UDC 004.9: 004.05
https://doi.org/10.31891/CSIT-2022-1-9

MYKYTA LEBIGA,
TETIANA HOVORUSHCHENKO, MARIIA KAPUSTIAN
Khmelnytskyi National University

NEURAL-NETWORK MODEL OF SOFTWARE QUALITY PREDICTION BASED ON QUALITY ATTRIBUTES

The paper proposes a neural-network model of software quality prediction based on quality attributes. The proposed model differs from the known models, because it provides considering the importance of each quality attribute and their interaction within each software quality characteristic. The artificial neural network (ANN) outputs correspond to the values of software quality characteristics (functional suitability, performance efficiency, usability, reliability, compatibility, security, maintainability, portability). The artificial neural network (ANN) outputs make it possible assessing the total impact of quality attributes on software quality characteristics.

Key words: *software, software quality, software quality attributes, software quality characteristics, artificial neural network (ANN).*

МИКИТА ЛЕБІГА,
ТЕТЯНА ГОВОРУЩЕНКО, МАРІЯ КАПУСТЯН
Хмельницький національний університет

НЕЙРОМЕРЕЖНА МОДЕЛЬ ПРОГНОЗУВАННЯ ЯКОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ НА ОСНОВІ АТРИБУТІВ ЯКОСТІ

Якість програмного забезпечення (ПЗ) стає все більш важливою темою протягом останнього десятиліття в зв'язку із стрімким зростанням ринку програмного забезпечення, із збільшенням конкуренції в індустрії ПЗ, а також із зростанням складності програмного забезпечення та покладеної відповідальності на нього. Актуальність проблем підвищення якості ПЗ обумовлює необхідність прогнозування якості програмного забезпечення на ранніх етапах життєвого циклу та розроблення моделей, методів та засобів прогнозування якості ПЗ на основі атрибутів.

Універсальними структурами, що дозволяють узагальнити інформацію та виявити залежності між вхідними і результуючими даними, є штучні нейронні мережі (ШНМ), тому для реалізації функцій, які враховуватимуть взаємовпливи характеристик та атрибутів якості запропоновано використовувати саме ШНМ. Авторами запропоновано концепцію прогнозування якості ПЗ на основі ШНМ, в якій враховуються взаємовпливи атрибутів якості при визначенні характеристик якості ПЗ.

Вперше запропонована нейромережна модель прогнозування якості програмного забезпечення на основі атрибутів якості відрізняється від відомих тим, що дає можливість враховувати важливість кожного атрибуту якості, а також взаємний вплив атрибутів в межах кожної характеристики якості програмного забезпечення. Вихідні функціонали ШНМ, що відповідають значенням характеристик якості ПЗ (функційна придатність, ефективність, зручність використання, надійність, сумісність, захищеність, супроводжуваність, можливість переносу), дають можливість оцінити сумарний вплив атрибутів якості на характеристики якості програмного забезпечення.

Враховуючи запропоновану концепцію, перспективними завданнями для подальших досліджень є: розроблення нейромережного методу прогнозування рівня якості програмного забезпечення на основі атрибутів якості; реалізація та навчання нейромережної складової запропонованого методу; проєктування та реалізація технології прогнозування рівня якості програмного забезпечення на основі атрибутів якості на ранніх етапах життєвого циклу.

Ключові слова: *програмне забезпечення (ПЗ), якість ПЗ, атрибути якості ПЗ, характеристики якості ПЗ, штучна нейронна мережа (ШНМ).*

Introduction

Software quality has become an increasingly important issue over the last decade due to the rapid growth of the software market, increasing competition in the software industry, and the increasing complexity and responsibility of software. Insufficient software quality is the cause of disasters, accidents, financial, information, and reputational losses [1, 2].

Today, the software industry has many challenges in evaluating and monitoring the software quality. The main reasons are: the lack of an objective model for the software quality assessing, the lack of a clearly defined mechanism for quality attributes measuring [2, 3].

Software quality is determined by factors such as how the software is made, not how it is tested. Software quality assessment is important to ensure high-quality software products. ISO/IEC standards identify methods for the software quality assessing and provide common approaches. Software quality assessment according to ISO 25010:2011 [4] is performed as follows: based on 138 quality attributes from ISO 25023:2016 [5], 31 sub-characteristics and 8 quality characteristics are evaluated, after that the comprehensive software quality assessment is conducted.

The purpose of developing the software quality models is using of the attributes from the early stages of the life cycle of software development in order to provide initial assessments of the quality of the developed software. Currently, in most cases the software quality attributes are evaluated for ready-made software code. However the software quality can be predicted based on values of the attributes from the specification. After all any error in the specification will be reflected in the next stages of software development, leading to the development of low-quality software.

Timely predicting of software quality can be used to take any preventive measures to reduce the number of software failures during its operation. Therefore, the growth of the software market and increasing competition in this market requires the development of models, methods, and tools for quality assessment, and especially forecasting the quality of software in the early stages of the life cycle, taking into account economic and time feasibility.

Overview of known models, methods and tools for software quality prediction

The paper [6] defines the attributes of software testing quality, presents methods for assessing the quality of software testing, and analyzes and discusses models for assessing the quality of software testing based on the theory of fuzzy mathematics. This model allows you to quickly assess the quality of software testing, avoids local highs, overcomes the shortcomings of existing models and assessment tools, and can correctly reflect the relationship between internal and external properties of software.

The paper [7] proposes a model of the software product quality assessment process. This model is on the basis of the international standard ISO/IEC 14598 - Software Product Assessment. The proposed model generates an adapted model for assessing the quality of the software product considering the types of information systems. Accordingly, the necessary software measures are collected and further analyzed to provide feedback to improve the quality of the software product.

The paper [8] proposes a software development model that includes the integration of Goal-Question-Metrics and Deployment of software quality function. Using this model, software engineers formulate project objectives and verify that they are being achieved. The software development process using this model increase the usability of the software. The main contribution of this article is to help software developers create better software that not only meets the requirements of customers but also meets the objectives of the project.

The paper [9] developed a model for assessing the quality of system-level software that controls hardware devices. In particular, quality assessment metrics to evaluate system software for radio transmitter control software have been developed. To calculate the scores, the authors derived requirements based on the operating environment, applications, and operating software performance.

In the paper [10] a new method of software quality prediction is proposed, using the method of reference vectors of classification of software modules based on complexity indicators. A model of software classification based on the method of reference vectors is proposed, the characteristics of which are suitable for early forecasting of software quality when there is only a small amount of sample data.

The study [11] presents a model of the software maintenance process, which emphasizes the impact of software quality on maintenance and takes into account the quality of modified software.

The paper [12] proposes a software quality model for testing the various factors that impact on the software quality, and for increasing the software performance considering the complexity of the software that software developers may face. The proposed model showed how to provide a safe, reliable, and high-quality software product.

The paper [13] describes SQUID's approach to modeling, measurement, and evaluation of software quality. Project SQUID proposed a method and experimental tools for supporting and automating these actions. The tools helps in quality specification, planning, control, and assessment. These tools provide the set targets for product quality requirements and assess their feasibility (quality specification). These tools supports the identification of the internal attributes of the software product and process which must be managed during development to meet the quality requirements of the project (planning and quality control). Finally, these tools help assessing the compliance with project quality requirements (quality assessment).

The paper [14] is devoted to a systematic review of the various metaheuristic methods used to analyze various aspects of software quality, including error susceptibility, defect prediction, susceptibility to change, serviceability prediction, and software reliability prediction.

In the study [15], the authors used a hybrid method of quality prediction. Predicting is conducted with using the advanced neural network integrated with the Hybrid Cuckoo search (HCS) optimization algorithm for improving the accuracy of forecast. The neural network is improved by using HCS, which optimizes the weighting factor for improving the forecast. Quality indicators (maintainability and reliability) are evaluated for prediction of the software quality. The results are compared with other methods with the purpose of testing the effectiveness of the proposed method.

The authors [16] discuss statistical approaches to software quality forecasting based on data on software development processes using generalized linear modeling. In particular, Poisson, logistic and additional logarithmic regression models have been developed to predict the number of errors to be detected when testing the system.

Machine learning is also used for predicting the quality of the final product in the early stages of software life cycle. The machine learning-based predicting model uses software metrics and erroneous data from previous software projects for identification of the high-risk modules for future projects, then testing efforts will be focused on these specific high-risk modules. Thus, machine learning predictors facilitate the early and cost-effective detection of design anomalies and provide the timely delivery of the trouble-free, successful, high-quality, budget software product [17].

The document [18] presents an approach to software quality forecasting using the Software Requirements Specification (SRS). The SRS document is converted to a graph, and various parameters are extracted from it, including the readability index, complexity, size, and communication score. These parameters are transmitted to the fuzzy inference system (FIS) to predict the quality of the final product.

The paper [19] aims to compare different methods of data mining (for example, obtained by machine learning) to develop effective models for predicting software quality. To achieve this, the authors addressed various issues, such as collecting software metrics from open source repositories, evaluating forecasting models to identify software problems, and implementing statistical methods to evaluate data mining techniques.

The aim of the study [20] is to develop a basis for comprehensive software quality forecasting. This integration is reflected in a number of quality attributes included in the model, as well as the relationships between these attributes. The model is formulated as a Bayesian network. The structure allows to include expert knowledge about the subject area, as well as related empirical data and to encode them in the Bayesian network model. This model can be used to support decision-making by software analysts and managers.

Thus, currently, the models, methods, and tools for predicting the software quality in the early stages of the software development are imperfect. Therefore, predicting the software quality in the early stages of the software development is *the urgent task*. *The goal of this study* is the development of the neural-network model for software quality prediction based on quality attributes.

Neural-network model of software quality prediction based on quality attributes

For eliminating the problem of formal quality satisfaction, it is necessary to build a software quality model that will take into account the interaction of indicators and characteristics when determining the quality of software. Formalized set-theoretic software quality model based on the ISO 25010 standard was developed by one of the authors and presented in [2, 3].

Difficult-formalized tasks of software quality assessment are the determination of weights and the interaction of quality attributes within each software quality characteristic. Artificial neural networks (ANNs) are one of the means to summarize information and identify relationships between input and output data. Hecht-Nielsen's theorem also implies the possibility of representing a multidimensional function of several variables using a neural network. ANN provides considering the importance of each quality attribute in determining the software quality characteristics. The mutual influence of software quality attributes on software quality characteristics is also considered by ANN. So, for these reasons the ANN apparatus is selected for this task. Determining the effects of attributes on software quality characteristics was performed by one of the authors and described in [21].
So, it is then necessary to develop an ANN that will process a set of quality attributes, will approximate them, and will provide predictable quantitative estimates of software quality characteristics.
The concept of software quality prediction based on ANN is presented in Fig.1.

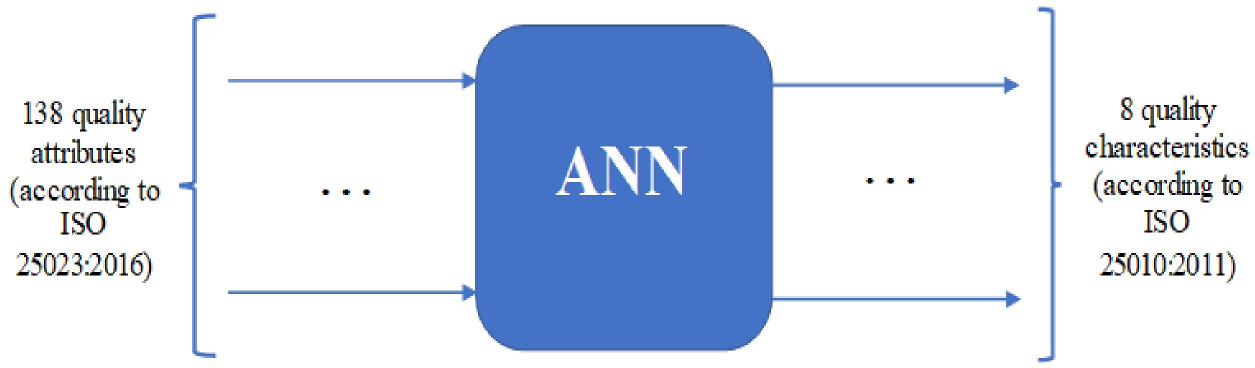
138 quality attributes (according to ISO 25023:2016)
. . .
ANN
. . .
8 quality characteristics (according to ISO 25010:2011)

Fig. 1. The concept of software quality prediction based on ANN

The choice of neural network structure and architecture is made in accordance with the characteristics and complexity of the problem. Optimal configurations already exist for certain types of tasks.

To select the architecture for predicting the characteristics of software quality, an analysis of known architectures of artificial neural networks was performed. Regression radial basis networks are used to analyze numerical series. Probabilistic radial basis networks are designed to solve probabilistic problems, in particular, classification problems. The Kohonen map is intended for data clustering. Networks for the classification of input vectors are used for clustering and classification. Elman and Hopfield networks are networks with dynamic inverse connections, focused on the development of dynamic models that take into account the prehistory of processes. Because the task of predicting the software quality characteristics doesn't have the properties of the numerical series, inverse relationships and needs for classification and clustering of data, let's choose a multilayer perceptron to solve this task. If we will use a different type of neural network for this task solving, its nature will be artificially distorted, resulting in inconsistent results of the ANN.

The structure of this ANN is presented in Fig. 2.

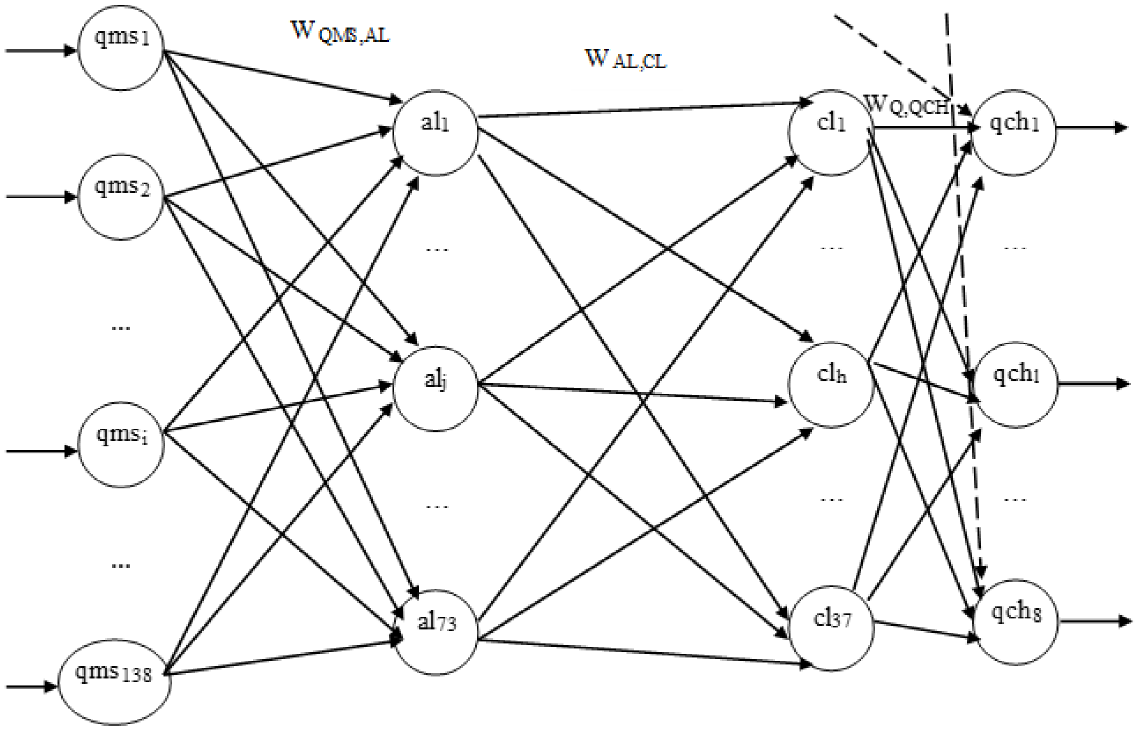

Fig. 2. Neural-network model of software quality prediction based on quality attributes

Neurons of the input (receptor) layer are defined by the set of software quality attributes $QMS = \{qms_i\},\ i = \overline{(1,\ 138)}$, where qms_i – i-th of the input (receptor) layer (i-th software quality attribute defined by ISO 25023), the number of input neurons (number of software quality attributes) is currently 138 (according to ISO 25023).

Neurons of the output (effector) layer define a set of software quality characteristics $QCH = \{qch_l\},\ l = \overline{(1,\ 8)}$, where qch_l – l-th functional of the effector layer of the multilayer perceptron (l-th software quality characteristic defined by ISO 25010), the number of output neurons (number of software quality characteristics) is currently 8 (according to ISO 25010).

Neurons of the first hidden (approximating) layer of the multilayer perceptron are the set $\overline{AL} = \{al_j\},\ j = \overline{(1, k_{al})}$, where al_j – j-th neuron of the first hidden (approximating) layer, k_{al} – the number of neurons of the first hidden (approximating) layer (for ANN with 138 input neurons and 8 output neurons the number of neurons in the first hidden layer $k_{al} = 73$ is enough).

Neurons of the second hidden (correcting) layer of the multilayer perceptron are defined by the set $\overline{CL} = \{cl_h\},\ h = \overline{(1, k_{cl})}$, where cl_h – h-th neuron of the second hidden (correcting) layer, k_{cl} – the number of neurons of the second hidden (correcting) layer (for ANN with 138 input neurons and 8 output neurons the number of neurons in the second hidden layer $k_{cl} = 37$ is enough).

The vector of threshold values of the offsets of the set of neural elements is defined as: $\overline{Q} = \{q_l\},\ l = \overline{(1,\ 8)}$, where q_l – the offset of the l-th neural element, $l = 8$ – the number of offsets of neural elements (the number of software quality characteristics according to ISO 25010).

The weights of the connections are represented by weight matrices: $\overline{W_{QMS,AL}} = \left\langle w_{qms_i,al_j} \right\rangle,\ i = \overline{(1,\ 138)}, j = \overline{(1, k_{al})}$, where w_{qms_i,al_j} – is the weighting coefficient of the connection between the qms_i-th input and al_j-th neuron of the first hidden layer; $\overline{W_{AL,CL}} = \left\langle w_{al_j,cl_h} \right\rangle,\ j = \overline{(1, k_{al})}, h = \overline{(1, k_{cl})}$, where w_{al_j,cl_h} – s the weighting coefficient of the connection between the al_j-th neuron of the approximating hidden layer and cl_h-th neuron of the correcting hidden layer; $\overline{W_{Q,QCH}} = \left\langle w_{q_l,qch_l} \right\rangle, \;\; l = (\overline{1,\ 8})$, where w_{q_l,qch_l} – is the weighting relationship between the q_l-th offset and qch_l-th neuron of the output layer.

The formula for determining the l-th functional of the effector layer of the ANN qch_l has the form:

$$qch_l = f_1\left(f_2(AL) \cdot \left(\sum_{j=1}^{k_{al}} \sum_{h=1}^{k_{cl}} (al_j \cdot w_{al_j,cl_k}) \right) + \sum_{i=1}^{e_{qch_l}} (qms_i \cdot w_{qms_i,al_i}) \right) - e_{qch_i} \cdot w_{q_i,qch_i} \qquad (1)$$

where f_1 – activation function of neurons of the effector layer of ANN (linear function, the results of which lie in the interval [0; 1]), f_2 – activation function of neurons of hidden layers (hyperbolic tangent), e_{qch_l} – number of software quality attributes that affect the qch_l-th ANN output (the qch_l-th predicted quality characteristic of the software), qms_i – i-th ANN input, which affects the qch_l-th ANN output.

The first time proposed neural-network model of software quality prediction based on quality attributes differs from the known models in that it allows to take into account the importance of each quality attribute, as well as the interaction of attributes within each software quality characteristic. The original functionals of ANN corresponding to the values of software quality characteristics (functional suitability, performance efficiency, usability, reliability, compatibility, security, maintainability, portability), allow assessing the total impact of quality attributes on software quality characteristics.

Conclusions

Software quality (software) has become an increasingly important topic over the last decade due to the rapid growth of the software market, increasing competition in the software industry, and the increasing complexity and responsibility of software. The urgency of software quality improvement issues necessitates software quality forecasting in the early stages of the life cycle and the development of models, methods, and tools for software quality forecasting based on attributes.

Universal structures for summarizing the information and identifying the relationships between input and output data are artificial neural networks (ANNs). Thereby the ANN was used for implementing the functions that considering the interaction of characteristics and quality attributes.

The authors propose the ANN-based concept of software quality prediction, which takes into account the interaction of quality attributes during determining the software quality characteristics.

The first time proposed neural-network model of software quality prediction based on quality attributes differs from the known models in that it allows to take into account the importance of each quality attribute, as well as the interaction of attributes within each software quality characteristic. The original functionals of ANN corresponding to the values of software quality characteristics (functional suitability, performance efficiency, usability, reliability, compatibility, security, maintainability, portability), allow assessing the total impact of quality attributes on software quality characteristics.

Given the proposed concept and model of software quality prediction based on quality attributes, promising tasks for further research are: development of a neural-network method for predicting the level of software quality based on quality attributes; implementation and training of the neural network component of the proposed method; design, development, and implementation of software quality forecasting technology based on quality attributes in the early stages of the life cycle.

References

1. W. E. Wong, V. Debroy, A. Surampudi, H. Kim, M. F. Siok. Recent Catastrophic Accidents: Investigating How Software was Responsible. Fourth International Conference on Secure Software Integration and Reliability Improvement: Proceedings (Singapore, June 9-11, 2010), Singapore, 2010. Pp. 14-22.

2. T. Hovorushchenko. Methodology of Evaluating the Sufficiency of Information for Software Quality Assessment According to ISO 25010. Journal of Information and Organizational Sciences. 2018. Vol. 42. No.1. Pp. 63-85.

3. T. Hovorushchenko. Models and Methods of Evaluation of Information Sufficiency for Determining the Software Complexity and Quality Based on the Metric Analysis Results. Central European Researchers Journal. 2016. Vol. 2. Issue 2. Pp. 42-53.

4. ISO/IEC 25010:2011. Systems and software engineering. Systems and software Quality Requirements and Evaluation (SQuaRE). System and software quality models. [Introduced 01.03.2011]. Geneva (Switzerland), 2011. 34 p. (International standard).

5. ISO 25023:2016. Systems and software engineering. Systems and software Quality Requirements and Evaluation (SQuaRE). Measurement of system and software product quality. [Introduced 31.03.2016]. Geneva (Switzerland), 2016. 45 p. (International standard).

6. T. Sun, X. Lv, Y. Cai, Y. Pan, J. Huang. Software test quality evaluation based on fuzzy mathematics. Journal of intelligent & fuzzy systems. 2021. Vol. 40. Issue 4. Pp. 6125-6135.

7. S. Huang, W. Chen, P. Chiu. Evaluation Process Model of the Software Product Quality Levels. IEEE 2015 International Conference on Industrial Informatics - Computing Technology, Intelligent Technology, Industrial Information Integration (ICIICII): Proceedings (Wuhan (China), December 03-04, 2015). Wuhan, 2015. Pp. 55-58.

8. W. Pai, C. Wang, D. Jiang. A software development model based on quality measurement. Computer applications in industry and engineering. 2000. Pp. 40-43.

9. S. Cho, S. Yoo. A Quality Evaluation Model for Hardware-Control Software. Advanced science letters. 2017. Vol. 23. Issue 10. Pp. 9607-9611.

10. F. Xing, P. Guo, M. Lyu. A novel method for early software quality prediction based on support vector machine. 16th IEEE International Symposium on Software Reliability Engineering: Proceedings (Chicago, November 08-11, 2005). Chicago, 2005. Pp. 213-222.

11. A. AL-Badareen, Z. Muda, M. Jabar, J. Din, S. Turaev. Software Quality Evaluation through Maintenance Processes. Advances in communications, computers, systems, circuits and devices. 2010. Pp. 131.

12. Y. Alsultanny, A. Wohaishi. Requirements of Software Quality Assurance Model. The Second International Conference on Environmental And Computer Science: Proceedings (Dubai, December 28-30, 2009). Dubai, 2009. Pp. 19-23.

13. B. Kitchenham, A. Pasquini, U. Anders, J. Boegh, S. DePanfilis, S. Linkman. Automating software quality modelling, measurement and assessment. Reliability, quality and safety of software-intensive systems. 2007. Pp. 43-53.

14. K. Lakra, A. Chug. Application of metaheuristic techniques in software quality prediction: a systematic mapping study. International Journal of intelligent engineering informatics. 2021. Vol. 9. Issue 4. Pp. 355-399.

15. K. Sheoran, P. Tomar, R. Mishra. Software Quality Prediction Model with the Aid of Advanced Neural Network with HCS. 2nd International Conference on Intelligent Computing, Communication & Convergence: Proceedings (Bhubaneswar (India), January 24-25, 2016). Bhubaneswar, 2016. Vol. 92. Pp. 418-424.

16. S. Inoue, S. Yamada. Statistical Prediction of Software Quality Based on Generalized Linear Models. OF The 13th International Conference on Industrial Management: Proceedings (Hiroshima (Japan), September 21-23, 2016). Hiroshima,2016. Pp. 403-408.

17. S. Goyal. Comparison of Machine Learning Techniques for Software Quality Prediction. International Journal of knowledge and systems science. 2020. Vol. 11. Issue 2. Pp. 20-40.

18. M. Masood, M. Khan. Early Software Quality Prediction Based on Software Requirements Specification Using Fuzzy Inference System. Lecture Notes in Artificial Intelligence. 2018. Vol. 10956. Pp. 722-733.

19. M. Canaparo, E. Ronchieri. Data Mining Techniques for Software Quality Prediction in Open Source Software: An Initial Assessment. EPJ Web of Conferences. 2014. Article Number 05007.

20. L. Radlinski. A Framework for Integrated Software Quality Prediction Using Bayesian Nets. Lecture Notes in Computer Science. 2011. Vol. 6786. Pp. 310-325.

21. T. Hovorushchenko, O. Pomorova. Evaluation of Mutual Influences of Software Quality Characteristics Based ISO 25010:2011. 11-th International Conference on Computer Science and Information Technologies: Proceedings (Lviv, September 06-10, 2016). Lviv, 2016. Pp. 80-83.

Mykyta Lebiga **Микита Лебіга**	PhD Student of Computer Engineering & Information Systems Department, Khmenlnytskyi National University https://orcid.org/0000-0002-5849-1514 e-mail: lebiganikita1996@gmail.com	аспірант кафедри комп'ютерної інженерії та інформаційних систем, Хмельницький національний університет
Tetiana Hovorushchenko **Тетяна Говорущенко**	DrSc (Engineering), Professor, Head of Computer Engineering & Information Systems Department, Khmenlnytskyi National University https://orcid.org/0000-0002-7942-1857 e-mail: govorushchenko@gmail.com	доктор технічних наук, професор, завідувач кафедри комп'ютерної інженерії та інформаційних систем, Хмельницький національний університет
Mariia Kapustian **Марія Капустян**	PhD (Engineering), Associate Professor, Associate Professor of the Computer Engineering & Information Systems Department, Khmenlnytskyi National University https://orcid.org/0000-0001-9200-1622 e-mail: kapustian.mariia@gmail.com	кандидат технічних наук, доцент, доцент кафедри комп'ютерної інженерії та інформаційних систем, Хмельницький національний університет