

20. Wu, X. A Comparison of Objects with Frames and OODBs [Text] / X. Wu // Object Currents. – 1996. – Vol. 1, Issue 1.
21. Minskiy, M. Freymy dlya predstavleniya znaniy [Text] / M. Minskiy. – Moscow: Energiya, 1979. – 152 p.
22. Iskusstvennyy intellekt. Kn. 2. Modeli i metody [Text]: spravochnik / D. A. Pospelov (Ed.). – Moscow: Radio i svyaz', 1990. – 304 p.
23. Maciaszek, L. A. Requirements Analysis and System Design [Text] / L. A. Maciaszek. – 2nd ed. – Reading: Addison Wesley, Harlow England, 2005. – 504 p.
24. Gavrilov, A. V. Sistemy iskusstvennogo intellekta [Text] / A. V. Gavrilov. – Novosibirsk: NGTU, 2004. – 59 p.
25. Savitch, W. Java: An Introduction to Computer Science and Programming [Text] / W. Savitch. – 2nd ed. – Pearson: Prentice Hall, Inc, 2001. – 1039 p.
26. Deitel, H. M. C++ How to Program [Text] / H. M. Deitel, P. J. Deitel. – 5th ed. – Pearson: Prentice Hall, Inc, 2005 – 1536 p.
27. Levykin, V. M. Issledovanie i razrabotka freymovoy modeli struktury dokumenta [Text] / V. M. Levykin, M. A. Kernosov // Novi tekhnolohyi. – 2008. – Issue 1 (19). – P. 149–154.
28. Evlanov, M. V. Modeli patternov proektirovaniya trebovaniy k informacionnoy sisteme na urovne dannyh [Text] / M. V. Evlanov // Radioelektronni i kompiuterni systemy. – 2014. – Issue 1 (65). – P. 128–138.
29. Levykin, V. M. Parallelnoe proektirovanie informacionnogo i programmnogo kompleksov informacionnoy sistemy [Text] / V. M. Levykin, M. V. Evlanov, V. S. Sugrobov // Radiotekhnika. – 2006. – Issue 146. – P. 89–98.
30. Yevlanov, M. V. Paterny proektuvannia vymoh do informatsiynoi systemy [Text] / M. V. Yevlanov // Visnyk natsionalnoho universytetu «Lvivska politehnika». – 2014. – Issue 783. – P. 429–434.

*Розроблено алгоритм навчання багатоваріантного екстрактора ознак, що використовує принципи нейронного газу та розрідженого кодування. Запропоновано інформаційно-екстремальний метод двійкового кодування ознакового подання для побудови вирішальних правил. Це дозволяє зменшити вимоги до обсягів навчальних даних і обчислювальних ресурсів та забезпечити високу достовірність прогнозування порушення умов договору про рівень обслуговування в хмарному середовищі*

*Ключові слова: датацентр, розріджене кодування, нейронний газ, інформаційний критерій, машинне навчання, ройовий алгоритм*

*Разработан алгоритм обучения многослойного экстрактора признаков, использующий принципы нейронного газа и разреженного кодирования. Предложен информационно-экстремальный метод двоичного кодирования признакового представления для построения решающих правил. Это позволяет уменьшить требования к объемам обучающих данных и вычислительных ресурсов и обеспечить высокую достоверность прогнозирования нарушения условий договора об уровне обслуживания в облачной среде*

*Ключевые слова: датацентр, разреженное кодирование, информационный критерий, машинное обучение, роевой алгоритм*

UDC 004.891.032.26:616.127–073.7

DOI: 10.15587/1729-4061.2017.110073

# DEVELOPMENT OF THE METHOD OF FEATURES LEARNING AND TRAINING DECISION RULES FOR THE PREDICTION OF VIOLATION OF SERVICE LEVEL AGREEMENT IN A CLOUD-BASED ENVIRONMENT

V. Moskalenko

PhD, Associate Professor\*

E-mail: systemscoders@gmail.com

A. Moskalenko

Assistant\*

E-mail: a.moskalenko@cs.sumdu.edu.ua

S. Pimonenko

Postgraduate student\*

E-mail: pstsnets@gmail.com

A. Korobov

Postgraduate student\*

E-mail: artemkorr@gmail.com

\*Department of Computer Science

Sumy State University

Rimskoho-Korsakova str., 2, Sumy, Ukraine, 40007

## 1. Introduction

The increase in popularity of cloud-based services stimulates spreading of distributed centers of data processing on

the global scale, which leads to numerous problems in terms of resource planning for different administrative domains. Effective resource planning implies simultaneous provision of minimized violation of Service Level Agreement, SLA,

decreasing the cost of using cloud-based services and increasing the level of energy saving, as well as the profit of a service provider. However, non-stationarity of demand for cloud resources generates variable load peaks, making ineffective the mechanisms of reactive scaling of services that initialize the process of allocation of additional resources only after exceeding a critical value by a particular metric. That is why an active area of research is proactive and predictive principles of resource management that allow us to initialize allocation of necessary resources in advance. In addition, the use of predictive mechanisms makes it possible to provide effective redistribution of resources by identifying unsuccessful candidates (data centers or individual servers) for hosting virtual machines. In this case, prediction of SLA violation allows removing uncertainty regarding the functional state of services at different levels of the cloud system and increase efficiency of multi-criteria optimizing algorithms when planning allocation of resources [1].

One of the approaches to prediction of SLA violation at different levels of a cloud-based system is to use the ideas and methods of machine learning, which form a predictive model by analyzing time series of changes in key indicators of performance, key indicators of quality and system messages [2]. However, the use of traditional one-level methods of machine learning, which is characterized by exponential dependence of the number of model parameters on the number of recognition features, under conditions of multi-dimensionality of observation leads to an increase in requirements for computing resources and the volume of learning data. That is why the most promising direction of synthesis of analytic tools of the system of cloud environment management is usage of the methods of feature learning. These methods are designed to generate informative dictionary of independent features of the higher level of abstract character with relatively low dimensionality, which greatly simplifies the synthesis of decision rules.

Thus, development of the method of learning features and decision rules for prediction of SLA violation is a relevant direction of research as it is directed at increasing efficiency of the system of cloud environment management.

---

## 2. Literature review and problem statement

---

The main problem of service providers of cloud environment is to determine the best compromise between profit and users' satisfaction. However, solution of this problem is complicated by a priori uncertainty regarding the functional state of the service as a result of non-stationarity of demand and heterogeneity of physical and virtual components of IT-infrastructure. Papers [3, 4] suggested application of the algorithms of a decision tree, random forests and Naïve Bayes to remove uncertainty concerning compliance with SLA conditions, associated with exceeding service response time, service availability or a decrease in information safety. However, a small number of features were controlled in proposed approaches, which prevented obtaining a highly reliable predictive model for advances period of time sufficient for the use of necessary measures. The authors of [4, 5] proposed to examine the trends of using resources within a sliding window of the assigned size for the formation of feature description of predicted functional states. In study [5], the prediction model is based on recurrent neural network of Long Short-Term Memory, LSTM. The use of such a net-

work made it possible to reduce response time of the services, but the experiment was carried out on virtual simulators and on data of limited volume tracing. In this case, LSTM network is quite deep while deployed in time and requires large amounts of training dataset in order to avoid the overfitting effect, which makes the model ineffective for a long time of service operation. Paper [6] considers prediction of SLA violations as a result of overloading of network channels in the IT-infrastructure of the data center, based on a deep model that just needs a large amount of training dataset to avoid convergence to the local extremum of function of losses.

Development of the ideology of autonomous computing in cloud-based systems causes research and implementation of technologies of predictive analysis at all levels of information communication system. Telemetry data that are accumulated in the data center management system are characterized by high dimensionality, lack in the balance of the number of samples, representing functional states of services and relatively small amount of labeled data on SLA violation, especially at the beginning of deployment of new services. The authors of [7, 8], in order to analyze high-dimensional data with a small number of labeled samples, propose to use unsupervised feature learning for the full dataset and to carry out training of the classifier of functional states on labeled samples, encoded by learned features. Articles [9, 10] show a high efficiency of neural network algorithms for feature learning based on stacking of autoencoders and restricted Boltzmann machines. However, this approach requires very large amount of data and computational resources, which increases costs on data analysis and delays in building up an effective model for prediction of the state of individual services. That is why the methods of matrix factorization for analyzing multi-dimensional samples and stacking into a multilayer structure, based on nonlinear transformation and pooling operator, are actively explored [8, 11]. These methods include Principal component analysis, PCA, and Independent component analysis, ICA, and Non-negative matrix factorization, NMF. Papers [11, 12] show that the most effective factorization is the one, which provides sparse data representation. Sparse encoding allows getting noise immunity of compact representation of input data, where each observation can be represented as a linear combination of a small number of basic vectors, which makes its interpretation and subsequent analysis easier.

Papers [12, 13] propose the algorithm of sparsely encoding neural gas that allows us to perform incremental unsupervised learning of feature basis according to principles of self-organization and Orthogonal Matching Pursuit, OMP. In this case, the algorithm of sparse encoding neural gas is suitable for samples of limited volume. The proposed algorithm showed high efficiency in analysis of images and noisy signals, however, its organization in a multilayer structure for simplifying of analysis of multi-dimensional observations of little formalized process has not been explored yet.

The most effective methods of machine learning for labeled samples of limited number are based on building an optimum separate hypersurface in the framework of a geometric approach. Articles [10, 11] consider the use of the method of supporting vectors, which perform space transformation for construction of a separate hypersurface, however, its application requires computational time-consuming regularization of the model by selecting kernels and regularization coefficient. The authors of [14, 15] propose the method of transformation of the space of original features

using computationally efficient operations of comparison and “excluding OR” for building separate “hyperspheres” (hyper-parallelepiped) in the binary space of secondary features. In this case, binary feature encoding and population algorithm of optimization of parameters of decision rules by the information criterion makes it possible to create automatically an effective model of a classifier, which makes application of the approach for analysis of monitoring of cloud systems data promising.

### 3. The aim and objectives of the study

The aim of present research is to increase efficiency of formation of feature description and decision rules for the prediction of violation of SLA conditions in a cloud-based data center.

To accomplish the set goal, the following tasks had to be solved:

- to develop a method of learning of hierarchical feature extractor based on ideas and methods of neural gas and sparse encoding of observations and to compare its effectiveness with the autoencoder;
- to develop algorithms of machine learning for a system for prediction of SLA violation using binary feature encoding and populational optimization of parameters of decision rules by the information criterion;
- to explore dependence of reliability of prediction decisions of a system that are made in operating mode for test data, from parameters of feature extractor and decision rules.

### 4. Algorithms of feature learning and decision rules

Collection of observations for feature learning goes on by scanning the archive history of changing the metrics of productivity of info-communicative service by the fixed-size window  $W$ , within which their values are read in time with an assigned step  $\Delta$ . For training decision rules, the sample of these windows with a classified service state at the moment of time is formed, which is ahead of the window by  $\Delta t$  steps. In this case, two functional states are considered – class  $X_1^o$  – normal functioning state, and  $X_2^o$  – violation of SLA conditions.

An important step of data analysis is preliminary normalization with the view to removing linear correlation of components of observation and the unification of primary feature representation. Data whitening with the use of the method of ZCA (Zero-phase Component Analysis) is one of the most common methods of preliminary data normalization. ZCA method implies performance of the following steps:

- 1) calculation of mean selected value of features  $\mu = \text{mean}(X)$ ;
- 2) calculation of co-variative matrix of selected observations  $\Sigma = \text{cov}(X)$ ;
- 3) singular decomposition of co-variative matrix  $\Sigma \approx VDT^T$ ;
- 4) whitening of each observation by formula

$$x_j := VD^{-1/2}V^T(x_j - \mu).$$

In general case, learning of feature representation implies the search for a set of parameters by unlabeled data, for example, in the form of a set of basis vectors  $C$ , which are

subsequently used by the algorithm of encoding for reconstruction of input data distribution. For building a dictionary of basis vectors  $C$ , it is possible to use the algorithms of vector quantization, such as  $k$ -mean or neural gas. Neural gas is based on the principles of “mild” competition, which is why it is characterized by better convergence, independence on the initial search point and more optimum distribution of vectors of a code book. Formation of the feature representation can be performed by one of the method of sparse approximation, for example, the method of orthogonal matching pursuit (OMP). However, the method of optimized orthogonal matching pursuit (Optimized OMP, OOMP) [13] is more effective in terms of minimization of the norm of approximation of residual. Implementation of encoding in OOMP method is an iterative procedure and includes the following major steps:

- 1) search for the  $l$ -th column of the matrix of formed basis vectors  $C$ , which has not been selected (not added to set  $U$ ) yet, with the aim of minimizing the norm of the obtained residue at the current step:

$$l_{\min} := \arg \min_{l \in U} \min_a \|x_j - C^U a\|_2^2;$$

- 2) updating of a set of selected basis vectors

$$U := U \cup l_{\min};$$

- 3) solution of optimization problem

$$a_j^{OMP} := \arg \min_a \|x_j - C^U a\|_2^2;$$

- 4) calculation of current residue

$$\epsilon_j := x_j - C a_j^{OMP};$$

- 5) moving to step 1 until  $k$ -iterations are completed.

To decrease computational complexity of the first step, it is possible to use the population-based search algorithm or implementation, proposed in paper [12], where temporary matrix  $R$  is introduced, which in the beginning is equal to  $R = (r_1, \dots, r_l, \dots, r_M) = C$  at  $\epsilon_j^U := x_j^U$ , and at each step is specified by formula

$$r_l := r_l - (r_l^T r_l) r_{l_{\min}}, \quad (1)$$

where  $r_{l_{\min}}$  is the column of matrix  $R$  that has maximum overlap with the current residue  $\epsilon_j^U$ , the index of which has not yet been added to  $U$ , but determined from formula

$$l_{\min} := \arg \max_{l \in U} (r_l^T \epsilon_j^U)^2. \quad (2)$$

In the same way, value of remainder is updated during each iteration

$$\epsilon_i^U := \epsilon_i^U - (r_{l_{\min}}^T \epsilon_i^U) r_{l_{\min}}. \quad (3)$$

Neural gas, which is used to search for  $C$ , is the algorithm of self-organization of unstructured grid for identifying topological data structure. In general case, the algorithm of neural gas includes the following basic steps:

- 1) initialization of dictionary  $C = (c_1, \dots, c_M)$  by random values from uniform distribution;
- 2) selection of the  $t$ -th input observation  $x$  from set  $X$ , which has volume  $t_{\max}$ ;

3) calculation of coefficients of dimensions of vicinity of neighborhood and learning speed from formulas:

$$\lambda_t := \lambda_0 (\lambda_{\text{final}} / \lambda_0)^{t/t_{\text{max}}}, \quad (4)$$

$$\alpha_t := \alpha_0 (\alpha_{\text{final}} / \alpha_0)^{t/t_{\text{max}}}, \quad (5)$$

where  $\lambda_0, \lambda_{\text{final}}$  are the initiative and final values of coefficient  $\lambda_t$ ;  $\alpha_0, \alpha_{\text{final}}$  are the initial and final values of coefficient  $\alpha_t$ ;

4) calculation of the distance of input vector  $x$  to the words of code book  $C$  and their arrangement in ascending order

$$\|x - c_{i_k}\| \leq \dots \leq \|x - c_{i_k}\| \leq \dots \leq \|x - c_{i_{M-1}}\|;$$

5) performance of  $M-1$  iterations of updating of code words from formula

$$c_{i_k} := c_{i_k} + \alpha_t e^{-k/\lambda_t} (x - c_{i_k}), \quad k = \overline{1, M-1};$$

6) moving to step 2, if  $t < t_{\text{max}}$ .

For adaptation of the algorithm of vector quantization to a selected encoding scheme in order to reduce errors of sparse approximation, we propose to use the modified algorithm, studied in paper [13]. A modified algorithm of neural gas for sparse encoding of observations consists of the following steps:

1) initialization of dictionary  $C = (c_1, \dots, c_M)$  by random values from uniform distribution;

2) selection of the  $t$ -th input observation  $x$  from set  $X$ , which has volume  $t_{\text{max}}$ ;

3) normalization of basis vectors  $c_1, \dots, c_M$  by reducing them to unit vector;

4) calculation of coefficients of dimensions of vicinity of neighborhood and of learning rate from formulas (4) and (5);

5) initialization of a set of indices of columns  $C$ , which have already been used during  $t$ -iterations  $U = \emptyset$ ;

6) initialization of residue that is minimized,  $\varepsilon^U = x$ ;

7) initialization of temporary matrix  $R = (r_1, \dots, r_L, \dots, r_M) = C$ , orthonormalized according to  $C^U$ ;

8) initialization of the counter of steps of residue refinement  $h := 1, \quad h = \overline{1, K-1}$ ;

9) calculation of distance (scalar product) of vector  $r_{i_k}$  to  $\varepsilon^U$  and their arrangement in ascending order

$$-(r_{i_0}^T \varepsilon^U)^2 \leq \dots \leq -(r_{i_k}^T \varepsilon^U)^2 \leq \dots \leq -(r_{i_{M-h-1}}^T \varepsilon^U)^2;$$

10) initialization of the counter of steps of code book refinement

$$k = 0, k = \overline{0, M-h-1};$$

11) updating of code book words at the  $k$ -th step with the use of principles of orthogonality to the sub-space, assigned in  $C^U$  and by the Oja's rule [11]

$$c_{i_k} := c_{i_k} + \Delta_{i_k},$$

$$r_{i_k} := r_{i_k} + \Delta_{i_k},$$

where

$$\Delta_{i_k} := \alpha_t \exp(-k/\lambda_t) y (\varepsilon^U - y r_{i_k}),$$

where

$$y := r_{i_k}^T \varepsilon^U;$$

12) normalization  $r_{i_k}$  by reducing to unit vector;

13) if  $k < 0, \quad M-h-1$ , move to step 11;

14) determining of winner basis from formula (2);

15) updating of matrix  $R$  and current residue  $\varepsilon_i^U$  from formulas (1) and (3);

16) updating of matrix of selected basis vectors

$$U = U \cup I_{\text{win}};$$

17) if  $h < K-1$ , move to step 11;

18) if  $t < t_{\text{max}}$ , move to step 2, otherwise – end of processing.

The first layer of feature extractor may carry out analysis of input signals from several time windows that overlap in time. After studying the first layer of feature extractor, the whole training sample can be recoded to sparse concatenated representation and use it for learning the next layer. Prior to this, it is advisable to introduce non-linearity to the obtained representation and reduce the number of basis vectors in a new layer [11]. The simplest non-linearity is the limit in the form of a condition of non-negative features, in which the output of the  $S$ -th layer  $o_s$  with a sparse code  $a_s^{\text{OOMP}}$  can be calculated from formula

$$o_s(a_s^{\text{OOMP}}) := [\max(0, a_s^{\text{OOMP}}), \max(0, a_s^{\text{OOMP}})], \quad (6)$$

where 0 is the vector with zero components, of dimensionality  $M$ ; max is the operator of element by element maximum between two vectors.

Application of the proposed non-linearity (6) increases dimensionality of the resulting code twice  $o_s \in R^{2^M}$ , but enhances informativeness due to the possibility of separate analysis of negative and positive responses of a signal and retains scarcity property. Thus, nonzero values of feature representation of the higher-level signal about the activation of a certain group of low-level features. In this case, it is possible to submit to the classifier both the output of the last layer of feature extractor and outputs of the lower layers, which will allow carrying out classification analysis taking into account the specificity of functional state at each abstraction level.

Algorithm of rough binary encoding of the feature vector for classification of analysis involves comparing value of the  $i$ -th feature with a corresponding lower  $A_{B,i}$  and higher  $A_{T,i}$  limits of the asymmetrical field of control tolerances, which are calculated from formulas

$$A_{B,i} = y_{i,\text{max}} \left[ 1 - \frac{\delta_{i,i}}{\delta_{\text{max}}} \right],$$

$$A_{T,i} = y_{i,\text{max}} \quad \text{at } l = \overline{1, L},$$

where  $\delta_{i,i}$  is the parameter of the  $l$ -th field of control tolerances for the value of the  $i$ -th feature.

Formation of a binary learning matrix

$$\{x_{k,i}^{(j)} \mid i = \overline{1, L \cdot N}; j = \overline{1, n_k}; k = \overline{1, K}\},$$

where  $N$  is the number of features of a classifier,  $n_m$  is the number of vectors of class  $X_m^o$  and  $K$  is the number of recognition classes, performed by the rule

$$x_{k,(l-1)N+i}^{(j)} = \begin{cases} 1, & \text{if } A_{B,l,i} \leq y_{k,i}^{(j)} \leq A_{U,l,i}; \\ 0, & \text{else.} \end{cases}$$

Calculation of values of coordinates of binary averaged vector  $x_k$ , relative to which construction in radial basis of class containers takes place, is performed by the rule

$$x_{k,i} = \begin{cases} 1, & \text{if } \frac{1}{n_k} \sum_{j=1}^{n_k} x_{k,i}^{(j)} > \frac{1}{n} \sum_{k=1}^K \sum_{j=1}^{n_k} x_{k,i}^{(j)}; \\ 0, & \text{if else;} \end{cases} \quad i = \overline{1, N \cdot L},$$

where  $n$  is the total volume of labeled vectors of the initial sample.

As the criterion of efficiency of classifier's machine learning to recognize observations of class  $X_k^o$ , modification of Kullbak's information measure is considered in [14, 15]:

$$J_k = \frac{1 - (\alpha_k + \beta_k)}{\log_2(2 + \varepsilon) - \log_2 \varepsilon} \cdot \log_2 \left[ \frac{2 - (\alpha_k + \beta_k) + \varepsilon}{(\alpha_k + \beta_k) + \varepsilon} \right], \quad (7)$$

where  $\alpha_k, \beta_k$  are assessments of errors of first and second kind, which assign operation region of the criterion in the form of inequalities  $\alpha_k \geq 0,5$  and  $\beta_k \geq 0,5$ ;  $\varepsilon$  is the small sign-positive number for avoiding uncertainty while dividing by zero, equal, as a rule, to a number from range  $[10^{-4} \dots 10^{-2}]$ .

Optimization of parameters of the field of control tolerances  $\{\delta_{l,i}\}$  lies in searching for extremum of function (7) in decision hyperspace. In this case, as the search algorithm, this work proposes to use Particle Swarm Optimization, PSO, which is characterized by simplicity of implementation and interpretability [16]. Optimization of the radii of class containers can be implemented by the method of sequential direct lookup with the assigned step, because the number of steps of this search is relatively small.

To improve image compactness and inter-class gap in binary space of secondary features, the algorithm of machine learning takes into account fuzzy compactness of images that is calculated for class  $X_k^o$  from formula

$$L_k = \frac{d(x_k \oplus x_c)}{d_k + d_c}, \quad (8)$$

where  $d_k, d_c$  are the radii of class container  $X_k^o$  and the closest neighboring class  $X_c^o$  relatively;  $d(x_k \oplus x_c)$  is the code distance between the centers of class containers and  $X_c^o$ , which is calculated from formula

$$d(x_k \oplus x_c) = \sum_{i=1}^N (x_{k,i} \oplus x_{c,i}).$$

Effectiveness of each particle of the population-based algorithm, i. e. closeness to the global optimum, is measured with the help of pre-determined fitness function, the role of which in this case is performed by the function of criterion of machine learning efficiency (7). Each  $j$ -th particle, apart from its position  $P_j$  retains the following information:  $V_j$  is the current velocity of a particle,  $Pbest_j$  is the best personal position of a particle. The best personal position of the  $j$ -th particle is the position of the  $j$ -th particle, in which the value of fitness functions for the particle was maximum at the current point of time. In addition, with the aim of searching for the global extremum of fitness function, the best particle is sought for throughout the whole swarm and the position is designated as  $Gbest$ .

However, considered above swarm search algorithm is aimed at increasing the value of criterion of learning effectiveness, averaged by the class alphabet. For the purpose of additional increase in compactness of images, it is necessary to modify the procedure of updating the values of the best personal  $Pbest_j$  position of search agents by rule (9), in which objective function  $E(\dots)$  is the averaged value of function of criterion (7).

$$\begin{aligned} & \text{if } |E(P_j) - E(Pbest_j)| < \varepsilon \\ & \text{and } \bar{L}(P_j) > \bar{L}(Pbest_j), \text{ then } Pbest_j := P_j. \end{aligned} \quad (9)$$

Similarly, it is necessary to modify the procedure to updating the values of the best global  $Gbest_j$  position of search agents

$$\begin{aligned} & \text{if } |E(Pbest_j) - E(Gbest)| < \varepsilon \\ & \text{and } \bar{L}(Pbest_j) > \bar{L}(Gbest), \text{ then } Gbest := Pbest_j. \end{aligned} \quad (10)$$

Under the mode of examination, decision on belonging of vector-implementation  $x$  to one alphabet class  $\{X_k^o\}$  is made by calculation of geometrical membership function

$$\mu_k^*(x) = \max_{\{k\}} \{\mu_k(x)\},$$

In which  $\mu_k(x)$  is the membership function of vector  $x$  to the container of class  $X_k^o$ , which is calculated by the rule:

$$\mu_k(x) = 1 - \frac{d(x_k^* \oplus x)}{d_k^*}.$$

For more precise consideration of distribution of binary vectors in the hyperspheric container of class  $X_k^o$ , formula of membership function can be adjusted and will take the form

$$\mu_k(x) = \begin{cases} 1 - \frac{d(x_k^* \oplus x)}{d_k^*}, & \text{if } d(x_k^* \oplus x) > d_k^*; \\ \frac{n_k(d)}{n_{\max}}, & \text{else,} \end{cases} \quad (11)$$

where  $n_k(d)$  is the number of vectors of class  $X_k^o$ , which is at the distance  $d$  from the center  $x_k$ ;  $n_{\max}$  is the maximum value in array  $n_k(d)$ , i. e.,

$$n_{\max} = \max_d \{n_k(d)\}.$$

Thus, the proposed algorithms of feature learning and decision rules for prediction of conditions of SLA violations are not demanding to the amount of data and resources of the computer, which provides effectiveness of resource management at the early stages of service operation.

## 5. Results of physical simulation of the system of prediction of violation of SLA conditions

Testing effectiveness of the proposed algorithms is considered on the example of the problem of prediction of data center servers' overloading, which leads to SLA violation in accessibility metrics, resource capacity and response time. The simulation was carried out with the use of framework Clouds [16], where there were assigned 400 servers HP ProLiant ML110 G4 (Intel Xeon 3040, 2 cores×1860 MHz, 4 GB) and 400 servers HP ProLiant ML110 G5 (Intel Xeon



3075, 2 cores×2660 MHz, 4 GB). Workload data, collected on PlanetLab platform, were taken from project CoMon [16]. The forecast horizon is 10 minutes, which is sufficient for implementation of migration of a virtual machine. The architecture of the system of prediction of SLA violations is shown in Fig. 1 and includes the two-level feature extractor. The extractor analyzes data of monitoring of loading of processing resource of the virtual machine in two sub-windows, shifted in time, with 50 % overlap and the reading step equal to 1 minute. The length of the sub-window exceeds by several times the prediction horizon and is 50 minutes, which was chosen at our discretion and may be not optimal.

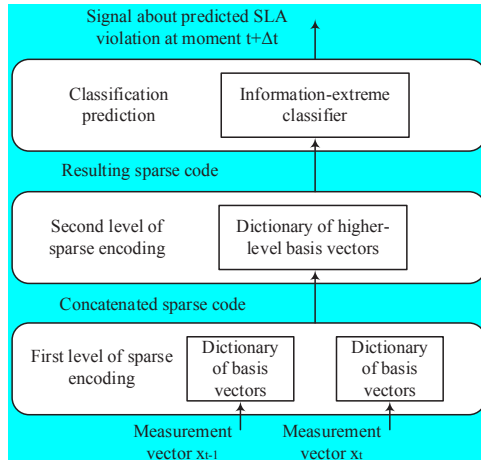


Fig. 1. Block diagram of system of classification prediction of violation of SLA terms

The selection of unlabeled samples for learning of the two-level feature extractor includes 10000 samples, and the volume of a priori classified learning sample of each of the two classes is 100 samples. The test sample of the classifier has the same volume as the learning sample. Table 1 shows results of machine learning at different capacity of dictionary of basis vectors of the first and second levels.

Table 1

Results of machine learning of classifier at various configurations of feature extractor

No. by order	Capacity of dictionary of basis vectors of the first level	Capacity of dictionary of basis vectors of the second level	Values of averaged information criterion (7)	Accuracy of a classifier by test sample
1	20	5	0.112	0.8
2	20	10	0.118	0.81
3	20	15	0.118	0.82
4	30	10	0.251	0.91
5	30	15	0.751	0.99
6	30	20	1.000	1.00
7	40	15	1.000	1.00
8	40	20	1.000	1.00
9	40	25	1.000	1.00

An analysis of Table 1 shows that the best of the checked configurations of feature extractor is the sixth configuration that provides error-free decision rules for the test sample with minimal number of basis vectors. Table 2 shows results

of machine learning of the classifier with the sixth configuration of feature extractor at varying number of control tolerances for recognition features.

Table 2

Results of machine learning of classifier with varying number of control tolerances for values of high-level features

No. by order	The number of control tolerances for features	Value of averaged information criterion	Accuracy of a classifier by test sample
1	1	0.51	0.98
2	2	0.75	0.99
3	3	1.000	1.00
4	4	1.000	1.00
5	5	1.000	1.00
6	6	1.000	1.00
7	7	1.000	0.99

An analysis of Table 2 shows that the optimum number of control tolerances for values of features is  $L=3$  and subsequent increase in the number of tolerances can lead to overfitting, which is evident from Table at  $L=7$ . In this case, Fig. 2 shows diagrams of change in accuracy of obtained decision rules by learning and test samples from the number of learning vectors of feature extractor.

An analysis of Fig. 2 shows that an increase in the number of the extractor's learning vectors leads to improving accuracy for learning and test samples for the classifier of functional states of the service. However, at the volume of learning sample of about 5,000 samples, the effect of overfitting of the system was observed after reaching 6,100 samples, it is possible to obtain the extractor that provides error-free decision rules for the test sample.

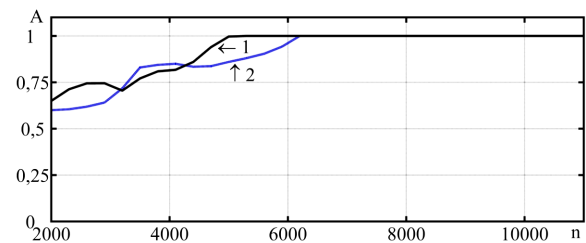


Fig. 2. Charts of dependence of effectiveness of decision rules on the number of learning vectors of feature extractor:

- 1 – the curve of accuracy change for learning sample;
- 2 – the curve of accuracy change for test sample

Thus, the developed algorithm of learning features and decision rules allows us to obtain error-free decision rules for test sample with the extractor, containing 30 basis vectors in the first layer and 20 vectors in the second layer. In this case, 6,100 learning samples are sufficient for learning of the extractor.

## 6. Discussion of results of physical simulation of machine learning process

The use of the proposed extractor and modification by rules (9) and (10) swarm algorithm of decision rules optimization, as shown in Fig. 2, makes it possible to obtain highly reliable decision rules. In this case, the diagram shows the overfitting section with the width of 1,100 samples, at the

end of which the accuracy for the test sample reaches the limit maximum value. Effect of overfitting has a component both of the extractor and the classifier. To assess the impact of the used rules (9) and (10) on the effect of overfitting, Fig. 3 shows diagrams of change in accuracy of obtained decision rules for learning and test samples depending on the number of learning vectors of the extractor without using these rules.

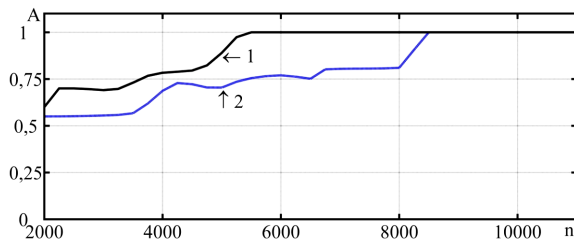


Fig. 3. Charts of dependence of effectiveness of decision rules on the number of learning vectors of feature extractor without using rules (9) and (10): 1 – the curve of accuracy change for learning sample; 2 – the curve of accuracy change for test sample

As Fig. 3 shows, without considering compactness of images by rules (9) and (10), in order to obtain highly reliable decision rules, it is necessary to use learning sample of much larger volume, which in this case includes 8,500 samples.

To compare generalizing ability of the proposed extractor with the popular extractor based on the deep autoencoder [9], Fig. 4 shows diagrams of change in accuracy of obtained decision rules for learning and test samples depending on the number of learning vectors of the autoencoder. In this case, the autoencoder has the following configuration: input dimensionality is 75 features; the number of nodes of the first hidden layer is 30; the number of nodes in the hidden layer that corresponds to feature representation is 20.

An analysis of Fig. 4 shows that deep autoencoder similarly allows obtaining of error-free decision rules for the test sample, but to do this, we need more training samples, the number of which exceeds 10,000.

Thus, the developed informational and algorithmic software make it possible to obtain highly reliable decision rules for the prediction of violation of SLA conditions. In this case, the implemented algorithms, compared with the autoencoder, require a smaller volume of learning data, which allows

previous introduction of predictive mechanisms of management of correspondent services.

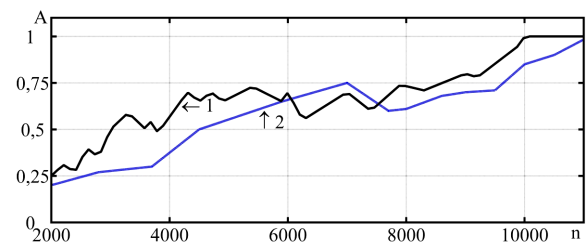


Fig. 4. Charts of dependence of effectiveness of decision rules on the number of learning vectors of autoencoder: 1 – the curve of accuracy change for learning sample; 2 – the curve of accuracy change for test sample

## 7. Conclusions

1. Results of physical simulation prove capability of both the proposed hierarchical feature extractor, based on ideas and methods of neural gas and sparse encoding, and of the autoencoder to obtain error-free decision rules for learning and test samples. However, the proposed extractor, unlike the autoencoder, requires 1.6 times smaller volume of learning samples for achievement of the same result, which makes it possible preliminarily to put in effect predictive mechanisms of management of appropriate cloud services.

2. It is shown that consideration of image compactness in binary space of secondary features during optimization of multi-level system of control tolerances for values of primary features allows us to significantly reduce the negative effect of overfitting of a classifier and requirements for the volume of learning samples.

3. It was shown that the proposed configuration of the extractor for the problem of prediction of violation of SLA condition is acceptable in terms of accuracy and complexity. In this case, at the input of the extractor, two time windows are used that intersect in time by 50 % and read through 50 features. The first layer of coding of the extractor contains 30 basis vectors, and the second layer – 20. The intralayer pooling and non-linearity were formed by concatenation of sparse codes of each of the windows and by continuation of resulting code twice as much in order to separate positive and negative code components and by transforming the resulting code into the vector of sign-positive features.

## References

1. Reyhane, A. H. SLA Violation Prediction In Cloud Computing: A Machine Learning Perspective [Electronic resource] / A. H. Reyhane, H. Abdelhakim // arXiv. – 2016. – Available at: <https://arxiv.org/pdf/1611.10338.pdf>
2. Minarolli, D. Tackling uncertainty in long-term predictions for host overload and underload detection in cloud computing [Text] / D. Minarolli, A. Mazrekaj, B. Freisleben // Journal of Cloud Computing. – 2017. – Vol. 6, Issue 1. doi: 10.1186/s13677-017-0074-3
3. Wajahat, M. Using machine learning for black-box autoscaling [Text] / M. Wajahat, A. Gandhi, A. Karve, A. Kochut // 2016 Seventh International Green and Sustainable Computing Conference (IGSC). – 2016. doi: 10.1109/igcc.2016.7892598
4. Meskini, A. Proactive Learning from SLA Violation in Cloud Service based Application [Text] / A. Meskini, Y. Taher, A. El gammal, B. Finance, Y. Slimani // Proceedings of the 6th International Conference on Cloud Computing and Services Science. – 2016. doi: 10.5220/0005807801860193
5. Ashraf, A. Automatic Cloud Resource Scaling Algorithm based on Long Short-Term Memory Recurrent Neural Network [Text] / A. Ashraf // International Journal of Advanced Computer Science and Applications. – 2016. – Vol. 7, Issue 12. doi: 10.14569/ijacsa.2016.071236

6. Gupta, L. Fault and Performance Management in Multi-Cloud Based NFV using Shallow and Deep Predictive Structures [Text] / L. Gupta, M. Samaka, R. Jain, A. Erbad, D. Bhamare, H. A. Chan // 7th Workshop on Industrial Internet of Things Communication Networks at The 26th International Conference on Computer Communications and Networks (ICCCN 2017). – Vancluver, 2017.
7. Tarsa, S. J. Workload prediction for adaptive power scaling using deep learning [Text] / S. J. Tarsa, A. P. Kumar, H. T. Kung // 2014 IEEE International Conference on IC Design & Technology. – 2014. doi: 10.1109/icidct.2014.6838580
8. Flenner, J. A Deep Non-Negative Matrix Factorization Neural Network [Electronic resource] / J. Flenner, B. Hunter // Available at: <http://www1.cmc.edu/pages/faculty/BHunter/papers/deepNMF.pdf>
9. Li, Y. Learning-based power prediction for data centre operations via deep neural networks [Text] / Y. Li, H. Hu, Y. Wen, J. Zhang // Proceedings of the 5th International Workshop on Energy Efficient Data Centres – E2DC '16. – 2016. doi: 10.1145/2940679.2940685
10. Zhao, Z. Stacked Multilayer Self-Organizing Map for Background Modeling [Text] / Z. Zhao, X. Zhang, Y. Fang // IEEE Transactions on Image Processing. – 2015. – Vol. 24, Issue 9. – P. 2841–2850. doi: 10.1109/tip.2015.2427519
11. Chan, T.-H. PCANet: A Simple Deep Learning Baseline for Image Classification [Electronic resource] / T.-H. Chan, K. Jia, S. Gao, J. Lu et. al. // arXiv. – 2014. – Available at: <https://arxiv.org/pdf/1404.3606.pdf>
12. Labusch, K. Learning Data Representations with Sparse Coding Neural Gas [Text] / K. Labusch, E. Barth, T. Martinetz // Proceedings of the European Symposium on Artificial Neural Networks. – Bruges, 2008. – P. 233–238.
13. Labusch, K. Sparse Coding Neural Gas: Learning of overcomplete data representations [Text] / K. Labusch, E. Barth, T. Martinetz // Neurocomputing. – 2009. – Vol. 72, Issue 7-9. – P. 1547–1555. doi: 10.1016/j.neucom.2008.11.027
14. Moskalenko, V. Optimizing the parameters of functioning of the system of management of data center it infrastructure [Text] / V. Moskalenko, S. Pimonenko // Eastern-European Journal of Enterprise Technologies. – 2016. – Vol. 5, Issue 2 (83). – P. 21–29. doi: 10.15587/1729-4061.2016.79231
15. Dovbysh, A. S. Information-Extreme Method for Classification of Observations with Categorical Attributes [Text] / A. S. Dovbysh, V. V. Moskalenko, A. S. Rizhova // Cybernetics and Systems Analysis. – 2016. – Vol. 52, Issue 2. – P. 224–231. doi: 10.1007/s10559-016-9818-1
16. Mosa, A. Optimizing virtual machine placement for energy and SLA in clouds using utility functions [Text] / A. Mosa, N. W. Paton // Journal of Cloud Computing. – 2016. – Vol. 5, Issue 1. doi: 10.1186/s13677-016-0067-7

*Запропоновано метод оптимізації класифікаційних нечітких баз знань з використанням полішувальних підстановок у вигляді розв'язків нечітких реляційних рівнянь. Полішувальні підстановки дозволяють формалізувати процес генерування та відбору варіантів нечіткої бази знань за критеріями «точність – складність», що спрощує процес налаштування*

*Ключові слова: оптимізація нечітких баз знань, міні-тах кластеризація, розв'язання нечітких реляційних рівнянь*

*Предложен метод оптимизации классификационных нечетких баз знаний с использованием улучшающих подстановок в виде решений нечетких реляционных уравнений. Улучшающие подстановки позволяют формализовать процесс генерирования и отбора вариантов нечеткой базы знаний по критериям «точность – сложность», что упрощает процесс настройки*

*Ключевые слова: оптимизация нечетких баз знаний, мин-тах кластеризация, решение нечетких реляционных уравнений*

UDC 681.5.015:007

DOI: 10.15587/1729-4061.2017.110261

# OPTIMIZATION OF FUZZY CLASSIFICATION KNOWLEDGE BASES USING IMPROVING TRANSFORMATIONS

H. Rakytyanska

PhD, Associate Professor

Department of software design

Vinnytsia National Technical University

Khmelnyske shose str., 95,

Vinnytsia, Ukraine, 21021

E-mail: h\_rakit@ukr.net

## 1. Introduction

Fuzzy classification knowledge base design is carried out according to the criteria of accuracy, complexity, and interpretability. The design criteria are provided by gradual transformations of the initial model. In the theory of

defect-free design of human-machine systems [1, 2], formalization of such transformations is achieved by the use of improving transformations.

Then improving transformations correspond to the addition (removal) of output classes, input terms, and rules. Improving transformations allow formalization of the process