

# Fuzzy Clustering Data Arrays with Omitted Observations

**Zhengbing Hu**

School of Educational Information Technology, Central China Normal University, Wuhan, China  
 E-mail: hzb@mail.ccnu.edu.cn

**Yevgeniy V. Bodyanskiy**

Kharkiv National University of Radio Electronics, Kharkiv, Ukraine,  
 E-mail: yevgeniy.bodyanskiy@nure.ua

**Oleksii K. Tyshchenko and Vitalii M. Tkachov**

Kharkiv National University of Radio Electronics, Kharkiv, Ukraine,  
 E-mail: lehatish@gmail.com, vitalii@tkachov.com

**Abstract**—An adaptive neural system which solves a problem of clustering data with missing values in an online mode with a permanent correction of restorable table elements and clusters' centroids is proposed in this article. The introduced neural system is characterized by both a high speed and a simple numerical implementation. It can process information in a real-time mode.

**Index Terms**—Computational Intelligence, Machine Learning, missing values, gaps' recovery, adaptive system, fuzzy clustering.

## I. INTRODUCTION

A clustering task is one of the most important problems in Data Mining [1-9]. There is a variety of different approaches and algorithms (from purely intuitive and heuristic methods to rigorously mathematical ones [10-20]) used for solving the problem. With that in many Data Mining tasks, initial data tables may contain empty cells (gaps, outliers, and omitted values) where information is missing for some reasons. Some attention was paid to a task of these missing values recovery [21-23]. The most effective approaches for this specific case were methods based on Computational Intelligence (mathematical tools of soft computing) [24-30] and primarily artificial neural networks [31-36]. Besides that, some widely known approaches to gaps' recovery and traditional clustering algorithms can be applied only in instances when an initial data table is given a priori and a number of its rows and columns can't be changed during data processing. Simultaneously, there's a wide enough group of tasks when data come to processing sequentially in an online mode while it is unknown beforehand which vector to be processed may contain missing values. These processes of data recovery and their clustering should be run simultaneously in a real-time mode.

The remainder of this paper is organized as follows: Section 2 describes an adaptive linear element. Section 3

declares description of the problem. Section 4 describes an adaptive algorithm for filling out gaps. Section 5 describes adaptive restoration of data with omitted values based on an ensemble of neo-fuzzy neurons. Section 6 declares fuzzy clustering data with omitted values. Conclusions and future work are given in the final section.

## II. AN ADAPTIVE LINEAR ELEMENT (ADALINE)

One of the simplest neurons which are capable of learning is an adaptive linear element (adaline). It was proposed by B. Widrow and it is shown in Fig.1.

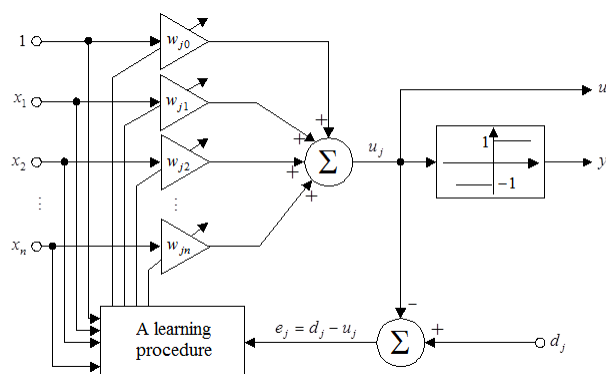


Fig.1. An Adaptive Linear Element

The adaline is structurally similar to a neuron by McCullough-Pitts with an activation signum function and comprises two main parts: an adaptive linear associator and a nonlinear activation function. The adaline has  $n+1$  inputs  $x_0, x_1, \dots, x_n$  and two outputs – an analog one  $u_j$  and a binary one  $y_j$ . Additionally, there's an additional input, where a learning signal  $d_j$  is fed. This signal demonstrates what a desired response for a neuron should be to every specific set of input signals. The analog output  $u_j$  is a weighted sum of input values  $x_j$

$$u_j = \sum_{i=0}^n w_{ji} x_i = w_j^T x, \quad (1)$$

and a binary output  $y_j$  may take on values either +1 or -1 depending on polarity of the analog signal  $u_j$ . The output signal  $u_j$  is compared to the external reference signal  $d_j$  and an error signal  $e_j = d_j - u_j$ , which occurs in this case, comes to a learning procedure, which transforms synaptic weights in such a way that some error function  $e_j$  is minimized (also known as a learning criterion). A quadratic function is usually used as this function that makes it possible to use not only a native learning algorithm (developed by B. Widrow and M. Hoff especially for the adaline) but also other recurrent procedures for adaptive identification.

The adaline may be used both as an elementary neuron in a neural network and for solving pattern recognition problems, signal processing problems, and for implementation of logical functions.

In our case, the adalines are used for processing data which come in a sequential manner.

### III. DESCRIPTION OF THE PROBLEM

Let's suppose that an  $(N \times n)$ -table,  $N \geq 1, n \geq 1$  is given that describes «object-property» characteristics.

Table 1. An Example of a Data Table with Missing Data

	$1$	$\dots$	$p$	$\dots$	$j$	$\dots$	$n$
$1$	$x_{11}$	$\dots$	$x_{1p}$	$\dots$	$x_{1j}$	$\dots$	$x_{1n}$
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
$i$	$x_{i1}$	$\dots$	$x_{ip}$	$\dots$	$x_{ij}$	$\dots$	$x_{in}$
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
$k$	$x_{k1}$	$\dots$	$x_{kp}$	$\dots$	$x_{kj}$	$\dots$	$x_{kn}$
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
$N$	$x_{N1}$	$\dots$	$x_{Np}$	$\dots$	$x_{Nj}$	$\dots$	$x_{Nn}$
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$

Table 1 contains information about  $N$  objects, where each one is described as an  $(1 \times n)$ -row vector of features  $\underline{x}_i = (x_{i1}, \dots, x_{ip}, \dots, x_{ij}, \dots, x_{in})$ . It is assumed here that  $N_G$  rows may have a single gap and  $N_F = N - N_G$  rows filled completely. While processing a table, it is necessary to fill out gaps and form  $m$  clusters. Thus, it's required that recovered elements are in some sense «the most plausible ones» or «close enough» to previously unknown regularities, which are contained in the table and can be changed during data processing in some unknown manner.

### IV. AN ADAPTIVE ALGORITHM FOR FILLING OUT GAPS

First of all, let's represent the Table 1 in the form of an

$(N \times n)$ -matrix  $X$  where either an element  $x_{kj}$  is missing or  $N_G$  elements are missing in a more general case. It's supposed that there's a linear correlation between columns  $\vec{x}_j = (x_{1j}, \dots, x_{ij}, \dots, x_{kj}, \dots, x_{Nj})^T$  based on which a gap recovery is performed with the help of an equation of the linear regression

$$\hat{x}_{kj} = w_{j0} + w_{j1}x_{k1} + w_{j2}x_{k2} + \dots + w_{jn}x_{kn} \quad (2)$$

or

$$\hat{x}_{kj} = \underline{X}_{kj} w_j \quad (3)$$

where  $w_j = (w_{j0}, w_{j1}, \dots, w_{jn})^T$  is a  $(n+1)$ -column vector of features to be determined,  $\underline{X}_{kj} = (1, x_{k1}, \dots, x_{k,j-1}, x_{k,j+1}, \dots, x_{kn})$  is a  $(1 \times n)$ -row vector of features for the  $k$ -th object without an element  $x_{kj}$  and having 1 at the first position.

A vector of unknown parameters  $w_j$  may be found with the help of the standard least squares method. For this reason, the  $k$ -th row should be excluded from the matrix  $X$  as well as the  $j$ -th column. A column formed by 1's should be added from the left side. Based on the obtained  $((N-1) \times n)$ -matrix  $X_j$ , parameters' estimates are calculated

$$w_j = (X_j^T X_j)^+ X_j^T \vec{X}_j \quad (4)$$

where  $\vec{X}_j = (x_{1j}, \dots, x_{ij}, \dots, x_{k-1,j}, x_{k+1,j}, \dots, x_{Nj})^T$ ,  $(\bullet)^+$  is a symbol of pseudoinversion by Moore-Penrose [37].

If there are gaps in  $N_G$  rows and in different columns, all these rows are excluded from the matrix  $X$ . Based on a truncated  $(N_F \times n)$ -matrix, vectors of parameters  $w_j$  (3) are calculated  $n$  times for all  $j=1, 2, \dots, n$ . Then all the gaps are filled out by obtained estimates  $\hat{x}_{kj}$  by means of the equations (2) and (4).

The proposed algorithm can be presented as a scheme.

- Step 1.** An initial  $(N \times n)$ -table (a matrix  $X$ ).
- Step 2.** Exclude  $N_G$  rows with missing values and form  $n$  matrices  $X_j$ .
- Step 3.** Calculate  $w_j$  vectors for all columns  $j=1, 2, \dots, n$ .
- Step 4.** Calculate  $\hat{x}_{kj}$  estimates for all rows  $k=1, 2, \dots, N$ .
- Step 5.** Restore excluded rows.
- Step 6.** A recovered  $(N \times n)$ -table (a matrix  $\hat{X}$ ).

This algorithm may be extended to a case when data

about objects come to Table 1 sequentially (an object by an object).

Let's suppose that information is given in the form of Table 1 that contains  $N$  rows, and  $N_F$  rows are filled out completely. Based on this information, an estimate for recovering the initial table can be performed like

$$w_j(N_F) = (X_j^T(N_F)X_j(N_F))^{-1} X_j^T(N_F)\bar{X}_j(N_F). \quad (5)$$

If data come sequentially (a row by a row), a recurrent version of the least squares method may be used instead of the estimate (3). Using the recurrent version of the method implies time invariability of all links that exist in a data matrix. The exponentially weighted recurrent least squares method can be unstable at low values of a smoothing parameter.

In this regard for online data analysis, it seems appropriate to use adaptive learning algorithms, which possess both tracking and filtering properties [38, 39] (for non-stationary cases). If  $N$  rows were analyzed and recovered during processing data arrays, then when the  $(N+1)$ -th observation arrives (in the form of a completely filled out row  $\underline{x}_{N+1}$ ), estimates  $w_j$  are being specified with the help of an adaptive recurrent procedure

$$\begin{cases} w_j(N_F+1) = w_j(N_F) + \\ \frac{P_j(N_F)(x_{N+1,j} - \underline{X}_{N+1,j}w_j(N_F))}{1 + \underline{X}_{N+1,j}P_j(N_F)\underline{X}_{N+1,j}^T} \underline{X}_{N+1,j}^T, \quad (6) \\ P_j(N_F+1) = P_j(N_F) - \frac{P_j(N_F)\underline{X}_{N+1,j}^T\underline{X}_{N+1,j}P_j(N_F)}{1 + \underline{X}_{N+1,j}P_j(N_F)\underline{X}_{N+1,j}^T}. \end{cases}$$

Then renewed values  $\hat{x}_{kj}$  are calculated again.

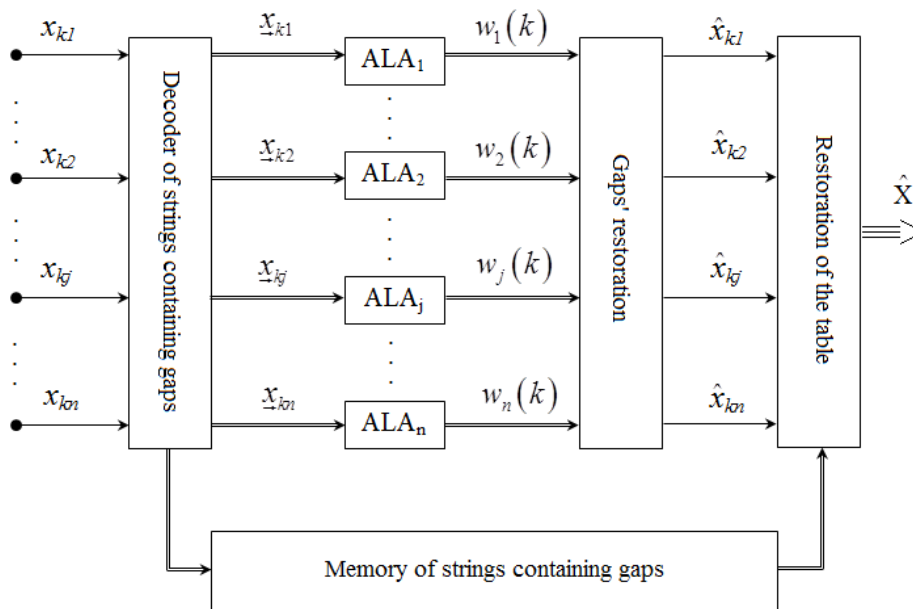


Fig.2. An Adaptive Neural System for Omitted Values' Recovery

If the  $(N+1)$ -th row has a gap, it will be skipped and the algorithm (5) will wait for a completely filled out row, for example,  $\underline{x}_{N+2}$ . After that,  $w_j(N_F+1)$  is calculated with the help of the  $(N+2)$ -th observation, and all the values  $\hat{x}_{kj}$  are corrected (including  $\hat{x}_{N+1,j}$ ).

At the initial stages of processing Table 1, an amount of completely filled out rows  $N_F$  is insufficient in comparison with a number of columns  $n$ , the estimates obtained by means of the least squares method are characterized by a low accuracy. Applying adaptive learning algorithms [38, 39] can make sequential processing more efficient. The proposed algorithm may look like

$$\begin{cases} w_j(N_F+1) = w_j(N_F) + \\ r_j^{-1}(N_F+1)(x_{N+1,j} - \underline{X}_{N+1,j}w_j(N_F))\underline{X}_{N+1,j}^T, \quad (7) \\ r_j(N_F+1) = \alpha r_j(N_F) + \|\underline{X}_{N+1,j}\|^2 \end{cases}$$

where  $0 \leq \alpha \leq 1$  is a smoothing parameter which specifies a compromise between processes of observations' filtration and tracking altering data characteristics.

That's really convenient to organize information processing in a sequential mode by means of a system based on a neural network. Its basic elements are  $n$  adaptive linear associators (adalines) [33, 34] that work in a parallel manner and can be tuned with the help of the expressions (6) and (7). A scheme of the proposed system can be found at Fig.2. An index  $k$  here stands for an object's number whose characteristics are currently being processed.

Besides that, if the system receives an input vector  $\tilde{x}_k = (x_{k1}, x_{k2}, \dots, x_{kin})$  (either full or containing some gaps) at the  $k$ -th time period, one can find exactly the same vector of observations at the system's output if  $\tilde{x}_k$  is full and its estimate  $\hat{x}_k$  in case it contains missing values.

V. ADAPTIVE RESTORATION OF DATA WITH OMITTED VALUES BASED ON AN ENSEMBLE OF NEO-FUZZY NEURONS

The problem of data recovery with omitted values based on an ensemble of the adalines considered in the previous section was solved by assuming that there's a linear correlation between columns  $\tilde{x}_j$  of the "object-property" matrix. Parameters of this correlation may be estimated either with the help of the traditional least squares method (4), or its recurrent modification (6), or the adaptive learning algorithm (7).

The case is getting more complicated when a dependence between columns  $\tilde{x}_j$  can't be adequately described by linear relations, and it has a more complex nonlinear character, and the nature of this nonlinearity is previously unknown. In a general case, an arbitrary type of nonlinearity can be arbitrarily accurately reconstructed by means of neural networks [31-36]. However, firstly, artificial neural networks require sufficiently large volumes of training samples for their learning procedures

(albeit with gaps [21-23]); secondly, implementation of the recovery system gets dramatically complicated and the system's speed reduces (and this network can't operate in an online mode).

This difficulty may be overcome with the help of new building blocks – neo-fuzzy neurons [40-43], which are neuro-fuzzy systems with high approximating properties (on the one hand), and, on the other hand, they can be considered nonlinear extensions of the adalines. That's why linear learning algorithms (optimal in speed) can be used for their training.

An architecture of the neo-fuzzy neuron is shown in Fig.3. In addition to the adjustable synaptic weights  $w_{li}$ , it also contains nonlinear membership functions  $\mu_{li}(x_i)$  that provide necessary approximating properties. After a vector signal  $\tilde{x}_k = (\tilde{x}_{k1}, \tilde{x}_{k2}, \dots, \tilde{x}_{kn})^T$  (here the input data are previously configured so that  $0 \leq \tilde{x}_{ki} \leq 1$  for all  $1 \leq k \leq N$  and  $1 \leq i \leq n$ ) is fed to the neo-fuzzy neuron's input, a scalar value appears at its output

$$y_k = \sum_{i=1}^n f_i(\tilde{x}_{ki}) = \sum_{i=1}^n \sum_{l=1}^h w_{li}(k-1) \mu_{li}(\tilde{x}_{ki}), \quad (8)$$

which can be defined by adjustable synaptic weights  $w_{li}(k-1)$  (similar to the adaline) and by membership functions  $\mu_{li}(\tilde{x}_{ki})$ .

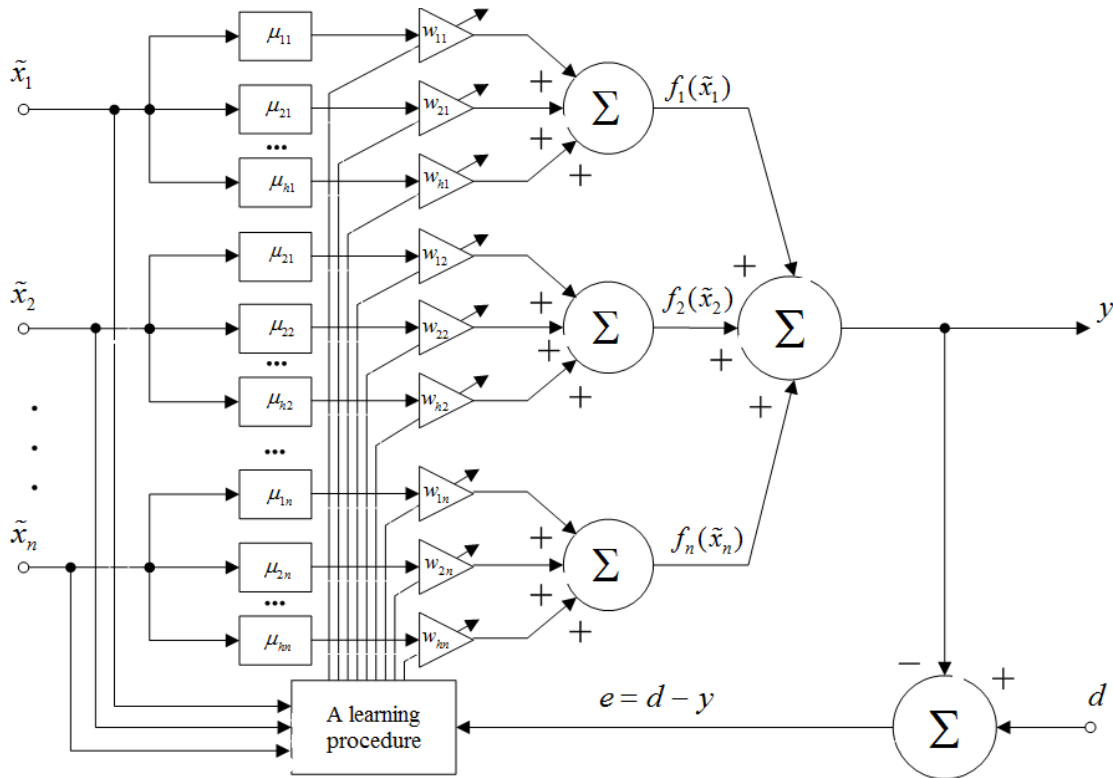


Fig.3. An Architecture of the Neo-Fuzzy Neuron

As a learning criterion for the neo-fuzzy neuron, the quadratic function (similar to the adaline's one) is used

$$E(k) = \frac{1}{2}(d_k - y_k)^2 = \frac{1}{2}e_k^2 = \frac{1}{2}\left(d_k - \sum_{i=1}^n \sum_{l=1}^h w_{li}(k-1)\mu_{li}(\tilde{x}_{ki})\right)^2, \quad (9)$$

which is minimized with the help of a gradient algorithm

$$w_{li}(k+1) = w_{li}(k) + \eta e_k \mu_{li}(\tilde{x}_{ki}) \quad (10)$$

where  $\eta$  is a scalar parameter of a learning rate.

Introducing  $(hn \times 1)$ - vectors of membership functions

$$\mu(\tilde{x}_k) = (\mu_{11}(\tilde{x}_{k1}), \mu_{21}(\tilde{x}_{k1}), \dots, \mu_{h1}(\tilde{x}_{k1}), \mu_{21}(\tilde{x}_{k2}), \dots, \mu_{hm}(\tilde{x}_{kn}))^T$$

and synaptic weights  $w = (w_{11}, w_{21}, \dots, w_{h1}, w_{12}, \dots, w_{hm})^T$ , the equation (8) can be written down in a vector form (11), which owns a similar structure to the image (1) implemented by the adaline.

$$y_k = w^T(k-1)\mu(\tilde{x}_k), \quad (11)$$

It is easy to note that evaluation of the vector of synaptic weights may be carried out both by the standard least squares method (if the Table 1 is given beforehand) and by means of the algorithm (7) for a sequential mode,

which in this case takes on the form

$$\begin{cases} w(k+1) = w(k) + r^{-1}(k+1)(d_{k+1} - w^T(k)\mu(\tilde{x}_{k+1}))\mu(\tilde{x}_{k+1}), \\ r(k+1) = \alpha r(k) + \|\mu(\tilde{x}_{k+1})\|^2 \end{cases} \quad (12)$$

for an individual neo-fuzzy neuron; and

$$\begin{cases} w_j(N_F + 1) = w_j(N_F) + \\ r^{-1}(N_F + 1)(\tilde{x}_{N+1,j} - \mu(\tilde{x}_{N+1,j})w_j(N_F))\mu^T(\tilde{x}_{N+1,j}), \\ r_j(N_F + 1) = \alpha r(N_F) + \|\mu(\tilde{x}_{N+1,j})\|^2 \end{cases} \quad (13)$$

for the neo-fuzzy neuron in the adaptive neuro-fuzzy system for gaps' recovery (Fig.4).

On the truth of speaking, this system is an extension of the adaptive system from Fig.2 except the fact that the dadlines ( $ALA_j$ ) were substituted for the nonlinear neo-fuzzy neurons ( $NFN_j$ ).

As it can be seen, the nonlinear expansion of the problem from the theoretical point of view does not cause any problems, and a sacrifice for the nonlinear properties is a  $h$  times more number of tunable synaptic weights in comparison with the system in Fig.2. This feature could lead to a decrease of data processing speed in a number of cases.

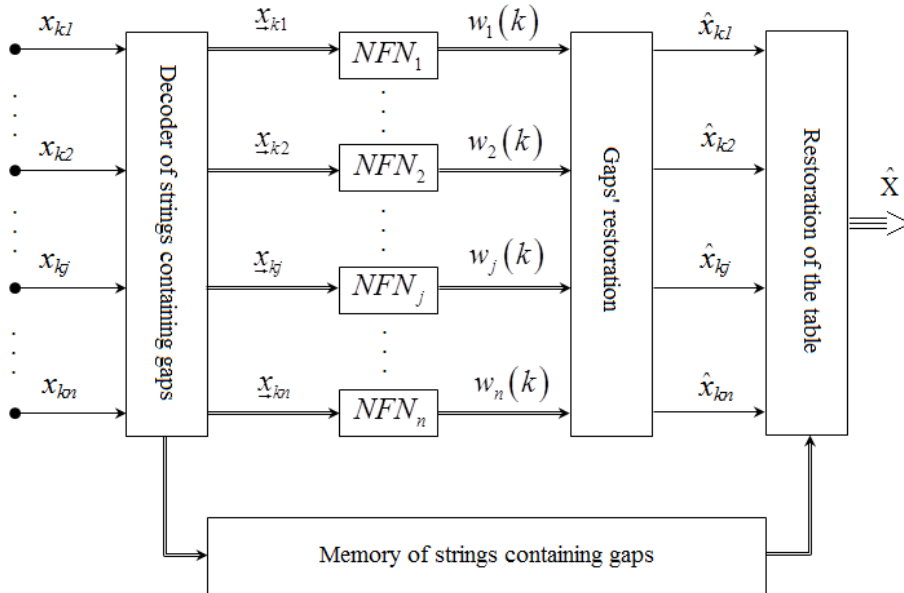


Fig.4. An Adaptive Neuro-Fuzzy System for Omitted Values' Recovery

## VI. FUZZY CLUSTERING DATA WITH OMITTED VALUES (AN EXPERIMENTAL RESEARCH)

Three fuzzy clustering algorithms (FCM, algorithm by Gustafson-Kessel, algorithm by Gath-Geva) were used for quality assessment of data containing missing values. We used the UCI repository data (the Iris dataset, the

Wine dataset, and the Breast-Cancer-Wisconsin data) to carry out experiments.

The Wine dataset contains results of a chemical analysis for wines from three varieties of grape which grow in one region of Italy. The data sample consists of 178 observations and 13 attributes per each sample. Attributes describe main chemical indicators of wine

analysis such as an amount of alcohol in wine, a malic acid, intensity of a wine’s color, its shade and so on.

It should be noted that information we use is full and contains no gaps. This fact makes it possible to check an algorithm for incoming data in a real-time mode. Initially, the data are loaded into the system without being distorted; later gaps are artificially inserted into the table; and calculation of missing elements and an error of obtained results is carried out. The chosen approach is convenient to perform a comparison of the obtained calculations for missing elements and originally presented ones. Then rows and columns which contain gaps are deleted from the table. If there are some gaps in  $N_G$  rows and in different columns, all these rows are excluded from the matrix  $X$ . And on the basis of the obtained truncated  $(N_F \times n)$ -matrix, parameters’ vectors (3) can be found  $n$  times for all  $j = 1, 2, \dots, n$ . As a result of the application’s functioning, an errors’ table is obtained, according to which it is possible to plot a graph for changing errors when new instances are being processed. As it can be seen from Fig.6, an error for 169 samples with 10 gaps is gradually decreasing. It may be concluded that it’s low enough. Initially, due to a small number of samples provided, the error was higher than the mean error, but it was decreasing and stabilizing during the analysis of remaining samples.

As criteria of clustering quality assessment, we used Partition Coefficient (PC), Classification Entropy (CE), Partition Index (SC), Separation Index (S), Xie-Beni’s Index (XB), and Dunn’s Index (DI) [5]. The criteria under consideration are widely used for estimation of fuzzy clustering methods’ functioning and demonstrate a quality of data partition into clusters.

Table 2 and Table 3 present results of the above

mentioned algorithms for the Wine dataset with restored 100 missing values. According to the tables, the developed approach to restoration of missing values provides the best quality results for data partition into clusters.

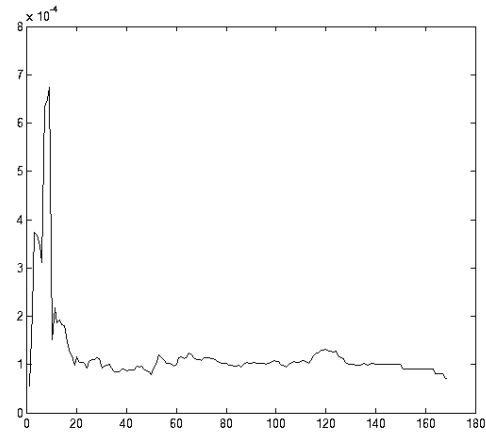


Fig.5. Errors for the Wine Dataset (10 Gaps)

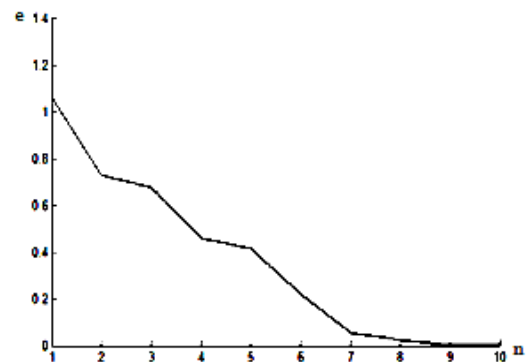


Fig.6. Changes in the errors for the Developed Data Recovery Method

Table 2. Cluster Quality Assessment

Clustering algorithms	PC		CE		SC	
	AdA	Mean	AdA	Mean	AdA	Mean
FCM	0.5036	0.5025	0.8541	0.8559	1.6255	1.6374
GK	0.2755	0.2726	1.3386	1.3441	44.7481	51.0859
GG	0.2542	0.2542	1.3785	1.3785	4.4064e-5	4.4134e-5

Table 3. Cluster Quality Assessment

Clustering algorithms	S		XB		DI	
	AdA	Mean	AdA	Mean	AdA	Mean
FCM	0.0119	0.0120	0.0119	0.9692	0.1423	0.1423
GK	0.3350	0.3793	0.5717	0.5137	0.0782	0.0782
GG	3.2491e-7	3.2378e-7	53.3548	52.2223	0.1145	0.1145

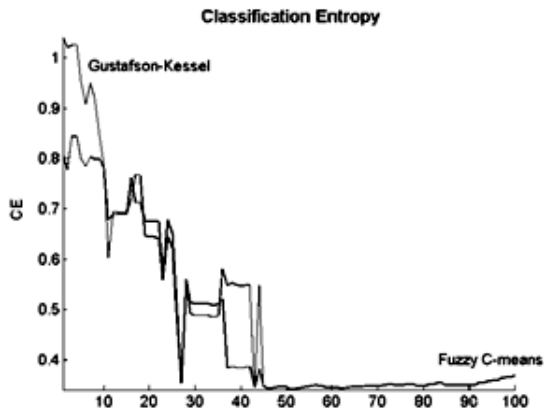


Fig.7. The CE index for FCM and GK

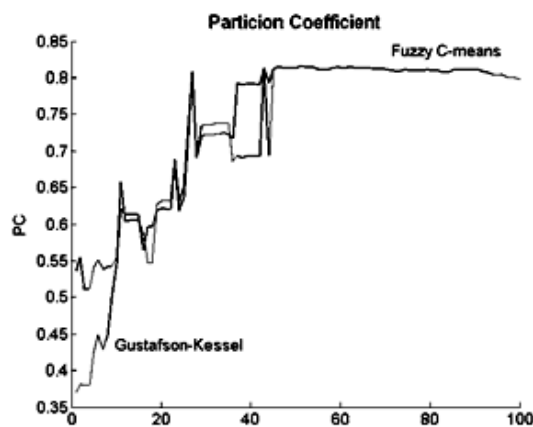


Fig.8. The PC index for FCM and GK

## VII. CONCLUSION

The task of gaps' recovery in "object-property" tables has been considered in the paper. Data come sequentially in this case. The adaptive neural system based on adaptive linear associators was developed that makes it possible to constantly adjust restored elements of the table in an online mode. This system can be characterized by a high speed and simple numerical implementation. The quality assessment procedure has been performed for various fuzzy clustering algorithms for data with missing values depending on their recovery methods.

## ACKNOWLEDGMENT

This scientific work was supported by RAMECS and CCNU16A02015.

## REFERENCES

- [1] A.K. Jain and R.C. Dubes, *Algorithms for Clustering Data*. Englewood Cliffs, N.J.: Prentice Hall, 1988.
- [2] L. Kaufman and P.J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*. N.Y.: John Wiley & Sons, Inc., 1990.
- [3] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*. San Francisco: Morgan Kaufmann, 2006.
- [4] G. Gan, C. Ma, and J. Wu. *Data Clustering: Theory, Algorithms, and Applications*. Philadelphia: SIAM, 2007.
- [5] J. Abonyi and B. Feil, *Cluster Analysis for Data Mining and System Identification*. Basel: Birkhäuser, 2007.
- [6] D.L. Olson and D. Dursun, *Advanced Data Mining Techniques*. Berlin: Springer, 2008.
- [7] C.C. Aggarwal and C.K. Reddy, *Data Clustering: Algorithms and Applications*. Boca Raton: CRC Press, 2014.
- [8] R. Xu and D.C. Wunsch, *Clustering*. Hoboken, NJ: John Wiley & Sons, Inc. 2009.
- [9] F. Hoepfner, F. Klawonn, R. Kruse, T. Runkler, *Fuzzy Clustering Analysis: Methods for Classification, Data Analysis and Image Recognition*. Chichester: John Wiley & Sons, 1999.
- [10] Ye. Bodyanskiy, O. Tyshchenko, and D. Kopalani, "An evolving neuro-fuzzy system for online fuzzy clustering", Proc. Xth Int. Scientific and Technical Conf. "Computer Sciences and Information Technologies (CSIT'2015)", pp.158-161, 2015. DOI: 10.1109/STC-CSIT.2015.7325456.
- [11] Zh. Hu, Ye.V. Bodyanskiy, O.K. Tyshchenko, and V.O. Samitova, "Fuzzy Clustering Data Given in the Ordinal Scale", International Journal of Intelligent Systems and Applications (IJISA), Vol.9, No.1, pp.67-74, 2017. DOI: 10.5815/ijisa.2017.01.07.
- [12] Zh. Hu, Ye.V. Bodyanskiy, O.K. Tyshchenko, and V.O. Samitova, "Fuzzy Clustering Data Given on the Ordinal Scale Based on Membership and Likelihood Functions Sharing", International Journal of Intelligent Systems and Applications (IJISA), Vol.9, No.2, pp.1-9, 2017. DOI: 10.5815/ijisa.2017.02.01.
- [13] Zh. Hu, Ye.V. Bodyanskiy, O.K. Tyshchenko, V.O. Samitova, "Possibilistic Fuzzy Clustering for Categorical Data Arrays Based on Frequency Prototypes and Dissimilarity Measures", International Journal of Intelligent Systems and Applications (IJISA), Vol.9, No.5, pp.55-61, 2017. DOI: 10.5815/ijisa.2017.05.07.
- [14] Zh. Hu, Ye.V. Bodyanskiy, and O.K. Tyshchenko, "A Cascade Deep Neuro-Fuzzy System for High-Dimensional Online Possibilistic Fuzzy Clustering", Proc. of the XI-th International Scientific and Technical Conference "Computer Science and Information Technologies" (CSIT 2016), pp.119-122, 2016. DOI: 10.1109/STC-CSIT.2016.7589884.
- [15] Zh. Hu, Ye.V. Bodyanskiy, and O.K. Tyshchenko, "A Deep Cascade Neuro-Fuzzy System for High-Dimensional Online Fuzzy Clustering", Proc. of the 2016 IEEE First Int. Conf. on Data Stream Mining & Processing (DSMP), pp.318-322, 2016. DOI: 10.1109/DSMP.2016.7583567.
- [16] Ye. Bodyanskiy, O. Tyshchenko, and D. Kopalani, "An Evolving Cascade Neuro-Fuzzy System for Data Stream Fuzzy Clustering", in *International Journal of Computer Science and Mobile Computing (IJCSMC)*, 2015, vol. 4(9), pp.270-275.
- [17] Zh. Hu, Ye. Bodyanskiy, O. Tyshchenko, and O. Boiko, "An Ensemble of Adaptive Neuro-Fuzzy Kohonen Networks for Online Data Stream Fuzzy Clustering", in *I.J. Modern Education and Computer Science (IJMECS)*, 2016, vol. 8(5), pp.12-18.
- [18] K.-L. Du and M.N.S. Swamy, *Neural Networks and Statistical Learning*. London: Springer-Verlag, 2014.
- [19] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning. Data Mining, Inference, and Prediction*. N.Y.: Springer Science & Business Media, LLC, 2009.
- [20] C.C. Aggarwal, *Data Mining*. Cham: Springer, Int. Publ. Switzerland, 2015.

- [21] T. Marwala, *Computational Intelligence for Missing Data Imputation, Estimation, and Management: Knowledge Optimization Techniques*. Hershey-New York: Information Science Reference, 2009.
- [22] A.N. Gorban, A.A. Rossiev, D.C. Wunsch, "Neural network modeling of data with gaps", *Radioelectronics Informatics. Control*, 2000, No.1 (3), pp. 47-55.
- [23] M. Tkacz, "Artificial neural networks in incomplete data sets processing", *Intelligent Information Processing and Web Mining*, 2005, pp.577-583.
- [24] R. Kruse, C. Borgelt, F. Klawonn, C. Moewes, M. Steinbrecher, and P. Held, *Computational Intelligence. A Methodological Introduction*. Berlin: Springer-Verlag, 2013.
- [25] Ye. Bodyanskiy, "Computational intelligence techniques for data analysis", in *Lecture Notes in Informatics*, 2005, P-72, pp.15-36.
- [26] Ye. Bodyanskiy, A. Dolotov, "Image processing using self-learning fuzzy spiking neural network in the presence of overlapping classes", Proc. of the 11th Biennial Baltic Electronics Conf. (BEC 2008), pp.213-216, 2008. DOI: 10.1109/BEC.2008.4657517.
- [27] J-S.R. Jang, C.T. Sun and E. Mizutani, *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*, New Jersey: Prentice Hall, 1997.
- [28] Ye. Bodyanskiy, O. Tyshchenko, and D. Kopalani, "A hybrid cascade neural network with an optimized pool in each cascade", *Soft Computing (Soft Comput)*, Vol.19, No.12, pp.3445-3454, 2015. DOI: 10.1007/s00500-014-1344-3.
- [29] Ye. Bodyanskiy, O. Tyshchenko, and D. Kopalani, "Adaptive learning of an evolving cascade neo-fuzzy system in data stream mining tasks", *Evolving Systems*, Vol.7, No.2, pp.107-116, 2016. DOI: 10.1007/s12530-016-9149-5.
- [30] Zh. Hu, Ye.V. Bodyanskiy, O.K. Tyshchenko, and O.O. Boiko, "An Evolving Cascade System Based on a Set of Neo-Fuzzy Nodes", *International Journal of Intelligent Systems and Applications (IJISA)*, Vol. 8(9), pp.1-7, 2016. DOI: 10.5815/ijisa.2016.09.01.
- [31] D. Graupe, *Principles of Artificial Neural Networks (Advanced Series in Circuits and Systems)*. Singapore: World Scientific Publishing Co. Pte. Ltd., 2007.
- [32] K. Suzuki, *Artificial Neural Networks: Architectures and Applications*. NY: InTech, 2013.
- [33] S. Haykin, *Neural Networks and Learning Machines (3rd Edition)*. NJ: Prentice Hall, 2009.
- [34] S. Haykin, *Neural networks: a comprehensive foundation*. Upper Saddle River, N.J.: Prentice Hall, 2008.
- [35] A. Gorban, B. Kegl, B. Wunsch, and A. Zinovyev, *Principal Manifolds for Data Visualization and Dimension Reduction*. Berlin-Heidelberg-New York: Springer, 2007.
- [36] C.M. Bishop, *Neural Networks for Pattern Recognition*. Oxford: Clarendon Press, 1995.
- [37] A. Albert, *Regression, pseudo-inversion and recurrent estimation*. Moscow: Nauka, 1977.
- [38] Ye. Bodyanskiy, I.P. Pliss, T.V. Solovyova, "Multistep optimal predictors of multidimensional nonstationary stochastic processes", *Doklady AN USSR*, 1986, No.12, pp.47-49. (in Russian)
- [39] Ye. Bodyanskiy, V. Kolodyazhnyi, and A. Stephan, "An adaptive learning algorithm for a neuro-fuzzy network", *Computational Intelligence. Theory and Applications*, 2001, pp.68-75.
- [40] T. Yamakawa, E. Uchino, T. Miki and H. Kusanagi, "A

neo fuzzy neuron and its applications to system identification and prediction of the system behavior", Proc. 2nd Int. Conf. on Fuzzy Logic and Neural Networks, pp. 477-483, 1992.

- [41] E. Uchino and T. Yamakawa, "Soft computing based signal prediction, restoration and filtering", *Intelligent Hybrid Systems: Fuzzy Logic, Neural Networks and Genetic Algorithms*, Boston: Kluwer Academic Publisher, pp. 331-349, 1997.
- [42] T. Miki and T. Yamakawa, "Analog implementation of neo-fuzzy neuron and its on-board learning", *Computational Intelligence and Applications*, Piraeus: WSES Press, pp. 144-149, 1999.
- [43] Ye. Bodyanskiy, O. Tyshchenko, and D. Kopalani, "An Extended Neo-Fuzzy Neuron and its Adaptive Learning Algorithm", *IJ. Intelligent Systems and Applications (IJISA)*, Vol.7(2), pp.21-26, 2015.

### Authors' Profiles



**Zhengbing Hu:** Ph.D., Associate Professor of School of Educational Information Technology, Central China Normal University, M.Sc. (2002), Ph.D. (2006) from the National Technical University of Ukraine "Kiev Polytechnic Institute". Postdoc (2008), Huazhong University of Science and Technology, China. Honorary Associate Researcher (2012), Hong Kong University, Hong Kong. Major interests: Computer Science and Technology Applications, Artificial Intelligence, Network Security, Communications, Data Processing, Cloud Computing, Education Technology.



**Yevgeniy Bodyanskiy.** graduated from Kharkiv National University of Radio Electronics in 1971. He got his PhD in 1980. He obtained an academic title of the Senior Researcher in 1984. He got his Dr.Habil.Sci.Eng. in 1990. He obtained an academic title of the Professor in 1994. Prof. Bodyanskiy is Professor of Artificial Intelligence Department at KhNURE, Head of Control Systems Research Laboratory at KhNURE. He has more than 600 scientific publications including 40 inventions and 10 monographs. His research interests are Hybrid Systems of Computational Intelligence: adaptive, neuro-, wavelet-, neo-fuzzy-, real-time systems that have to do with control, identification, and forecasting, clustering, diagnostics and fault detection.

Prof. Bodyanskiy is an IEEE Senior Member and a member of 4 scientific and 7 editorial boards.



**Oleksii Tyshchenko** graduated from Kharkiv National University of Radio Electronics in 2008. He got his PhD in Computer Science in 2013. He is currently working as a Senior Researcher at Control Systems Research Laboratory, Kharkiv National University of Radio Electronics. He has currently published more than 50 publications. He is a reviewer of such journals as *Neural Computing and Applications (NCAA)*; *Soft*



Computing (SoCo); Evolving Systems (EvoS); Neurocomputing (NeuroComp); IEEE Transactions on Cybernetics.

His current research interests are Evolving, Reservoir and Cascade Neuro-Fuzzy Systems; Computational Intelligence; Machine Learning; Deep Learning; High-Dimensional Fuzzy Clustering.



**Vitalii Tkachov** graduated from Kharkiv National University of Radio Electronics in 2011. He got his PhD in Information Technology in 2015. He is currently Senior Lecturer in the Department of Electronic Computers and Head of Wireless and Campus Network Research Laboratory at Kharkiv National University of Radio Electronics. His research interests are telecommunications and information technology.

**How to cite this paper:** Zhengbing Hu, Yevgeniy V. Bodyanskiy, Oleksii K. Tyshchenko, Vitalii M. Tkachov, "Fuzzy Clustering Data Arrays with Omitted Observations", *International Journal of Intelligent Systems and Applications(IJISA)*, Vol.9, No.6, pp.24-32, 2017. DOI: 10.5815/ijisa.2017.06.03