UDC 004.052.3:004.032.26

V.S. Yakovyna, PhD, Assoc.Prof.
Lviv Polytechnic National University, 12 Bandera Str., 79013 Lviv, Ukraine; e-mail: vitaliy.s.yakovyna@lpnu.ua

SOFTWARE FAILURES PREDICTION USING RBF NEURAL NETWORK

В.С. Яковина. **Прогнозування відмов програмного забезпечення засобами нейронної мережі RBF.** До перспективних методів прогнозування надійності програмного забезпечення можна віднести методи на основі непараметричних моделей, зокрема штучних нейронних мереж. В роботі проведено дослідження впливу кількості вхідних нейронів мережі на основі радіально-базисних функцій на ефективність прогнозування відмов програмного забезпечення, представлених у вигляді часових рядів. Часові ряди відмов програмного забезпечення отримували з результатів тестування веб-браузера Chromium та операційної системи Chromium-OS з відкритим вихідним кодом з запропонованою подальшою обробкою у вигляді нормалізованих значень кількості відмов програмних продуктів за рівні інтервали часу з наступним переведенням до людино-днів. Показано, що найкраща точність прогнозування досягалась для функції активації Inverse Multiquadric при 10…20 нейронах вхідного шару та 30 нейронах прихованого шару.
Ключові слова: програмне забезпечення, надійність, відмова, RBF нейронні мережі, часові ряди.

V.S. Yakovyna. **Software failures prediction using RBF neural network.** One of the prospective techniques for software reliability prediction based on nonparametric models, in particular on artificial neural networks. In this paper the study of influence of number of input neurons of network based on radial basis function on the efficiency of software failures prediction presented in the form of time series is carried out. Software faults time series are constructed using Chromium and Chromium-OS open source software systems testing data with proposed further processing as a normalized values of the number of software failures in equal intervals, followed by transfer to man-days. It is demonstrated that the closest prediction can be achieved using Inverse Multiquadric activation function with 10…20 input layer neurons and 30 hidden neurons.
Keywords: software, reliability, failure, RBF neural network, time series.

Introduction. To date, significant achievements in science and creating breakthrough technologies require increasing the growth of capacity of modern computer technology. Hardware development of modern computer technology has led to the rapid development of software industry. At the same time hardware-software complexes are included in all spheres of human life and failures of program component carry a significant risk. However, the growing complexity of software systems makes the nontrivial task of quality control. The important characteristic of software quality that directly affects the amount of risk is its reliability.

The vast majority of software reliability models are analytical models that are derived from assumptions about the mechanisms of failure. These studies are focused on the model assumptions and the interpretation of its parameters. However, the complexity of modern software and complex interactions with the environment of functioning significantly complicate the construction of analytical models of satisfactory degree of adequacy. As a result, there is a contradiction between responsibility and complexity of modern software and its reliability analysis methods [1].

Literature review. Promising methods of software reliability prediction, including the number of failures (or failure flow parameter as a derivative function) can be methods based on nonparametric models [2, 3]. Such models do not have main drawbacks and difficulties of analytical models because they do not make any assumption about the mechanism of software failures. The basis of these methods is the description of stochastic process software failures based on reliable and representative statistical sample. As these models artificial neural networks are widely used because of their proven quality of generalization and approximation of almost any smooth functions [4]. The main advantage of such methods of prognosis include the lack of a priori assumptions about the regularities of stochastic

DOI 10.15276/opu.2.46.2015.20

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

process, while the main disadvantage is the need to obtain sufficient statistical sampling and the need to train the neural network for each new sample.

One of the perspective class of neural networks which is used for approximation of functions is network based on radial basis function (RBF). In general form, the radial basic function is real function whose value depends only on the distance from a point **c**, so

$$\varphi(\mathbf{x},\mathbf{c}) = \varphi(\|\mathbf{x}-\mathbf{c}\|) . \quad (1)$$

Norm in (1) usually corresponds to Euclidean distance though any other function of distance can be used [5, 6].

Radial basic functions (1) are often used to approximate functions in the form

$$y(\mathbf{x}) = \sum_{i=1}^{N} w_i \varphi(\|\mathbf{x}-\mathbf{x}_i\|) \quad (2)$$

Here the approximating function $y(\mathbf{x})$ is represented as the sum of N radial basis functions with weight coefficients w_i, which is also interpreted as a simple single-layer type of neural network [5].

Radial basic neural network is characterized by following three properties [7]:

— one hidden layer;

— only the hidden layer neurons have nonlinear activation function;

—synaptic weights of connections of input and hidden layers are equal to one.

The most common activation functions of RBF neural network are Gaussian, Multiquadric, Inverse Quadratic, Inverse Multiquadric and Polyharmonic spline [8].

Activation function Gaussian is

$$\varphi(r) = e^{-(\varepsilon r)^2} , \quad (3)$$

and function Inverse Multiquadric is

$$\varphi(r) = \frac{1}{\sqrt{1+(\varepsilon r)^2}} . \quad (4)$$

At the same time the normalizing multiplier before the exponential in (3) is absent, because normalization is carried out through the weight coefficients of neural network (2). This function is widely used in the method of support vectors that are used for tasks of classification and regression analysis [9].

Another widely used activation function is Inverse Multiquadric (4). So in [10] it is demonstrated that single-layer neural networks with activation function Inverse Multiquadric and (n+1) hidden layer neurons can learn (n+1) different sets of input data with zero error. These neural networks can interpolate with arbitrary accuracy any set of single- or multiple-dimensional data; they can uniformly approximate any continuous function of one variable [10].

Aim of the Research. In previous works [11, 12] it is shown the feasibility and efficiency of usage of neural network based on radial basis functions. Also it is studied the influence of activation function of such neural network on the prediction efficiency of software failures number and it is shown that Gaussian function is the best in terms of training speed while Inverse Multiquadric is the best in terms of prediction accuracy. Aim of the research is to study the influence of the number of neurons in the input layer of network RBF with activation functions of Gaussian (3) and Inverse Multiquadric (4) on the efficiency of prediction of failures process for definition the configuration of neural network for development of tools of software failures prediction.

Main Body. There are two series of experiments for each activation function, which used a different number of neurons in the input layer — from 5 to 45 with increments of 5. Neural network with 30 hidden layer neurons is constructed at first series, with 10 ones — at second series. Training of neural network is carried out till the error 0,005 is achieved or till 5000 teaching periods, depending on what occurred earlier.

The input data for neural network training and failures prediction are time series received from the results of testing of two software products with open source: web browser Chromium [13] and operating system Chromium-OS [14]. Usage of the results of testing of two software products is to confirm the results and conclusions of the applicability of the investigated configuration of neural network for software failures prediction. Each point of the time series is the value of failures number on the time interval for which the whole range of test length has been split. These intervals are uniformly selected, while the test results are presented in original form, without any normalization. Therefore, the input data are normalized before the study. Since different number of testers has participated the testing, the initial data is transferred in man-days. Conducted in previous research experiments have shown that the time series in the form of dependence "number of failures — time interval" is unsatisfactory predicted both Elman neural network and RBF [12]. Instead, this study uses time series as the cumulative number of failures detected at all time intervals till the current one. The use of such presentation of software test results allows to improve significantly the accuracy of prediction of failures in the software product using neural networks (both recurrent and RBF) [11, 12]. After such preliminary processing the obtained time series is used for training the neural network. All the test results in this study are divided into 150 equal time intervals. The first 100 intervals are used for training the neural network, and the last 50 — for verification the prediction accuracy (predicted by neural network results are compared with these control values).

To evaluate the efficiency of prediction there are used the following parameters: the number of periods of study that characterizes the rate of neural network training; the coefficient of determination (R^2), which is considered universal measure of one random variable dependence on set of others, and the mean square error of approximation that characterizes the approximation of model to statistics in the uniform metric and shows proximity of predicted and experimental data.

Results. Typical results of statistical describe the prediction efficiency of software failures are given in Table 1 for the Chromium browser and in Table 2 for a system of Chromium-OS (columns in these tables are indicated by the letter G correspond to the activation function Gaussian, and columns indicated by the letters IM — to the activation function Inverse Multiquadric). These tables show that although the absolute values of the study duration, the coefficient of determination and mean square error of approximation are different, the trends of their change depending on the configuration of neural networks are very similar.

Table 1

The dependence of prediction efficiency of defects of Chromium browser on the number of input neurons

Number of input neurons	Number of hidden neurons	Study duration, periods		Coefficient of determination		Mean square error, %	
		G	IM	G	IM	G	IM
10	30	146	478	0,996	0,998	4,9	1,3
15	30	14	171	0,997	0,998	8,0	2,4
45	30	443	27	0,997	0,994	64,7	29,0
5	10	69	1976	0,996	0,998	3,2	3,6
15	10	78	3916	0,997	0,998	2,2	12,4
25	10	68	6991	0,507	0,998	12,1	8,5

Table 2

The dependence of prediction efficiency of defects of Chromium-OS on the number of input neurons

Number of input neurons	Number of hidden neurons	Study duration, periods		Coefficient of determination		Mean square error, %	
		G	IM	G	IM	G	IM
10	30	212	324	0,882	0,983	5,1	3,2
15	30	59	279	0,969	0,970	4,8	3,3
45	30	113	26	0,988	0,971	24,5	22,4
5	10	364	1618	0,958	0,991	3,2	2,4
15	10	37	1536	0,987	0,979	5,4	3,0
25	10	62	260	0,982	0,967	7,4	3,4

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

The dependence of the mean square error of approximation on the number of neurons in the input layer in case of failures prediction of browser Chromium is shown at Fig. 1 for activation function Gaussian, and at Fig. 2 for the case of activation function Inverse Multiquadric. These figures show that in both cases too big number neurons in the input layer leads to a significant increase of prediction errors, which in general can be explained by the difficulty of approximation of a random process without explicit distribution for a long period of time (the number of input neurons determines the background which is taken into account when predicting the next value).

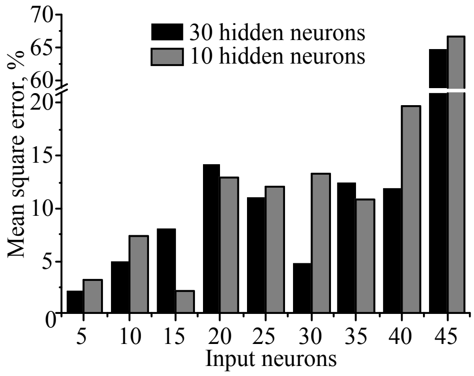

Fig. 1. The dependence of the mean square error of approximation on the number of neurons in the input layer for activation function Gaussian

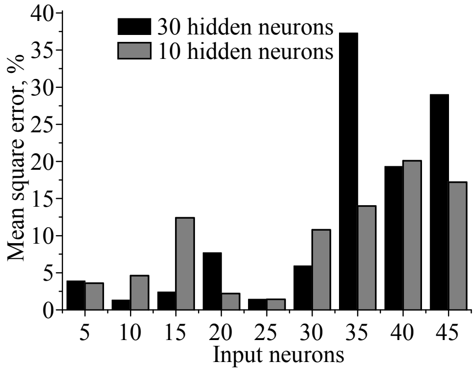

Fig. 2. The dependence of the mean square error of approximation on the number of neurons in the input layer for activation function Inverse Multiquadric

To compare the accuracy of prediction of time series of different software failures Fig. 3 shows the dependence of the mean square error of approximation on neural network configuration (submitted at figure in the form of two numbers *n*–*m*, where *n* is the number of neurons in the input layer, *m* is the number of hidden layer neurons) for activation function Gaussian for both studied software products. As it can be clearly seen at Fig. 3, the trend of prediction accuracy changes depending on the configuration of the neural network is the same for both software products — the significant increase of error with increasing the number of neurons in the input layer and generally lower error values for the configuration with ten hidden layer neurons.

Results of research of influence of neural network configuration on duration of training are shown at Fig. 4. It shows the dependence of the relative efficiency of training (the ratio of training duration when using activation function Inverse Multiquadric to training duration when using activation function Gaussian) on neural network configurations for two studied software products. As shown at Fig. 4, despite the various absolute values of training duration (see Tables 1, 2) and some variations in values of training relative efficiency, the overall trend remains for both time series — the minimal difference in rate of training is observed for configuration "45—30" and maximal — for configurations "15—10" and "25—10". Since that we can also conclude that training processes are approximately the same using different time series with similar characteristics.

Fig. 5 and 6 show the time dependencies of predicted and experimental number of failures of web browser Chromium, and Fig. 7 — the dependencies of predicted and experimental number of failures of system Chromium-OS.

The prediction is carried out using RBF neural network with the following parameters: function activation Gaussian, 15 neurons in the input layer, 10 neurons in the hidden layer; duration of neural network training of such configuration is 78 periods (Fig. 5); activation function Inverse Multiquadric, 10 neurons in the input layer, 30 neurons in the hidden layer (Fig. 6, 7); duration of neural network training of such configuration is 478 periods in case of the system Chromium and 324 periods in case of system Chromium-OS. As can be seen from these figures, a high degree of proximity of predicted and empirical data can be achieved by selection of parameters.

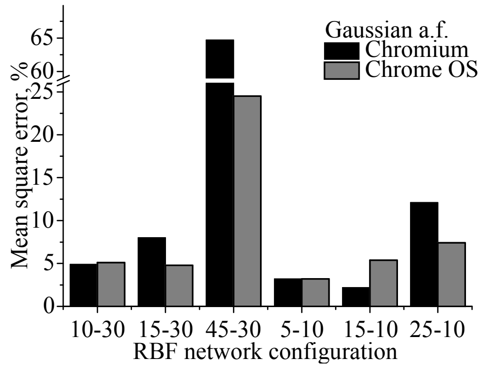

Fig. 3. The dependence of the mean square error of approximation on neural network configuration for two software products in the case of activation function Gaussian

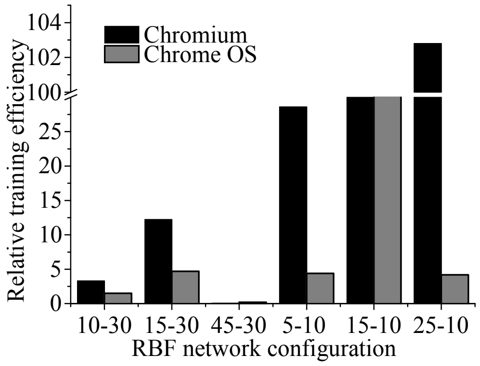

Fig. 4. The dependence of the relative duration of study for two activation functions on neural network configurations for two software products

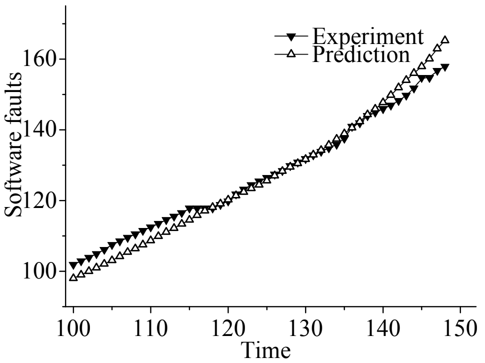

Fig. 5. The example of prediction of the number of failures for browser Chromium by RBF neural network with activation function Gaussian

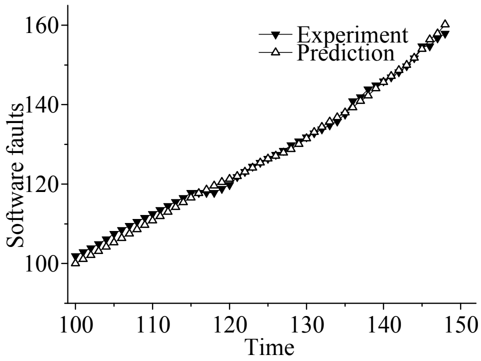

Fig. 6. The example of prediction of the number of failures for browser Chromium by RBF neural network with activation function Inverse Multiquadric

As can be seen from Fig. 1, 5 in the case of usage of activation function Gaussian in neural network the best results of prediction are obtained at small values of incoming neurons number (up to 15). This fact corresponds to taking into account while predicting the small software failures prehistory. When the number of input neurons is 40…45 (26,7…30 % of the training set) the prediction results are unacceptable, which is confirmed by the data of Table 2, obtained on the basis of Chromium-OS testing. However, it should be noted that the minimal duration of training is not always consistent with the maximal accuracy of prediction.

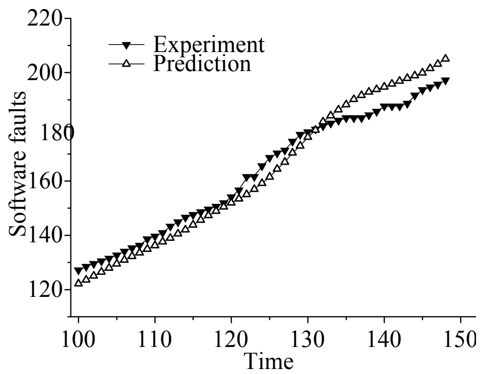

Fig. 7. The example of prediction of the number of failures for system Chromium-OS by RBF neural network with activation function Inverse Multiquadric

In the case of system Chromium-OS the usage of activation function Gaussian gives the best prediction accuracy (mean square error is 3,2 %) in the configuration “5—10”, as in the case of browser Chromium (see Tables 1, 2). At the same time the duration of study in this configuration for both studied software is much smaller in the case of activation function Gaussian (69 against 1976 periods in the case of browser Chromium, and 364 against 1618 periods in the case of system Chromium-OS).

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

In the case of activation function Inverse Multiquadric (Fig. 2, 6, 7) we can conclude that just as in the case of activation function Gaussian, the best predicting results are obtained at small values of incoming neurons number (up to 25). This fact corresponds to taking into account while predicting the small software failures prehistory. When the number of input neurons is 40…45 (30 % of the training set) the prediction results are unacceptable in the case of failures prediction for both studied in this paper software products. Just as in the case of using the activation function Gaussian, the minimal duration of training is not always consistent with the maximal accuracy of the prediction. However, it should be noted that using the activation function Inverse Multiquadric unlike Gaussian often leads to paralysis of neural network training, and overall the average duration of education is higher in the order. At the same time, the maximal prediction accuracy is achieved when using the activation function Inverse Multiquadric (mean square error of approximation is 1,3 % for browser Chromium and 3,2 % for system Chromium-OS), which confirms the conclusions of [11].

Table 2 shows that in the case of time series constructed on the basis of Chromium-OS testing the best results of failures prediction (mean square error of approximation is 3,2%, coefficient of determination is 0,983) are achieved using activation function Inverse Multiquadric and configuration "10—30" (10 input neurons and 30 hidden layer neurons) as it is illustrated at Fig. 7. Thus, Table 2 shows that a slightly smaller approximation error for this software can be achieved using activation function Inverse Multiquadric and configuration "5—10" or "15—10" (2,4% and 3,0 % accordingly), but the training duration is 1618 and 1536 periods against 324 periods in the case of configuration "10—30". On the other hand activation function Gaussian in configuration "5—10" makes it possible to achieve a similar approximation error at comparable duration of training, but the coefficient of determination in this case is worse — 0,958 (Table 2). Thus the conclusion about the best accuracy of prediction of time series of software failures with specified configuration of RBF neural network is also confirmed for software failures of Chromium-OS with other metrics and complexity that gives grounds to conclude that the applicability of such a configuration for prediction of time series of failures of various software products is possible.

Conclusions. In summary, we can conclude that at the use of RBF neural network the best results were obtained in the configuration of activation function Inverse Multiquadric, 10 neurons in the input layer and 30 neurons in the hidden one. In this configuration the mean square error of approximation is 1,0 % and the coefficient of determination is between the predicted and control samples — 0,9965. However, due to a slight deterioration in accuracy of prediction to 1,7 %, the duration of the neural network training can be reduces in 3…6 times through the use of activation function Gaussian with 15 input neurons at 10 neurons in hidden layer [15]. In this configuration, the relative mean square error of approximation is 1,7 % and the coefficient of determination is between the predicted and control samples — 0,9969. Obtaining of similar results for two different by characteristics software products leads to the conclusion that the resulting configuration of RBF neural network can be successfully used to predict similar time series — the uniform failures process, in the form of cumulative time series. Further studies will focus on determining the number of hidden layer neurons, which will more accurately predict the failure of the software.

Література

1. Pham, H. System Software Reliability / H. Pham. — London: Springer, 2007. — 440 p.
2. Khoshgoftaar, T.M. Predicting software quality, during testing, using neural network models: A comparative study / T.M. Khoshgoftaar, R.M. Szabo // International Journal of Reliability, Quality and Safety Engineering. — 1994. — Vol. 1, Issue 3. — PP. 303 — 319.
3. Zheng, J. Predicting software reliability with neural network ensembles / J. Zheng // Expert Systems with Applications. — 2009. — Vol. 36, Issue 2. — PP. 2116 — 2122.
4. Paliwal, M. Neural networks and statistical techniques: A review of applications / M. Paliwal, U.A. Kumar // Expert Systems with Applications. — 2009. — Vol. 36, Issue 1. — PP. 2 — 17.

5. Karlik, B. Performance analysis of various activation functions in generalized MLP architectures of neural networks / B. Karlik, A.V. Olgac // International Journal of Artificial Intelligence and Expert Systems. — 2010. — Vol. 1, Issue 4. — PP. 111 — 122.
6. Buhmann, M.D. Radial Basis Functions: Theory and Implementations / M.D. Buhmann. — Cambridge: Cambridge University Press, 2009. — 272 p.
7. Broomhead, D.S. Multivariable functional interpolation and adaptive networks / D.S. Broomhead, D. Lowe // Complex Systems. — 1988. — Vol. 2, Issue 3. — PP. 321 — 355.
8. Yee, P.V. Regularized Radial Basis Function Networks: Theory and Applications / P.V. Yee, S. Haykin. — New York: Wiley, 2001. — 191 p.
9. Comparing support vector machines with Gaussian kernels to radial basis function classifiers / B. Schölkopf, K.-K. Sung, C.J.C. Burges, *etc.* // IEEE Transactions on Signal Processing. — 1997. — Vol. 45, Issue 11. — PP. 2758 — 2765.
10. Han, X. Approximate interpolation by neural networks with the inverse multiquadric functions / X. Han // Proceedings of Second International Symposium on Advances in Computation and Intelligence (ISICA 2007), Wuhan, China, September 21–23, 2007. — Berlin: Springer-Verlag, 2007. — PP. 296 — 304.
11. Yakovyna, V. Influence of the activation function of RBF neural network on software reliability time series prediction / V. Yakovyna, V. Ketsman // Proceedings of the VIIIth International Conference on Perspective Technologies and Methods in MEMS Design (MEMSTECH’2012), Polyana-Svalyava, Ukraine, April 18–21, 2012. — Lviv: Publishing House of Lviv Polytechnic National University, 2012. — PP. 91 — 92.
12. On the possibility of software reliability prediction using RBF neural network / V. Yakovyna, O. Synytska, T. Kremen, V. Smirnov // Proceedings of the VIth International Scientific and Technical Conference „Computer Science and Information Technologies“ (CSIT’2011), Lviv, Ukraine, November 16–19, 2011. — Lviv: Publishing House Vezha & Co., 2011. — PP. 39 — 42.
13. Chromium: An open-source project to help move the web forward [Електронний ресурс] / Google. — Режим доступу: https://code.google.com/p/chromium/issues/list (Дата звернення: 10.03.2015).
14. Chromium-OS [Електронний ресурс] / Google. — Режим доступу: https://code.google.com/p/chromium-os/issues/list (Дата звернення: 10.03.2015).
15. Yakovyna, V. Influence of input neurons count for RBF neural network with Gaussian activation function on software faults prediction efficiency / V. Yakovyna, I. Garandzha // Proceedings of International conference “Intellectual Systems for Decision Making and Problems of Computational Intelligence” (ISDMCI’2013), Yevpatoria, Ukraine, May 20–24, 2013. — Kherson: Kherson National Technical University, 2013. — PP. 520 — 522.

References

1. Pham, H. (2007). *System Software Reliability*. London: Springer.
2. Khoshgoftaar, T.M. and Szabo, R.M. (1994). Predicting software quality, during testing, using neural network models: A comparative study. *International Journal of Reliability, Quality and Safety Engineering*, 1(3), 303-319.
3. Zheng, J. (2009). Predicting software reliability with neural network ensembles. *Expert Systems with Applications*, 36(2), 2116-2122.
4. Paliwal, M. and Kumar, U.A. (2009). Neural networks and statistical techniques: A review of applications. *Expert Systems with Applications*, 36(1), 2-17.
5. Karlik, B. and Olgac, A.V. (2010). Performance analysis of various activation functions in generalized MLP architectures of neural networks. *International Journal of Artificial Intelligence and Expert Systems*, 1(4), 111-122.
6. Buhmann, M.D. (2009). *Radial Basis Functions: Theory and Implementations*. Cambridge: Cambridge University Press.
7. Broomhead, D.S. and Lowe, D. (1988). Multivariable functional interpolation and adaptive networks. *Complex Systems*, 2(3), 321-355.
8. Yee, P.V. and Haykin, S. (2001). *Regularized Radial Basis Function Networks: Theory and Applications*. New York: Wiley.
9. Schölkopf, B., Sung, K.-K., Burges, C.J.C., Girosi, F., Niyogi, P., Poggio, T. and Vapnik, V. (1997). Comparing support vector machines with Gaussian kernels to radial basis function classifiers. *IEEE Transactions on Signal Processing*, 45(11), 2758-2765.

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

10. Han, X. (2007). Approximate interpolation by neural networks with the inverse multiquadric functions. In L. Kang, Y. Liu, S. Zeng (Eds.), *Proceedings of Second International Symposium on Advances in Computation and Intelligence (ISICA 2007)* (pp. 296-304). Berlin: Springer-Verlag.
11. Yakovyna, V. and Ketsman, V. (2012). Influence of the activation function of RBF neural network on software reliability time series prediction. In *Proceedings of the VIIIth International Conference on Perspective Technologies and Methods in MEMS Design (MEMSTECH'2012)* (pp. 91-92). Lviv: Publishing House of Lviv Polytechnic National University.
12. Yakovyna, V., Synytska, O., Kremen, T. and Smirnov, V. (2011). On the possibility of software reliability prediction using RBF neural network. In *Proceedings of the VIth International Scientific and Technical Conference „Computer Science and Information Technologies“ (CSIT'2011)* (pp. 39-42). Lviv: Publishing House Vezha & Co.
13. *Chromium: An open-source project to help move the web forward*. Retrieved from https://code.google.com/p/chromium/issues/list
14. *Chromium-OS*. Retrieved from https://code.google.com/p/chromium-os/issues/list
15. Yakovyna, V. and Garandzha, I. (2013). Influence of input neurons count for RBF neural network with Gaussian activation function on software faults prediction efficiency. In *Proceedings of International Conference "Intellectual Systems for Decision Making and Problems of Computational Intelligence" (ISDMCI'2013)* (pp. 520-522). Kherson: Kherson National Technical University.

Received May 5, 2015