressed in the literature. In this regard, the relevance of the topic of this article is dictated by the need for further development of intelligent decision support tools in organizational systems in the context of operational monitoring based on adaptive resonance neural networks.

The problem with using the neural ART system lies in the complexity of practical network use due to the lack of a generally accepted authorial formalization. The authors' publications reveal the concepts of functioning but do not pay attention to the training of structured algorithms [1 – 4]. Because of this, researchers using the system describe the structure of ART and training algorithms in different ways [4 – 6]. Therefore, it can be stated that there is no single standard for describing the ART family in scientific literature.

The purpose of the study

Development of means for algorithmizing the processes of management decision-making in OS based on incremental neural network methods of monitoring data analysis.

To achieve this study, the following tasks must be completed:

Conduct an analysis of management systems based on monitoring data in organizational systems, identify problems, and, on this basis, formulate relevant directions for the development of decision support systems.

Develop a structural and functional model for supporting management decision-making based on monitoring indicators in order to improve the information support for management processes in organizations.

Develop a modified algorithm for clustering monitoring data based on the Fuzzy ART neural network model to improve the cluster approach to management with the ability to select different control actions for different clusters of monitoring indicator values in the context of their operational analysis.

Subject of – methods for intellectualizing decision-making processes in OS based on monitoring data received in real time.

To solve the problems posed in the dissertation, methods of systems analysis, decision making, machine learning, big data processing, fuzzy logic, artificial neural network theory, and modern programming methods and tools were used.

Statement of the main material

Development of a clustering algorithm for operational analysis of monitoring data based on the FuzzyART neural-fuzzy network

There are several approaches to developing management information technologies based on monitoring data. Let's consider the main ones.

Real-time data analysis. This method involves continuous monitoring of various parameters and sensors to collect data in real time. The data obtained is analyzed to identify trends, patterns, and anomalies, which allows for quick management decisions.

Use of machine learning. This approach is based on the application of machine learning algorithms to analyze and process monitoring data.

Machine learning models can be trained to recognize certain patterns or predict future parameter values based on historical data.

Use of big data analytics. This approach involves collecting and analyzing large amounts of information using specialized tools and algorithms. Big data allows you to identify complex relationships and trends that can be used in management decision-making.

Using the Internet of Things (IoT). With the development of IoT technologies, monitoring capabilities have expanded significantly. IoT devices can collect and transmit data from various objects and sensors, enabling more accurate monitoring and control.

Process automation and optimization. This approach involves developing information technologies that enable the automation and optimization of various processes based on monitoring data. This may include automatic real-time parameter adjustment or the proposal of optimal management strategies based on the data obtained. The final choice of method for developing information technologies for management based on monitoring data depends on the specific task and requirements of the organization. However, machine learning methods are currently used as the main technology for the operational analysis of monitoring information. Let's consider the main reasons for this.

First, machine learning algorithms can quickly analyze and extract information from streaming data, which greatly simplifies working with large amounts of data.

Second, machine learning methods are capable of searching for and discovering hidden patterns and relationships in monitoring data that may be invisible even to an expert. This allows you to identify similar patterns in sets of monitoring indicators and predict future trends based on available data, which is valuable information for decision-making.

Third, machine learning methods allow you to create models that take into account complex nonlinear interactions and dependencies in data. Such models can be more accurate and of higher quality than linear approaches.

Fourth, machine learning methods can learn from new data and update model parameters, allowing them to adapt to changes and improve their performance. This is especially important for processing monitoring data, which can change significantly over time.

Fifth, machine learning methods can be used to automatically generate management actions based on previously accumulated knowledge.

 ISSN 2076-2429 (print)
ISSN 2223-3814 (online)

Figure 1 shows a diagram of the decision support process based on machine learning methods using monitoring data.

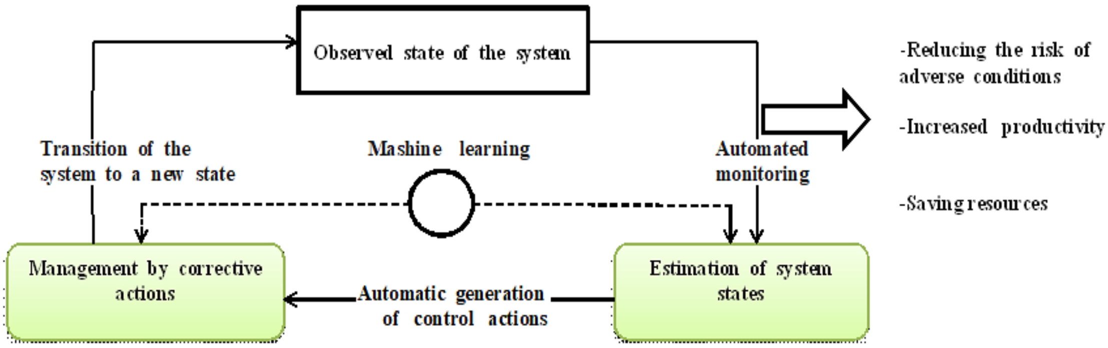

Fig. 1. Decision-making based on monitoring data using machine learning methods

Typically, methods for structuring monitoring data include clustering and classification algorithms. Clustering refers to a fundamental method of researching and processing indicators, widely used for:

− recognizing similar situations [7];

− identifying key features (characteristic features) of problematic situations [8 – 10];

− intelligent analysis, i.e., extracting knowledge from data [10].

Clustering identifies common structures present in a data set based on some measure of similarity.

Classification algorithms are used to systematize the information being processed. Classification can be binary or multi-class, produce overlapping or non-overlapping classes, and output a class label or a probability vector for each of the available classes. In any case, the classification algorithm structures the source data according to some feature.

The availability of modern capabilities for collecting and storing indicators helps to generate huge amounts of information for further use, searching for connections, interactions, and patterns. Information arrays demonstrate the need to include subprograms in comprehensive monitoring programs that conduct systematic information analysis using the characteristic features of indicators—multidimensionality and multirelationality. This contributes to difficulties in assessing and forecasting problematic situations. The decision-maker uses assessment methods, but they are not based on integration, so in practice, they are often considered ineffective and require more detailed elaboration on all strategies for responding to existing problems and damage caused. Particular attention should be paid to certain individual factors that influence their complex interaction. For example, aggregation is a traditional method of collecting and analyzing information based on the average monitoring value of indicators. Studies have shown that approaches to existing problems somewhat level out the differences between problematic situations that may be signaled by monitoring indicators [11]. To improve the accuracy and timeliness of management decisions, new intelligent technologies should be used in analysis. Such an intelligent approach to analysis will allow work to be carried out even in conditions of insufficient information. For example, studies [12 – 14] and [15] suggest using a scientific and methodological apparatus of a qualimetric approach based on fuzzy technologies when managing based on monitoring data. This will help to construct a non-additive integral assessment, which will serve as a weighted average "quasi-geometric" value. In order for system analysis to yield tangible results in problematic situations, it is necessary to build an integral assessment based on monitoring data by applying different approaches that establish structural links between the system's monitoring indicators, as this will allow the most logical and consistent approach to be chosen when making important decisions.

In general, methods based on fuzzy logic are good at processing qualitative and imprecise indicator values, while also being highly interpretable. However, these methods require significant effort to configure their hyperparameters, involving subject matter experts. If such tuning were required only once, it would not be a significant limitation, but even a small change in the set of parameters being analyzed requires significant refinement of the system. Another group of methods used to process monitoring indicators are evolutionary modeling methods. The key to using genetic algorithms (GA) in monitoring data processing is to frame the problem as an optimization task. Many monitoring data processing tasks are optimization tasks, for example, numerous applications of inverse models (i.e., tasks of finding such values of monitoring indicators that would lead to the desired effect in management). [16] describes a specific optimization problem in which GA proved useful – the use of pollutant concentration data in combination with a transport and dispersion model to reverse-calculate control actions based on meteorological information. Genetic algorithms (GA) can be used to optimize the parameters of monitoring data processing algorithms. However, this is an auxiliary tool, not a primary one. One of the main areas of computer monitoring technologies based on artificial intelligence is the use of neural networks, which greatly facilitate the analysis and prediction of problem situations. Today, artificial neural networks are one of the most advanced and modern methods of processing monitoring information and extracting knowledge from monitoring data. This is not surprising, since neural network algorithms have high potential for solving monitoring data processing tasks and are good at processing noisy and incomplete data. The only problem is that, as is well known, neural networks operate in a "black box" mode, i.e., they are not understandable and interpretable methods, which limits their applicability in many subject areas. In particular, in OS management tasks, situations sometimes arise that require urgent intervention, when the cost of an error can be very high. In such cases, experts cannot rely on the result predicted by the "black box." In such subject areas, in order to apply a decision support algorithm, it must not only be accurate, but also understandable to experts (decision makers) so that they can make adjustments to the resulting decision. Due to the inherent requirements for interpretability, experts often have to use a narrow range of models in practice: classification trees, fuzzy knowledge bases, or linear regression, while neglecting the low accuracy of the results obtained. Prior knowledge of the problem domain can help a neural network learn to solve the problem of insufficient interpretability. In particular, previously existing classification rules can be used to initialize the neural network architecture before training. The use of initial rules not only increases the efficiency of network training, but also allows you to obtain knowledge that cannot be obtained as a result of training or that cannot be easily learned by a neural network, thereby improving the predictive performance of the system. In addition, incomplete or partially correct knowledge formulated in the form of classification rules can be improved or expanded using neural network training algorithms. Thus, the introduction and refinement of rules into neural networks automates the expansion and restoration of expert knowledge. On the other hand, when there are high requirements for algorithm accuracy, experts are forced to resort to the use of deep learning algorithms [16 – 18]. Deep learning uses a backpropagation algorithm to learn to predict certain output vectors in response to processed input vectors [18]. However, deep learning can lead to information forgetting: at any stage of training, part of the neural network's memory may be destroyed [19]. The reason for this phenomenon is that all network input data is processed by the backpropagation algorithm through a common set of adjustable weights, while there is no mechanism within the algorithm for selectively buffering previous training that is predictively useful. Such forgetting can occur in any training algorithm in which weight updates are based on the use of the error gradient in response to the current input data packet.

Thus, one of the main problems with using backpropagation networks to improve rule-based knowledge is the preservation of accumulated knowledge. In the process of adjusting weights using the backpropagation algorithm, the initial rules quickly lose their original values. In fact, such training can result in large shifts in the values of the weights of hidden layers [20]. Another serious limitation of the backpropagation method is that the initial rule base must be almost complete, otherwise the initial network architecture created may not be accurate enough to process the entire data set. Since the standard backpropagation algorithm cannot dynamically create additional neurons or connections between them during training, a network initialized with a small set of rules may even have less chance of ultimately solving the classification problem. This problem was noted and partially solved in [21], which uses virtual rules to create potential connections for training. However, in general, it is difficult to determine the desired decision rules in advance. In [22], a learning algorithm is used that allows additional neurons to be created during the learning process. However, since this model consists only of rules that directly link input attributes to output predictions, the network is not versatile enough to work on a rule-based basis using intermediate attributes and rule chains. From all of the above, we can conclude that an effective adaptive decision support algorithm in an OS based on monitoring data must be capable of incremental learning, i.e., using all newly arriving data to retrain the existing model. As already noted, the specifics of management tasks in organizations based on monitoring indica-

ISSN 2076-2429 (print)
ISSN 2223-3814 (online)

tors include decision-making using data that contains noise and missing values. Input data may be partially incorrect due to faulty sensors. However, during training, such data should not "spoil" correctly configured weights. Thus, the peculiarities of management tasks in OS based on monitoring data dictate the need to use incrementally learning neural network algorithms. At the same time, the resulting data structuring results must be interpretable, that is, they must induce IF...THEN rules that explain which combinations of input features predict these particular results.

The first stage of processing incoming monitoring data is often clustering, which is used to perform preliminary systematization. The monitoring environment uses machine learning methods to perform cluster structuring of data across all aggregate indicators. The results determine the most appropriate alternative approach to structuring and enable the transition to the mathematical modeling stage. At the same time, such structuring methods of information processing allow focusing on specific goals.

By forming a cluster structure of the initial indicators, it is possible to understand the information they contain. The resulting clustering allows for the improvement of current approaches to management decision-making and the development of strategies for responding to deviations in the values of monitoring indicators. A special management technique can be applied to each cluster.

− During the clustering process, new and atypical data can be identified by detecting monitoring indicator values that do not belong to any of the clusters. These data require individual analysis to identify their characteristics.

− Clustering reduces the amount of information to be analyzed, allowing you to analyze not every set of monitoring indicator values, but only those combinations that are typical for individual clusters.

It is worth noting a particular pattern used to improve the assessment methodology. The widespread use of decision support systems in management based on monitoring data allows for the collection and processing of information, including the use of special sensors located at monitoring sites. Information from sensors is needed to conduct a full assessment and analysis of the functioning of facilities and the presence of man-made and negative natural phenomena. The quality of decisions made on the organization of further actions depends on a number of factors:

− timely receipt of information about all processes at the facilities;
− clear differentiation of processing results by clusters;
− objective processing of analytical data.

The indicators obtained from sensors are considered to be variants of more comprehensive indicators that require the application of intelligent control algorithms and processing of incoming analytical flows.

Thus, it is possible to formulate the characteristic features of monitoring data and the requirements for the algorithmic support for their processing.

As a rule, monitoring data is very large in volume, which requires its automatic processing (including in real time) and automated formation of control decisions.

Data for processing can be received in batch mode (for periodic monitoring) or in streaming mode (for continuous monitoring). Consequently, decision support methods are needed that can adapt to changing data flows and scale to their volume. Monitoring indicators are characterized by nonlinear interactions that affect the state of controlled objects, as well as inherent emergence. Consequently, decision support systems (DSS) require methods that take into account nonlinear relationships and at the same time form a structure that does not grow uncontrollably with the processing of large amounts of data.

Monitoring data usually contains noise, gaps, and inaccurate values (especially data transmitted from sensors). Therefore, methods are needed that can work with incomplete, fuzzy, and noisy information. Monitoring data used to support management decisions requires interpretive analysis. Therefore, methods are needed that can explain the results obtained. Practically all of the above requirements for algorithms for structuring monitoring data are met by the ART family of neural networks (based on adaptive resonance theory). The first networks of this family were proposed in the works of Carpenter and Grossberg [19-20]. However, these networks did not receive widespread application at that time, as they were inferior in accuracy to traditional networks when dealing with small amounts of data. In addition, ART networks have a more complex architecture than multilayer perceptron networks or Kohonen and Hopfield networks (proposed around the same time), and they are more difficult to configure (as they have a large number of hyperparameters). However, when solving big data processing tasks, especially those coming into the network as a continuous stream (such as monitoring information from sensors), these networks have undeniable advantages over many traditional networks. These advantages are based on the use of a built-in incremental learning mechanism focused on solving the stability-plasticity dilemma: ANNs are retrained "on the fly," and new information does not erase accumulated knowledge. In addition, the ART network training algorithm is easily parallelized, making it suitable for big data processing. Most importantly, some subtypes of the ART family of networks are interpretable models, i.e., they allow the presentation of learning results in the format of "if-then" rules, similar to what can be done in the decision tree method. ART networks can solve both classification and data clustering tasks, and even when solving classification tasks, they divide the data into clusters (clusters) in an adaptive self-learning mode, with each resulting class corresponding to several categories.

The task of updating ART family networks in algorithms for structuring streaming monitoring data is of paramount importance, since there are currently no other architectures that are as suitable for solving this task. Table 1 provides a comparative analysis of the advantages and disadvantages of various methods of processing monitoring information, including ART family networks.

Table 1

Advantages and disadvantages of existing methods for processing monitoring information

Methods	Advantages	Disadvantages
Empirical-statistical methods	Well researched, long used, many of them interpretable	They do not have the ability to process incoming information (including noisy and incomplete information) in a timely manner, relying only on linear dependencies. They do not have the ability to process incoming information (including noisy and incomplete information) in a timely manner, relying only on linear dependencies.
Methods based on neural networks not belonging to the ART family	There are powerful learning algorithms that automatically generate features based on input indicators and allow nonlinear dependencies to be constructed, easily adapting to changes in the structure of the input data.	They do not have the property of incremental learning, are not interpretable, do not cope well with noisy and qualitative data, and require hyperparameter tuning.
Evolutionary, genetic algorithms	Allow automatic configuration of method parameters	They do not have the property of interpretability and do not cope well with noisy and qualitative data.
Methods based on fuzzy logic	They handle data containing gaps and erroneous or inaccurate values well, as well as qualitative data, which is important when processing monitoring data that may be partially incorrect due to faulty sensors. They have the property of interpretability.	Difficult to configure and train, requiring significant involvement of experts in developing methods and complete renewal of the entire system whenever the set of input indicators changes.
Neuro-fuzzy networks ANFIS (Adaptive Neuro-Fuzzy Inference System) and TSK (Takagi-Sugeno-Kang)	They have the ability to adapt to changes in data. They provide the ability to interpret results. They have learning algorithms for adjusting model parameters and combine the advantages of neural networks and fuzzy logic.	Training requires a large amount of data and computing resources, as well as the involvement of experts to select fuzzy membership functions. Prone to overfitting. Requires prior training.
Methods based on neural networks of the ART family	They are a hybrid approach combining neural network and fuzzy methods, retaining all their advantages. They possess the property of stability-plasticity: they are capable of incrementally memorizing new information without losing old information. They do not use explicit methods of fuzzification and defuzzification, which greatly facilitates their configuration.	They cannot fully cope with large flows of monitoring data, begin to form too many different categories in the data structure, and cannot cope with the task of classification in conditions of overlapping classes (this requires modification).

ISSN 2076-2429 (print)
ISSN 2223-3814 (online)

Structural and functional model for supporting management decision-making based on monitoring indicators When processing monitoring data, especially in the context of its operational analysis, the first step is to structure it, i.e., cluster or classify it. Clustering allows for a better "understanding" of the specifics of the incoming data and the identification of possible anomalies, while classification allows for the assignment of current measurements of indicators to one of the previously recorded classes of states of the monitored system. The structuring stage is followed by the decision-making stage, where the necessary corrective actions in a given situation are determined on the basis of the promptly received and already structured information. Neural network models can be used for the automated formation of control actions. Formally, such a system can be described as a tuple:

$$<P_t, K_t, R_t, U_t, N_t>, \tag{1}$$

where:

P_t − input set of monitoring indicator values at time t;

K_t − set of clusters into which the state of controlled objects is divided, taking into account the current values of monitoring indicators;

R_t − classification of possible system states depending on the clusters obtained;

U_t − a set of possible control actions depending on the class of the current state;

N_t − a set of neuro-fuzzy models that establish correspondences:

$P_t \to K_t$ (clustering model);

$K_t \to R_t$ (classification model);

$R_t \to U_t$ (decision rule construction model).

The ART neural network currently consists of 10 models based on the principle of stability-plasticity, the main ones being:

− ART-1 – this network processes binary input vectors;

− ART-2 – this network is designed for clustering continuous input vectors (demonstrates high quality performance);

− ART-2a – a faster version of ART-2, but inferior in quality;

− ARTMAP (combining two ART-1 or ART-2 networks into a single entity) – solves issues with the distribution of information across classes, i.e., it is designed for classification tasks;

− Fuzzy-ART and Fuzzy-ARTMAP – modify the architectures of ART-2 and ARTMAP networks by using fuzzy logic in the calculation process, which allows for better processing of incomplete and noisy data. To solve the problem of approaching the description of the system, it is necessary to formulate the functional principles of ART networks at general stages, namely the following:

All types of existing ART networks have at least one layer of neurons, which during operation are transformed into prototypes of corresponding clusters. The result of clustering is expressed in a specific "recognition" response by the neuron responsible for a certain cluster of the current input vector. If the input vector is not similar to any of the previously formed clusters, this leads to the creation of a new cluster corresponding to the input vector, which, in turn, leads to the addition of another neuron to the network structure, which will be the prototype of the new cluster. The cluster prototype is capable of "learning" and changing. The cluster prototype will not be modified if the input vector is not sufficiently similar to it (which solves the stability-plasticity problem). Input vectors that are not similar to previously processed data can lead to the formation of new clusters, but cannot destroy the accumulated memory of the network and "corrupt" previously formed clusters. To consider the general model of the ART family network, let us introduce several key definitions.

Definition 1 (initial sample). Let $X = \{x^1; x^2; \ldots; x^C\}$ – a set of data (vectors of features or objects). X – the initial sample of power C, each element of which is a vector in R^n type $x^k = (x_1; x_2; \ldots; x_n)$.

Definition 2 (cluster or cluster). A cluster (cluster) is a subset $Z \in X$. The result of clustering is a set of clusters that the clustering algorithm has detected in the dataset. *X.* "Hard" clustering is the division of X into mutually exclusive clusters, while "soft" (or overlapping) clustering allows a data element to belong to more than one cluster.

Definition 3 (prototype). Each j-th ART neural network $(j = \overline{1,M})$ has a vector associated with it $w^j = (w_{j1}; w_{j2}; \ldots; w_{jn})$, which consists of weighting factors w_{ji} on the connections of a neuron j. Such a vector w^j is called the prototype of the cluster, i.e., the internal representation of the cluster described by neuron j. The functionality of the network can be described using a general model, which can be used to analyze the training results of various ART networks. The results of this analysis can be conditionally divided into the following stages [20].

Setting the initial network parameters

In order for the ART network to work as planned, its initial parameters must be initialized in advance. Any network of the ART family includes at least one working layer that contains prototypes of the categories (clusters) created by the network in the input data. All network inputs are connected to each of the prototypes via a weight coefficient matrix $W = (w_{ij})$, which at this stage is usually filled with zeros, since the cluster prototypes are not yet known. At the initial stage, all the main network parameters must be initialized, including the learning rate coefficient β, the parameter regulating the similarity of vectors from the same cluster ρ, and the maximum number of iterations.

Preparation of input vectors (preprocessing)

Depending on the characteristics of the input data and the type of network, different processing of the input vector takes place at this stage. The procedure includes any changes to it: noise removal, normalization, dimension doubling (complementation), etc., without which the network cannot function correctly.

Primary data analysis (clustering)

At this point, each network in the ART family performs an initial evaluation of the input vector using a selection function designed to measure the distance between sets of input features based on one of the existing metrics. In Fuzzy-ART and Fuzzy-ARTMAP networks, fuzzy metrics are used to measure this distance.

Detailed compliance analysis

At this stage, a specific neuron belonging to the prototype layer is activated. It is determined using a special matching function. This function is responsible for the accuracy and non-linearity of the analysis in the recognition process concerning vector similarity. An important circumstance is taken into account here: if the vector and the prototype pass a successful check for functional correspondence with each other, a new stage will begin, in which the weights of the prototype vector will change, i.e., they will undergo training. If the check fails, the activated neuron will be temporarily deactivated. Thus, the selection function determines the cluster closest to the current input vector, and the matching function determines whether the input vector is sufficiently similar to the prototype of that cluster.

Training stage

During the training phase, the weights of the neuron, which is the prototype of the cluster selected in the previous phase, are changed. If none of the existing prototypes pass a detailed functional check for compliance, a new cluster will be created based on the vector that the network was unable to recognize. In this case, a new neuron is activated, the characteristics of which are determined by the coordinates of the unrecognized vector. At the same time, the size of the network increases. That is, the ART network has a growing architecture.

Post-processing (post-processing of results)

After the learning process is complete, the network provides for subsequent processing of the network output values. For example, if a limit on the quantitative composition of clusters is specified in advance, they are combined or separated from each other in accordance with the necessary conditions. Based on the above, we can conclude that in order to solve a specific problem related to the structuring of certain input data, it is necessary to determine the ART network model option depending on the required output structural elements and to complete the stages according to this algorithm. To do this, it is necessary to determine the list of network control parameters, fix the selection function, as well as the correspondence and training functions, and set two basic algorithms: pre- and post-processing. It should be noted that the architecture of the ART family network can include one or more basic modules consisting of three blocks: a learning block, a comparison block, and a recognition block. Figure 2 shows a diagram of the basic module of the ART family networks.

 ISSN 2076-2429 (print)
ISSN 2223-3814 (online)

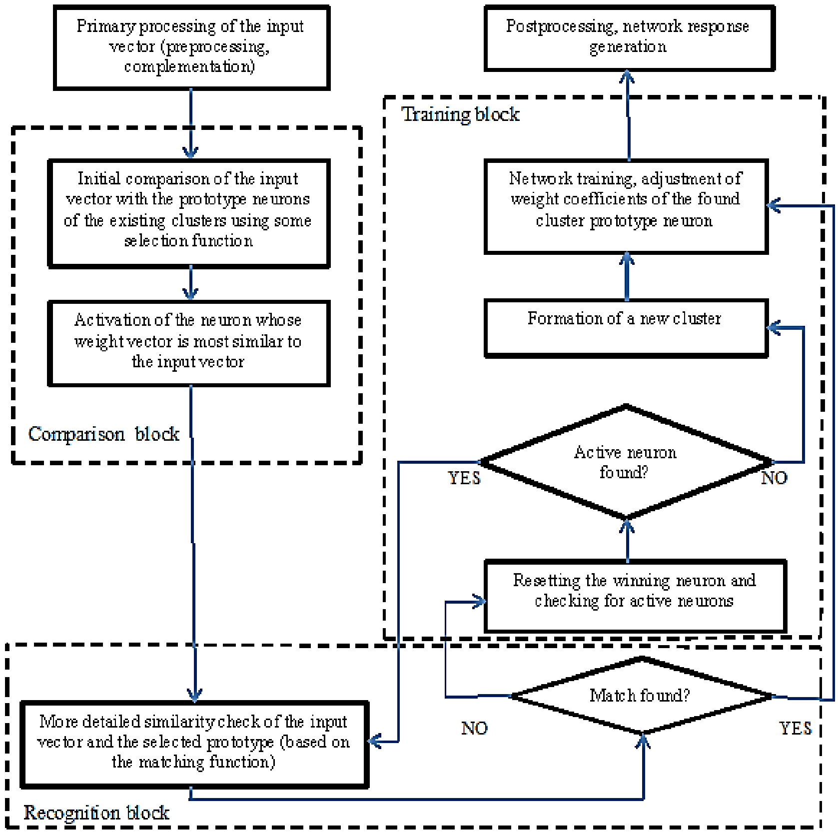

Fig. 2. Algorithm of the basic module of the ART family networks

Networks in the ART family can be trained either supervised or unsupervised. In any case, even if an ART network is supervised, it has a self-organizing module at its core, which divides all input vectors into clusters consisting of similar images. One class can correspond to multiple clusters. The converse is not true in the basic approach; each cluster can only correspond to one output class, which is not always convenient, as classes may overlap when processing real data. For example, the same set of input monitoring indicators may correspond to several problematic situations requiring management decisions, which obviously require modification. The above steps are common to all networks in the ART family, but their implementation varies across different networks in the family. Let us consider the content of the specified stages for two basic networks of this family that solve the clustering problem: the ART-2a network [23-24], which uses conventional arithmetic operations on real input data, and its modification, the Fuzzy ART network [25], which uses fuzzy logic operations in calculations (but does not require fuzzification of inputs). The use of fuzzy operations allows the network to be less sensitive to imprecise measurements and noise in the input data, and it is the neuro-fuzzy variations of ART models that are proposed in this study to be used as a basis for developing decision support systems in the OS in the context of operational analysis of monitoring data. However, to understand the features of the Fuzzy ART functioning, it is first necessary to describe the logic of the ART-2a network, of which Fuzzy ART is a modification.

B The list of parameters of the ART-2a network includes, in particular, the learning rate, which is a real number $\beta \in [0;1]$, normalization parameter θ, which is taken from the interval [0;0.01] and the boundary criterion similarities ρ, also belonging to the segment [0;1]. The ART-2a network requires mandatory normalization of input vectors to unit length before recognition:

$$x_{ij}^{new} = \frac{x_{ij}^{old}}{\sum_{j=1}^{J}(x_{ij}^{old})^2}, \qquad (2)$$

where through J the dimension of the next vector supplied to the input is indicated X^i, and through x_{ij} − his j-th component. Then the coordinates of the normalized vectors whose value became less than the normalization parameter θ, are reset, after which the normalization procedure is repeated once more. This procedure is designed to suppress small random noise in the source data, which could negatively impact the resulting clustering quality. The ART-2a network uses the cosine distance between vectors as a similarity metric, which is calculated as the scalar product of corresponding unit-length vectors:

$$T^k = M_k = (X^i, w^k,) = \sum_{j=1}^{J} x_{ij} w_i^k , \tag{3}$$

where: through X^i again denotes the *i*-th input vector, via T_k the selection function is designated for *k*-th cluster, M_k – This is the value of the established conformity function, vector w^k – This is the currently existing prototype of the *k*-th cluster.

The ART-2a network's operating algorithm consists of the following steps. In the first step, using the cosine metric, the network selects a cluster with number s whose prototype has the greatest similarity to the original vector. X^i, i.e. $s = \arg\max_k T_k$. In the second step for the selected cluster ***k*** the matching condition is checked $M_s > \rho$, where ρ – the similarity criterion value, which sets the upper limit for the homogeneity of vectors in a cluster. If the selected cluster prototype satisfies the similarity condition, the original vector is assigned to that cluster. In this case, it is necessary to recalculate the coordinates of the cluster prototype so as to increase its similarity to the input vector in formula (4):

$$w^{k+1} = \frac{(1-\beta)w^k + \beta * X^i}{\left\|(1-\beta)w^k + \beta * X^i\right\|}, \tag{4}$$

where: β learning speed.

The weight vector modified according to formula (4) undergoes further normalization. Note that the choice of the speed parameter value β the requirement to identify long-term patterns or track data trends is influenced. In the first case, small parameter values are set, while in the second, larger ones are more appropriate. Regarding the parameter ρ (level of similarity), then choosing it is much more difficult. This is both from a theoretical and practical standpoint. Since the researcher does not always have an understanding of the data structure, the task of determining the largest angle at which vectors can be considered similar becomes labor-intensive. This is the reason for a serious drawback of the ART-2a network. Namely, high values ρ lead to an increase in the number of clusters, which is difficult to control. And small values lead to the formation of a single large cluster. A solution to this problem can be obtained by applying the following strategy: decrease the learning rate parameter while monitoring the number of clusters, and stop the process when this number stabilizes. The Fuzzy ART network [24 – 27] is based on a slightly different principle of processing continuous input data. This network uses fuzzy logic operations, but does not require explicit data fuzzification. To process the values of monitoring indicators using this network, scaling of all components of the input and output vectors within the interval [0;1] is required, rather than unit normalization:

$$x_{ij}^{new} = \frac{x_{ij}^{old} - x_i^{\min}}{x_i^{\max} - x_i^{\min}}, \tag{5}$$

where: through x_i^{min} denotes the minimum coordinate of the input vector X_i, and through x_i^{max} – its maximum coordinate.

The structure of this network is shown schematically in Figure 3.

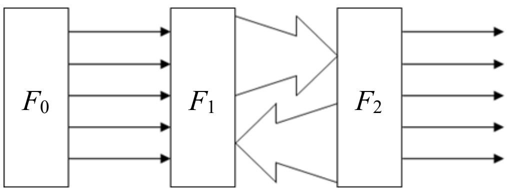

Fig. 3. Fuzzy ART network architecture

ISSN 2076-2429 (print)
ISSN 2223-3814 (online)

The Fuzzy ART network includes layer F_0, which represents the current input vector [28]; layer F_1, which processes both the input from layer F_0 and the feedback from layer F_2, which represents the recognized category (cluster).

Layer output F_0 is a vector $I=(a_1,\dots a_{1M})$, each component of which belongs to a segment [0,1]. Layer F_1 generates a vector at the output $X=(x_1,\dots,x_M)$, and the F_2 layer is a vector $Y=(y_1,\dots,y_N)$. Weight matrix W connects layers together F_1 and F_2, and for each neuron $j(j=1,\dots,N)$ layer F_2 corresponds to the weight vector $W^j=(w_{j1,\dots,}w_{jM})$, Initially it is assumed $w_{j1}=\dots=w_{jM}=1$. At the stage of selecting the winning neuron of layer F_2 of the ART module (shown in Fig. 2. above), the task of finding the global maximum of the selection function can be divided into subtasks for each stream to find its local maximum, and then combined to generate the global maximum using the reduction function. The Fuzzy ART network has a number of hyperparameters α, β, ρ, where β and ρ have the same meaning as in the ART-2a network, and α – a small number (approximately 10^{-6}), responsible for the non-degeneracy of cluster prototypes. Any existing implementation of “fuzzy AND” can be used as the selection function, for example in formula (6):

$$T_k=\frac{\left|X^i \wedge w^k\right|}{\alpha+\left|w^k\right|}, \tag{6}$$

where: $\wedge$ – fuzzy multiplication operator, which is defined here and below as:

$$(p \wedge q)_j=\min(p_j,q_j),\ j=\overline{1,J},$$

and the modulus sign denotes a fuzzy norm: $|p|=\sum_i p_i$. The Fuzzy ART network, if the similarity condition is not satisfied, applies a temporary cluster deactivation mechanism. Note that such a mechanism is not provided in the ART-2a network. Therefore, at the cluster determination stage, the cluster number will be determined by the maximum value of the function according to formula (6) across all active clusters, i.e. $s=\arg\max_k T_k, k \in Q$, where Q the set of numbers of clusters active at a given stage. Next, the s-th prototype is checked against the input vector. This check is performed using the following matching function in formula (7):

$$M_s=\frac{\left|X^i \wedge w^s\right|}{X^i}. \tag{7}$$

An input vector X belongs to cluster s only if: $M_s \ge \rho$, where ρ – similarity criterion [29]. If this inequality is not satisfied, the identified cluster s becomes temporarily inactive, and the selection function according to formula (6) is again applied to select a new candidate cluster. If, during the process, there are no more clusters to select (i.e., all clusters have become deactivated), a new cluster is created based on the input vector. The cluster weights are set equal to the weights of the input vector: $w=X^i$. After the cluster selection procedure [30], the learning function is applied, and the weights of the selected cluster (its prototype) are recalculated according to formula (8):

$$w^{t+1}=\left(1-\beta\right)w^t+\beta(X^i \wedge w^t). \tag{8}$$

In the Fuzzy ART network, weight vectors are not normalized to a unit length, so using this learning function causes a degeneration problem. The operator used, calculated as a coordinate minimum, contributes to a gradual decrease in the coordinates of the weight vectors. Their reduction will gradually lead to the degeneration of cluster prototypes when they forget the information learned earlier. In order to overcome this problem, complementary coding is applied to the input vectors at the preprocessing stage, thanks to which the dimension of the input vector is doubled by adding its components: $x_{i(J+j)}=1-x_{ij}$, which prevents all components of the weight vector w from degenerating to zero values. The main stages of the general scheme for solving the data clustering problem using the two considered networks of the ART family are described below.

Stage 1. Selection of the preprocessing algorithm. At this stage, input data preprocessing is implemented using a number of methods. Ranking of input variables. If information about the importance of input variables is known in advance, they can be ranked, after which the inputs can be scaled according to the assigned ranks (for example, the most important variable is scaled to the interval [0;1], the next most important to the smaller interval [0; 0.9], and so on). This will significantly change the clustering result, as more significant variables will contribute more.

Normalization. In ART-2a and Fuzzy ART networks, normalized input vectors must be used. For Fuzzy ART networks, complementation is performed before normalization.

Noise removal. Noise in the input data can be eliminated by entering validation parameters that set upper and lower limits for each of the input parameters. If a value exceeds these limits, it will be truncated or discarded in accordance with the specified limits in order to remove erroneous values.

Complementation. This method increases the dimension of the input vector by doubling its size. As a result, instead of the input vector $\overline{a}$ the vector is considered:

$$I^a = [\overline{a}\overline{a}^c] = [a_1, \ldots, a_n, a_1^c, \ldots, a_n^c], \text{ where } a_1^c = 1 - a_i .$$

Stage 2. Assigning selection, correspondence, and learning functions. After preprocessing, it is necessary to assign specific types of selection, correspondence, and learning functions. The generalized structure of the ART network family described above makes it possible to change these functions at any stage of network operation, since the architecture and general logic of the network do not depend on their specific type. However, the clustering results and characteristics of the clusters obtained strongly depend on the selected functions, so they must correspond to the task being solved.

Stage 3. Selection of the post-processing algorithm. This stage allows compensating for the shortcomings of the ART family model used after training is complete. For example, one of the main shortcomings of this family of networks is the complexity of calculations when selecting the similarity parameter ρ or its analogue proposed in this study–the integral similarity assessment of input vectors. An incorrect choice of the ρ parameter can lead to the merging of all available data into one large cluster or, conversely, to an uncontrolled increase in the number of clusters. To form a clustering algorithm, it is necessary to select preprocessing and postprocessing schemes. The problem of choosing a post-processing method is more complex, since if the ANN is used to cluster continuously incoming stream data, the number of clusters generated by the network will constantly grow over time, which complicates the analysis of incoming data and the making of management decisions based on it.

Research results

To solve the problem of management in an OS based on monitoring data, it is necessary to improve information support in terms of the ability to process large volumes of continuously incoming data. The first stage of processing is structuring based on cluster analysis. Algorithms for cluster structuring of stream data continuously process incoming input vectors, including in online mode. However, not all clustering algorithms can be used to solve this problem. As noted earlier, in order for the results to be reliable, such algorithms must meet certain conditions:

− display the final result in real time;
− instantly adapt to changing indicators;
− scale for an arbitrary number of objects;
− create a slightly expanding structure as more objects are processed;
− determine the presence of outliers in the data.

Based on this study, a computational experiment was conducted for the FuzzyART network, which involved processing a constantly incoming stream of input data. The stream was simulated using the scikit-learn library, designed to implement machine learning algorithms. This library contains random sample generators, whose data can be used to artificially create information of a controlled size and complexity [31]. The make blobs procedure used in this case helps to create multi-class data sets, each of which corresponds to one or more clusters. Based on the library procedure, a stream of 10-dimensional data consisting of three clusters of equal size was generated to analyze the possibility of using ART family networks for processing stream data. One of them was easily separable, while the other two were closely located. An illustration of this arrangement (for two-dimensional input vectors) is shown in Figure 4. Specific results were obtained during the computational experiment (Table 2). Errors in the operation of the networks under consideration are considered to be the discrepancy between the obtained result and the model classification of the initial data (the percentage of incorrect clustering).

ISSN 2076-2429 (print)
ISSN 2223-3814 (online)

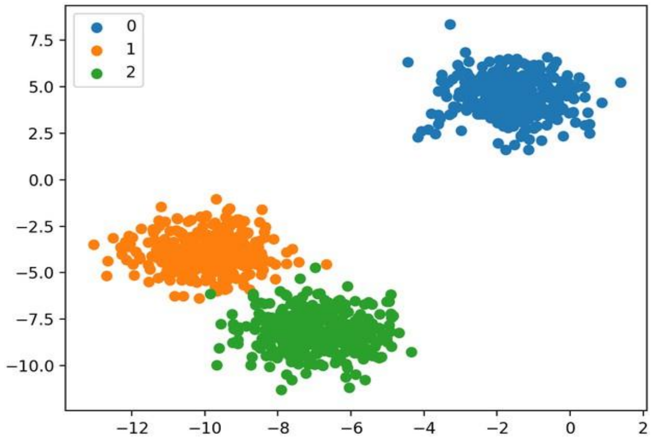

Fig. 4. Illustration of the mutual arrangement of generated clusters

Table 2

Dependence of the Fuzzy ART network operation on parameters.

0< ρ <0.68					
β	0.1	0.3	0.5	0.7	0.9
Errors, (%)	64%	60%	65%	66%	63%
Number of clusters	1	1	1	1	1
0.68≤ ρ<0.85					
β	0.1	0.3	0.5	0.7	0.9
Errors, (%)	33%	33%	6.6%	12%	12%
Number of clusters	2	2	3	4	4
0.85≤ ρ ≤ 0.9					
β	0.1	0.3	0.5	0.7	0.9
Errors, (%)	12%	12%	9.2%	6.6%	6.6%
Number of clusters	4	4	4	3	3

Conclusions

Based on the results of this experiment, the following conclusions were made regarding the applicability of the Fuzzy ART network for streaming clustering of monitoring data. Before obtaining the value ρ =0.68 the network connected the incoming vectors into a common cluster.

Meaning 0.68<ρ<0.85 indicates that the network divides the set into two clusters, one of which is easily separable, while the other consists of the remaining elements. Division into three clusters occurs when the value is 0.85≤ρ≤ 0.9. When there is a further increase ρ>0.9, the network begins to create an increasing number of clusters, i.e., the initial sample is divided into many small groups. It should be noted that Fuzzy ART has a number of positive functional characteristics, including stability, fast convergence, and clear dependence on the selected learning rate. However, during the experiment, it became clear that the functions used by Fuzzy ART for selection, learning, and matching can lead to high sensitivity to the order of the indicators provided. It was noted that if the data is fed into the input in a random order (shuffled) at the beginning of operation, the distribution across clusters will be much better. To solve the problem of uncontrolled growth in the number of clusters, this study proposes using the post-processing method [32 – 34], which allows the number of clusters to be limited to a specified value. However, it may not be known in advance what number to use to limit the number of clusters created. Therefore, this study proposes a special method for recalculating the similarity parameter ρ (which is responsible for the number of clusters). According to this method, it is proposed to set the maximum permissible number of clusters K at the initial stage of the algorithm and set the similarity parameter ρ very close to 1. If, during the execution of the next stage, the vector does not correspond to any of the existing clusters and the maximum possible number of clusters K has been reached, then a new cluster cannot be created. Thus, a new cluster will be created, but the total number of clusters will not change. The proposed approach to limiting the number of clusters is sufficiently versatile and can be easily applied to modify the training of any network of the ART family. Let us now present general recommendations obtained as practical results in the course of a computational experiment concerning the FuzzyART network, which was engaged in processing a constant data stream. In each specific case, the network should be selected based on the type of indicators and the operating mode in which it functions. The Fuzzy ART network is very sensitive to the order of input data organization, based on which it is capable of generating and obtaining different results of the clustering process. Therefore, at the initial stage, it is recommended to perform several iterations in batch mode (not online).

Select or set training speed parameters β It depends on the type of task being performed in the process. Here, a low initial learning rate allows you to identify general patterns in the results obtained, while a high rate allows you to quickly catch and adapt to existing trends.

Apply the strategy of adjusting the similarity parameter ρ. Specifically, at the beginning of the algorithm, set the maximum number of clusters K, and set ρ=1. If it is necessary to create during the operation of the algorithm $(K+1)$-ht cluster (i.e., the algorithm cannot classify the input vector into existing clusters), it is necessary to reduce the parameter ρ and repeat the procedure again. The parameter will decrease until the input vector is assigned to one of the existing clusters. At the same time, if in the process of decreasing ρ the prototypes of any two of the existing clusters become similar in terms of the current value of the similarity criterion, then these two clusters should be merged into one common cluster. When the network has low indicators β it works slowly and has long cluster migration, so Fuzzy-ART is only used for data that comes in a large but discrete stream (like sensor readings taken a few times a day). In this case, the data can be processed in batch mode. In a scenario where noise spikes occur in the indicator stream, with the appearance of atypical input vectors that are remembered by new clusters, the cluster limit may be exceeded. Therefore, when analyzing the results, the expert will either study the small clusters that have appeared (if one of the goals of the analysis is to find anomalies in the data), or delete them (so that they do not create unnecessary noise), or reduce the requirements for the number of clusters. The latter approach, given the chaotic order of input data, will lead to the absorption of small clusters by the main clusters. The most commonly used quality metrics in clustering tasks are the average intracluster distance, which must be minimized, or the average intercluster distance, which is maximized. The silhouette coefficient metric [31] combines both of these approaches in a sense, which is why it is proposed for use in evaluating the quality of clustering. The silhouette coefficient is used to evaluate the difference between objects within clusters and objects in other clusters. The silhouette coefficient can range from –1 to 1. The higher the silhouette coefficient, the better the quality of the clustering. Negative values indicate poor or incorrect clustering, values close to zero indicate overlap and superimposition of clusters, and values close to 1 indicate tightly grouped clusters. In the future, the quality of the developed algorithms will be evaluated using this metric.

Література

1. Amis G. P., Carpenter G. A. Self-supervised ARTMAP. *Neural Networks*. 2010. № 23. P. 265–282.
2. Brito da Silva L. E., Elnabarawy I., Wunsch D. C. 2nd. Dual vigilance fuzzy adaptive resonance theory. *Neural Networks*. 2019. № 109. P. 1–5.
3. Nandwani P., Verma R. A review on sentiment analysis and emotion detection from text. *Social Network Analysis and Mining*. 2021. Vol. 11, № 81. DOI: https://doi.org/10.1007/s13278-021-00776-6.
4. Fuzzy ARTMAP: An adaptive resonance architecture for incremental learning of analog maps / G. A. Carpenter et al. *Proc. of the Int. Joint Conf. on Neural Network (Boston, 1992)*. 1992.
5. Acheampong F. A., Wenyu C., Nunoo-Mensah H. Text-based emotion detection: Advances, challenges, and opportunities. *Engineering Reports*. 2020. DOI: https://doi.org/10.1002/eng2.12189.
6. Challenges and opportunities of text-based emotion detection: A survey / A. A. Maruf et al. *IEEE Access*. 2024. Vol. 12. P. 18416–18450. DOI: https://doi.org/10.1109/ACCESS.2024.3356357.
7. Probabilistic ensemble Fuzzy ARTMAP optimization using hierarchical parallel genetic algorithms / K. L. Chu et al. *Neural Computing and Applications*. 2014. № 26. P. 263–276.
8. French R. M. Catastrophic forgetting in connectionist networks. *Trends Cogn. Sci.* 1999. № 3. P. 128–135.
9. Acheampong F. A., Nunoo-Mensah H., Chen W. Transformer models for text-based emotion detection: a review of BERT-based approaches. *Artificial Intelligence Review*. 2021. Vol. 54. P. 5789–5829. DOI: https://doi.org/10.1007/s10462-021-09958-2.

ISSN 2076-2429 (print)
ISSN 2223-3814 (online)

10. Zhang X., Mao R., Cambria E. Multilingual Emotion Recognition: Discovering the Variations of Lexical Semantics between Languages. *International Joint Conference on Neural Networks (IJCNN) (Yokohama, Japan, 2024)*. 2024. P. 1–9. DOI: https://doi.org/10.1109/IJCNN60899.2024.10651409.
11. Ghosh J., Acharya A. Cluster ensembles. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*. 2011. Vol. 1, № 4. P. 305–315.
12. Grossberg S. The resonant brain: how attentive conscious seeing regulates action sequences that interact with attentive cognitive learning, recognition, and prediction. *Atten. Percep. Psychophys.* 2019. № 81. P. 2237–2264.
13. De León Languré A., Zareei M. Improving text emotion detection through comprehensive dataset quality analysis. *IEEE Access*. 2024. Vol. 12. P. 70489–70500. DOI: https://doi.org/10.1109/ACCESS.2024.3491856.
14. Kharchenko V., Fesenko H., Illiashenko O. Quality models for artificial intelligence systems: characteristic-based approach, development and application. *Sensors*. 2022. Vol. 22, № 13. Art. 4865. URL: www.scopus.com. DOI: https://doi.org/10.3390/s22134865.
15. Optimizing deep learning inference on embedded systems through adaptive model selection / V. Marco et al. *ACM Transactions on Embedded Computing Systems*. 2020. Vol. 19, № 1. P. 1–28. DOI: https://doi.org/10.1145/3371154.
16. Haupt S. E., Pasini A., Marzban C. Environmental Optimization: Applications of Genetic Algorithms. *Artificial Intelligence Methods in the Environmental Sciences*. 2009. P. 379–395.
17. Keskin G. A., Ozkan C. An Alternative Evaluation of FMEA: Fuzzy ART Algorithm. *Quality and reliability engineering international*. 2009. Vol. 25, № 6. P. 647–661.
18. Overcoming catastrophic forgetting in neural networks / J. Kirkpatrick et al. *Proc. Natl. Acad. Sci. U.S.A.* 2017. № 114. P. 3521–3526.
19. Kuncheva L. I., Whitaker C. J. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning*. 2003. № 51. P. 181–207.
20. Le Cun Y., Bengio Y., Hinton G. Deep learning. *Nature*. 2015. № 521. P. 436–444.
21. Massey L. Real-world text clustering with adaptive resonance theory neural networks. *Proceedings of 2005 international joint conference on neural network (Montreal, Canada, July 31–August 4, 2005)*. 2005.
22. Fuzzy ART Neural Network Parallel Computing on the GPU / M. Martinez-Zarzuela et al. *Computational and Ambient Intelligence*. 2007. P. 463–470.
23. Meng L., Tan A.-H., Wunsch D. C. Vigilance adaptation in adaptive resonance theory. *In Proc. IEEE International Joint Conference on Neural Networks (IJCNN)*. 2013. P. 1–7.
24. Oong T. H., Isa N. A. M. Feature-Based Ordering Algorithm for Data Presentation of Fuzzy ARTMAP Ensembles. *IEEE Transactions on Neural Networks and Learning Systems*. 2014. № 25. P. 812–819.
25. Rosenberg A., Hirschberg J. Detecting pitch accent using pitch-corrected energy-based predictors. *In Interspeech*. 2007. P. 2777–2780.
26. Rousseeuw P. Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis. *Computational and Applied Mathematics*. 1987. № 20. P. 53–65.
27. Pedregosa F. et al. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*. 2011. № 12. P. 2825–2830.
28. Seto K., Liu W. Comparing ARTMAP neural network with the maximum-likelihood classifier for detecting urban change. *Photogrammetric Engineering & Remote Sensing*. 2003. № 69(9). P. 981–990.
29. Solyanik S. A., Kravets O. JA. Development of distributed management complexes for territorial information systems data flows. Yelm, WA, USA : Science Book Publishing House LLC, 2020.
30. Skovajsova L., Rojcek M. Soft clustering algorithms based on neural networks. *IEEE 12th International Symposium on Computational Intelligence and Informatics (Budapest, 2011)*. 2011. P. 439–442.
31. Wasserman P. D. Neural Computing: Theory and Practice. Van Nostrand Reinhold Co., 1989. 230 p.
32. Versace M., Kozma R. T., Wunsch D. C. Adaptive Resonance Theory Design in Mixed Memristive-Fuzzy Hardware. *Advances in Neuromorphic Memristor Science and Applications*. 2012. P. 133–153.
33. Xu R., Wunsch D. C. 2nd. Survey of Clustering Algorithms. *IEEE transactions on neural networks*. 2005. Vol. 16, № 3. P. 645–678.
34. Calibrated learning for online distributed power allocation in small-cell networks / X. Zhang et al. *IEEE Transactions on Signal Processing*. 2019. Vol. 67, № 11. P. 8124–8136.

References

1. Amis, G. P., & Carpenter, G. A. (2010). Self-supervised ARTMAP. *Neural Networks*, *23*, 265–282.
2. Brito da Silva, L. E., Elnabarawy, I., & Wunsch, D. C., 2nd. (2019). Dual vigilance fuzzy adaptive resonance theory. *Neural Networks*, *109*, 1–5.

3. Nandwani, P., & Verma, R. (2021). A review on sentiment analysis and emotion detection from text. *Social Network Analysis and Mining*, *11*(81). https://doi.org/10.1007/s13278-021-00776-6.
4. Carpenter, G. A., Grossberg, S., Markuzon, N., Reynolds, J. H., & Rosen, D. B. (1992). Fuzzy ARTMAP: An adaptive resonance architecture for incremental learning of analog maps. *Proceedings of the International Joint Conference on Neural Networks*, 1992.
5. Acheampong, F. A., Wenyu, C., & Nunoo-Mensah, H. (2020). Text-based emotion detection: Advances, challenges, and opportunities. *Engineering Reports*. https://doi.org/10.1002/eng2.12189.
6. Maruf, A. A., Khanam, F., Haque, M. M., Jiyad, Z. M., Mridha, M. F., & Aung, Z. (2024). Challenges and opportunities of text-based emotion detection: A survey. *IEEE Access*, *12*, 18416–18450. https://doi.org/10.1109/ACCESS.2024.3356357.
7. Chu, K. L., Liew, W., Seera, M., & Lim, E. (2014). Probabilistic ensemble Fuzzy ARTMAP optimization using hierarchical parallel genetic algorithms. *Neural Computing and Applications*, *26*, 263–276.
8. French, R. M. (1999). Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences*, *3*, 128–135.
9. Acheampong, F. A., Nunoo-Mensah, H., & Chen, W. (2021). Transformer models for text-based emotion detection: A review of BERT-based approaches. *Artificial Intelligence Review*, *54*, 5789–5829. https://doi.org/10.1007/s10462-021-09958-2.
10. Zhang, X., Mao, R., & Cambria, E. (2024). Multilingual emotion recognition: Discovering the variations of lexical semantics between languages. *International Joint Conference on Neural Networks (IJCNN)*, 1–9. https://doi.org/10.1109/IJCNN60899.2024.10651409.
11. Ghosh, J., & Acharya, A. (2011). Cluster ensembles. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *1*(4), 305–315.
12. Grossberg, S. (2019). The resonant brain: How attentive conscious seeing regulates action sequences that interact with attentive cognitive learning, recognition, and prediction. *Attention, Perception, & Psychophysics*, *81*, 2237–2264.
13. De León Languré, A., & Zareei, M. (2024). Improving text emotion detection through comprehensive dataset quality analysis. *IEEE Access*, *12*, 70489–70500. https://doi.org/10.1109/ACCESS.2024.3491856.
14. Kharchenko, V., Fesenko, H., & Illiashenko, O. (2022). Quality models for artificial intelligence systems: Characteristic-based approach, development and application. *Sensors*, *22*(13), 4865. https://doi.org/10.3390/s22134865.
15. Marco, V., Taylor, B., Wang, Z., & Elkhatib, Y. (2020). Optimizing deep learning inference on embedded systems through adaptive model selection. *ACM Transactions on Embedded Computing Systems*, *19*(1), 1–28. https://doi.org/10.1145/3371154.
16. Haupt, S. E., Pasini, A., & Marzban, C. (2009). Environmental optimization: Applications of genetic algorithms. In *Artificial Intelligence Methods in the Environmental Sciences* (pp. 379–395).
17. Keskin, G. A., & Ozkan, C. (2009). An alternative evaluation of FMEA: Fuzzy ART algorithm. *Quality and Reliability Engineering International*, *25*(6), 647–661.
18. Kirkpatrick, J., Pancanu, R., Rabinowitz, N., Veness, J., Desjarkins, G., Rusu, A. A., ... & Hadsell, R. (2017). Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, *114*, 3521–3526.
19. Kuncheva, L. I., & Whitaker, C. J. (2003). Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning*, *51*, 181–207.
20. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*, 436–444.
21. Massey, L. (2005). Real-world text clustering with adaptive resonance theory neural networks. *Proceedings of 2005 International Joint Conference on Neural Networks*.
22. Martinez-Zarzuela, M., Diaz Pernas, F. J., Diez Higuera, J. F., & Rodriguez, M. A. (2007). Fuzzy ART neural network parallel computing on the GPU. In *Computational and Ambient Intelligence* (pp. 463–470).
23. Meng, L., Tan, A.-H., & Wunsch, D. C. (2013). Vigilance adaptation in adaptive resonance theory. *2013 IEEE International Joint Conference on Neural Networks (IJCNN)*, 1–7.
24. Oong, T. H., & Isa, N. A. M. (2014). Feature-based ordering algorithm for data presentation of Fuzzy ARTMAP ensembles. *IEEE Transactions on Neural Networks and Learning Systems*, *25*, 812–819.
25. Rosenberg, A., & Hirschberg, J. (2007). Detecting pitch accent using pitch-corrected energy-based predictors. *Interspeech 2007*, 2777–2780.
26. Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, *20*, 53–65.
27. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

ISSN 2076-2429 (print)
ISSN 2223-3814 (online)

28. Seto, K. C., & Liu, W. (2003). Comparing ARTMAP neural network with the maximum-likelihood classifier for detecting urban change. *Photogrammetric Engineering & Remote Sensing*, *69*(9), 981–990.
29. Solyanik, S. A., & Kravets, O. J. (2020). *Development of distributed management complexes for territorial information systems data flows*. Science Book Publishing House LLC.
30. Skovajsova, L., & Rojcek, M. (2011). Soft clustering algorithms based on neural networks. *2011 IEEE 12th International Symposium on Computational Intelligence and Informatics*, 439–442.
31. Wasserman, P. D. (1989). *Neural computing: Theory and practice*. Van Nostrand Reinhold Co.
32. Versace, M., Kozma, R. T., & Wunsch, D. C. (2012). Adaptive resonance theory design in mixed memristive-fuzzy hardware. In *Advances in Neuromorphic Memristor Science and Applications* (pp. 133–153).
33. Xu, R., & Wunsch, D. C., 2nd. (2005). Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, *16*(3), 645–678.
34. Zhang, X., Nakhai, M. R., Zheng, G., Lambotharan, S., & Ottersten, B. (2019). Calibrated learning for online distributed power allocation in small-cell networks. *IEEE Transactions on Signal Processing*, *67*(11), 8124–8136.

Тігарєв Володимир Михайлович; Volodymyr Tigariev, ORCID: https://orcid.org/0000-0001-8492-6633
Лопаков Олексій Сергійович; Oleksii Lopakov, ORCID: https:// orcid.org/0000-0001-6307-8946
Космачевський Володимир Володимирович; Volodymir Kosmachevskiy, ORCID: https://orcid.org/0000-0002-3234-2297
Люшенко Андрій Миколайович; Andrii Liushenko, ORCID: https://orcid.org/0009-0004-0410-3100

Accepted October 05, 2025

Accepted November 11, 2025