



HAL
open science

The Generalization of Selection of an Appropriate Artificial Neural Network to Assess the Effort and Costs of Software Projects

Dragica Rankovic, Nevena Rankovic, Mirjana Ivanovic, Ljubomir Lazic

► To cite this version:

Dragica Rankovic, Nevena Rankovic, Mirjana Ivanovic, Ljubomir Lazic. The Generalization of Selection of an Appropriate Artificial Neural Network to Assess the Effort and Costs of Software Projects. 18th IFIP International Conference on Artificial Intelligence Applications and Innovations (AIAI), Jun 2022, Hersonissos, Greece. pp.420-431, 10.1007/978-3-031-08333-4_34 . hal-04317170

HAL Id: hal-04317170

<https://inria.hal.science/hal-04317170v1>

Submitted on 1 Dec 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License



This document is the original author manuscript of a paper submitted to an IFIP conference proceedings or other IFIP publication by Springer Nature. As such, there may be some differences in the official published version of the paper. Such differences, if any, are usually due to reformatting during preparation for publication or minor corrections made by the author(s) during final proofreading of the publication manuscript.

The generalization of selection of an appropriate Artificial Neural Network to assess the effort and costs of software projects

Dragica Rankovic¹, Nevena Rankovic², Mirjana Ivanovic³ and Ljubomir Lazic⁴

¹ University Union “Nikola Tesla”, Faculty of Applied Sciences Nis, Department of mathematics, 18000 Nis, Serbia

² Union University, School of Computing Belgrade, Department of Computer Science, 11000 Belgrade, Serbia

³ University of Novi Sad, Faculty of Sciences, Department of mathematics and informatics, 21000 Novi Sad, Serbia

⁴ Union University, School of Computing Belgrade, Department of Computer Engineering, 11000 Belgrade, Serbia
nrankovic@raf.rs

Abstract. Estimation of effort and costs is crucial for successful implementation of software projects. Project development time is an essential factor, both for project clients and project developers. The amount of money needed to invest in a project influences the decision whether to start a project or not or whether it will be completed successfully or not. In practice, the cost of a project is most often compared to the cost of similar projects, which have been successfully completed. The article proposes combining the experimental information of the Taguchi method with ANN (*Artificial Neural Network*) learning, and constructs a progressive Taguchi neural network model to decrease the number of experimental runs and time required. The main goal of this paper is to make generalization of the conditions and criteria for successful project realization based on a large number of experimental results that authors obtained so far. Furthermore, the most reliable and efficient artificial intelligence model for effort and cost estimation is identified.

Keywords: software estimation, ANN, Taguchi method, generalization;

1 Introduction

Three approaches have been most commonly used in software project estimations: COCOMO2000 (*Constructive Cost model*), parametric model [1], [2], [3], COSMIC FFP (*Function Point Analysis*) model based on functional point analysis [4], [5], [6] and the UCP (*Use Case Point Analysis*) method, based on the analysis of users and use cases [7], [8]. For each of these three approaches, which are based on different effort and cost estimation methods, a new, improved model was selected and constructed. Previously used models of the proposed approaches did not give good enough results to improve projects' success significantly. Three new, improved models were constructed using different ANN architectures based on different Taguchi's orthogonal vector plans [9], [10]. ANNs are an excellent artificial intelligence tool for information

processing and can significantly contribute to the construction of new, improved software evaluation models. Due to ANN's ability to learn from different data sets (observations), it is possible to generate accurate and reliable results to avoid unforeseen situations. Different ANN architectures are used to identify the simplest one that meets the additionally set of criteria. Each of these enhanced approaches includes

- constructing and identifying the best model for estimating effort and cost,
- selection of the best ANN architecture whose values converge the fastest to the minimum magnitude relative error,
- use of publicly available real relevant datasets according to the input values of the proposed model [2], [8], [9], and [10],
- achieving the minimum number of performed experiments, reduced software effort estimation time due to convergence rate.

Additional criteria and constraints are introduced to monitor and execute experiments using a precise algorithm to execute all three new proposed models. In addition to monitoring the convergence rate of each ANN architecture, the influence of the input quantities of each model on the change in the value of the mean magnitude relative error (MMRE) of the model is monitored. The models constructed in this way have been experimentally tested and their positive results have been confirmed several times on different sets of real industrial projects and can be successfully applied in practice. Furthermore, obtained results indicate that the achieved error values are lower than these obtained in the previous models. Therefore, the proposed models in this paper can be reliably applied and used to estimate efforts and costs for software development and the development of projects in other areas of industry and science [11], [12].

The rest of the paper is organized as follows. The second section is devoted to related work. Selection of the appropriate ANN architecture based on Taguchi's orthogonal vector plan is discussed in the third section. The fourth section presents clustering method based on different ANN architectures. The section five is devoted to convergence rate and determination of optimal number of iterations. Selection of activation function and encoding/decoding method are presented in section six while last section concludes the paper.

2 Related work

Preliminary estimates of efforts can usually lead to delays in project implementation, requests for additional funds, overtime work of experts, and the other problems. Also, they can directly affect the quality of the software. Due to not considering all the necessary parameters for assessing the implementation of software, it often happens that individual activities, such as additional testing, completion of documentation, and additional definition of user requirements, are reduced to a minimum of effort. All this can lead to many failed projects that are currently frequent situation in the software industry. Previous methods of estimating efforts and costs during the implementation of software projects were based on unreliable and inaccurate methods, techniques, and models. The result of such incomplete assessments is many unsuccessful and not completed projects. All research indicates that only about 30% of projects are successfully completed, [13], [14], while the remaining percentage of projects remain unresolved or

ultimately fail. The most commonly used methods so far are similarity method, method of analysis and synthesis, assessment based on expert knowledge, and various parametric methods. In addition to various assessment methods, software companies use various auxiliary software tools and services to meet customer requirements. Many researchers [1, 4, 5, 6, 11, 12] have proposed different combinations of parametric and non-parametric effort and cost estimation models to construct reliable software of high standards and performance. It is necessary to analyze and experimentally test the most successful methods and models so far in order to adequately improve them and support more successful realization of software projects [15], [16].

3 Selection of the appropriate ANN architecture based on Taguchi's orthogonal vector plan

COCOMO2000 is a parametric model that calculates the size of a system as a combination of mathematical models. Basic input data are three parameters obtained experimentally by measuring actual values during project development. Measurement-based on the number of lines of source code is used to determine the size and complexity of the software project. It is most often used to determine the effort and time needed to implement a project. The new, improved COCOMO2000 model uses four different ANN architectures based on different Taguchi orthogonal vector plans. Artificial neural networks (ANNs) are recognized for their ability to provide good results when dealing with problems where there are complex relationships between inputs and outputs, and where the input data is distorted by high noise levels. The performance of neural networks depends on the architecture of the network and their parameter settings. Determining the architecture of a network (size, structure, connectivity) affects the performance criteria, such as the learning speed, accuracy of learning, noise resistance and generalization ability.

There is no clearly defined theory which allows for the calculation of the ideal parameter settings and as a rule even slight parameter changes can cause major variations in the behavior of almost all networks. This paper describes an innovative application of the Taguchi method for the determination of these parameters to meet the training speed and accuracy requirements. Using the Taguchi method, both the micro-structural and macro-structural aspects of the neural network design parameters can be considered concurrently. The most straightforward architecture, ANN-L9 (Fig. 1), is based on the orthogonal plan L9 on three levels and four weighting factors. More complex ANN-L18 (Fig. 1) is based on the orthogonal plan L18 on two and three levels and eight weighting factors. Slightly more complex ANN-L27 (Fig. 2) is based on the orthogonal plan L27 on three levels with thirteen weighting coefficients. The most complex ANN-L36 (Fig. 2) architecture used in this model is based on the L36 orthogonal plan at two and three levels and twenty-three weighting coefficients [17], [18], [19].

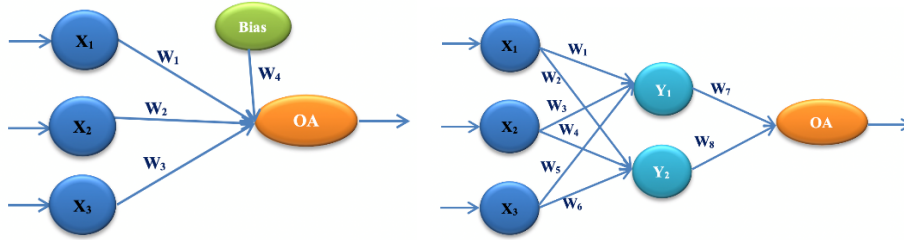


Fig 1. ANN architecture with none hidden layer (ANN-L9) and one hidden layer (ANN-L18).

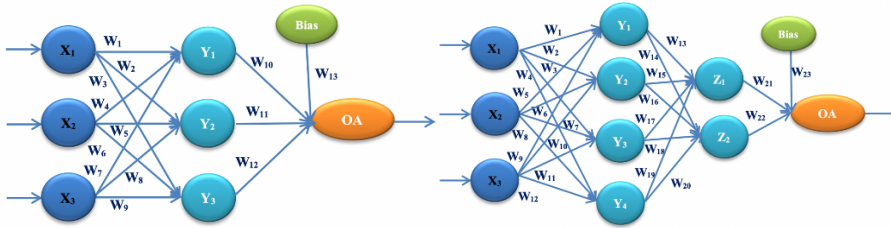


Fig 2. ANN architecture with one hidden layer (ANN-L27) and two hidden layers (ANN-L36).

Function point analysis is an approach that was created in order to overcome the shortcomings of the previous approach, which is founded on measuring the size of the system based on the number of lines of code. In this approach, the functionality of the system is measured on the basis of quantities, expressed in functional points. Different systems may have similar functionality, but may use different technologies or programming languages, so they differ in the number of source lines of code. A number-based approach has developed a large number of models to most effectively and accurately estimate functional size. This paper will present the recent method from the family of functional points, COSMIC FFP. This method is used in the process of estimating the effort and cost of the functional size of software projects and is based on fourteen parameters that are reduced to four input sizes. In previous functional point methods, five input quantities were used. The new, improved COSMIC FFP model used two different ANN architectures, based on different Taguchi's orthogonal vector plans. The simplest architecture ANN-L12 (Fig. 3), is based on the orthogonal plan L12 on two levels and eleven weighting coefficients. More complex ANN-L36prim, is based on the orthogonal plan L36prim (Fig. 3) at two and three levels and sixteen weighting coefficients [20], [21], [22].

UCP is the latest and the most commonly used method for estimating the effort and costs for the realization of software products. Although it is not standardized within ISO standards such as the COCOMO2000 and COSMIC FFP methods, it results in estimation errors between 20% and 35%. The best result achieved with this method is an error value of about 10%. System users and use cases are used to determine the functional size by the UCP method. The new, improved UCP model used two different ANN architectures based on different Taguchi orthogonal vector plans. The most

straightforward architecture, ANN-L16 (Fig. 4), is based on the orthogonal plan L16 on two levels and fifteen weighting coefficients. The more complex ANN-L36prim (Fig. 4) is based on the orthogonal plan L36prim at two and three levels and sixteen weighting coefficients [23], [24].

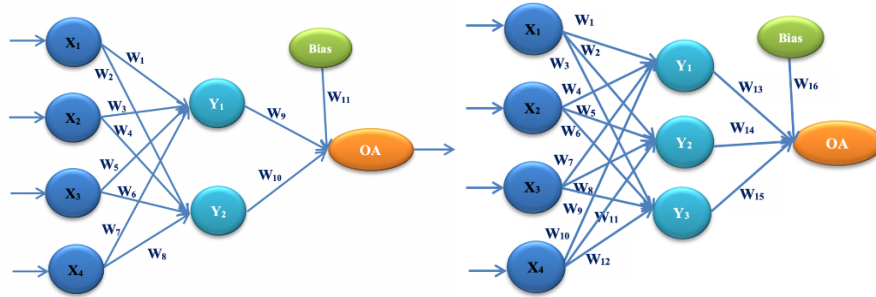


Fig 3. ANN architecture with one hidden layer (ANN-L12) and one hidden layer (ANN-L36prim).

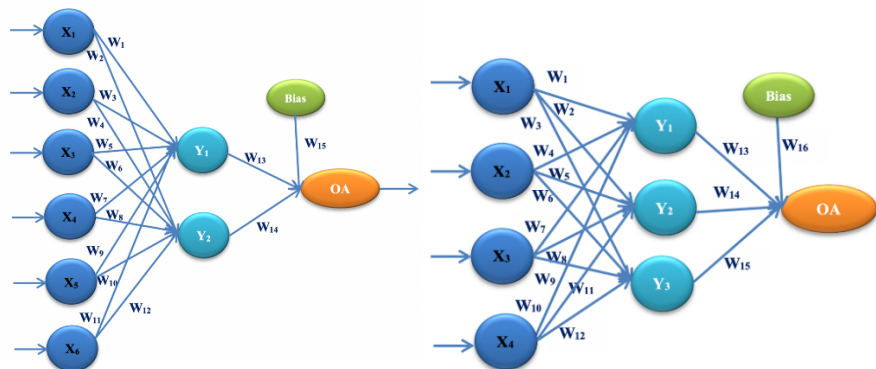


Fig 4. ANN architecture with one hidden layer (ANN-L16) and one hidden layer (ANN-L36prim).

In addition to different architectures, the clustering method significantly contributes to controlling the different values of actual projects from practice, which aims to realistically estimate and reduce the minimum relative error of the model. The activation function used is the sigmoid function of the hidden and output layers. This function enables additional homogenization of input quantities and contributes to the rate of convergence of each of the listed ANN architectures. Experiments have shown that the number of iterations in each ANN architecture is less than 10, significantly shortening the estimation time. The new proposed models' reliability, accuracy, and efficiency have been repeatedly tested and validated on several different data sets and by verification with statistical analysis methods. Based on the obtained estimated value, various metrics were calculated, such as MAE (Magnitude Absolute Error), MRE (Magnitude Relative Error), MMRE (Mean Magnitude Relative Error) [2], [8], [9],[10]

4 Clustering method using different ANN architectures

One of the main processes in the design of neural networks is to systematically select the design parameters at both the micro-structural and macro-structural levels. Technical experience together with experiments, through computer simulations, are needed to establish the most suitable values for these design parameters. A common approach to establish these values is through the trial and error method. The Taguchi method provides a mathematical tool called the orthogonal arrays which allows the analysis of the relationships between a large number of design parameters within the smallest number of possible experiments. Noise factors including the initial values of connection weights and the values of input data are considered as external to the system and as such they cannot be controlled at all. The Taguchi method provides a means to design a neural network that is insensitive to these noise factors by setting of the design factors, such as the number of hidden layers and hidden neurons, through a set of systematic experiments. Let W be the total number of weights in the network, then in the process of training ANN at least $2W$ observations in the training data set are required. Depending on the proposed approaches, the training process on different ANN architectures also involves using specific data sets. The choice of appropriate datasets depends on the model and size of the input and publicly available databases. A COCOMO2000 data set was used for the ANN training procedure in the first proposed approach. The same data set was used for the ANN testing process in the other projects, and the validation process used the data sets COCOMO81, NASA60, and Kemerer. The number of projects should be greater than the number of weighting factors of the proposed ANN architecture on each selected data set. For the ANN-L9, ANN-L18, and ANN-L27 architectures, the minimum project requirement is met for all datasets used. In the ANN-L36 architecture, which has 23 weighting coefficients, the number of testing projects was 20, so the required condition was not met (Dataset_2). In the ANN-L36 architecture, which has 23 weighting coefficients, the number of projects for the third validation was 15, so the required condition was not met (Dataset_5). It can be concluded that in all parts of the experiment within the first proposed approach the fulfillment of the conditions is 90% (see Table 1).

Table 1. The number of projects greater than the number of weighting factors (COCOMO2000).

Datasets		No.of projects	Experiment	Weighting factors (coefficients)			
				ANN-L9(4)	ANN-L18(8)	ANN-L27(13)	ANN-L36(23)
Dataset_1	COCOMO 2000	10 0	Training	+	+	+	+
Dataset_2	COCOMO 2000	20	Testing	+	+	+	/
Dataset_3	COCOMO81	51	Validation1	+	+	+	+
Dataset_4	NASA	60	Validation2	+	+	+	+
Dataset_5	Kemerer	15	Validation3	+	+	+	/

The ISBSG dataset divided into five selected clusters was used for the ANN training process for the second proposed approach. The same data set in the five selected clusters were used for the ANN testing procedure. The number of projects in both

procedures on each selected cluster is 70:30; 70 projects were used for training and 30 for testing. Desharnais data set and data set combined from different, realistic industrial projects were used for the validation process. The number of projects should be greater than the number of weighting factors of the proposed ANN architecture on each selected data set. In the ANN-L12 architecture, which has 11 weighting coefficients, only the number of testing projects was 7, so the required condition for one data set was not met (Dataset_5). With the ANN-L36prim architecture, which has 16 weighting coefficients, the number of testing projects is 15, so the set condition is not met (Dataset_1). There were 13 testing projects (Dataset_3), so the required condition was not met within the third cluster. The number of training and testing projects (Dataset_5) was 14 and 7 projects, respectively. Within the Desharnais data set (Dataset_6), the number of projects was 14, and the set condition was not met. The number of data sets that met the criteria was 7 out of 12 [3]. It can be concluded that in all parts of the experiment, according to the second proposed approach, the fulfillment of the conditions is 75% (see Table 2).

Table 2. The number of projects greater than the number of weighting factors (COSMIC FFP).

Datasets		Number of project	Experiment	Weighting factors (coefficients)	
				ANN-L12 (11)	ANN-L36prim (16)
Dataset_1	ISBSG (Functional Size<10)	37	Training	+	+
	ISBSG (Functional Size<10)	15	Testing	+	/
Dataset_2	ISBSG (10<Functional Size<50)	45	Training	+	+
	ISBSG (10<Functional Size<50)	17	Testing	+	+
Dataset_3	ISBSG (50<Functional Size<100)	30	Training	+	+
	ISBSG (10<Functional Size<100)	13	Testing	+	/
Dataset_4	ISBSG (100<Functional Size<500)	60	Training	+	+
	ISBSG (10<Functional Size<500)	17	Testing	+	+
Datset_5	ISBSG (Functional Size>500)	14	Training	+	/
	ISBSG (Functional Size>500)	7	Testing	/	/

Dataset_6	Desharnais	14	Validation1	+	/
Dataset_7	Combined	33	Validation2	+	+

For the ANN training process, the third proposed approach used the Benchmark (Mendeley) data set divided into a 70:30 scale, 70% of the projects for the training process and 30% for the testing process. Combined real, industrial projects (Dataset_3, Dataset_4) were used for the validation procedure. The number of projects should be greater than the number of weighting factors of the proposed ANN architecture on each selected data set. In the ANN-L16 architecture, which has 15 weighting factors, the condition is that the number of projects for all parts of the experiments is greater than the number of weighting factors. With the ANN-L36prim architecture, which has 16 weighting coefficients, the condition is met that the number of projects for all parts of the experiment is greater than the number of weighting coefficients [2]. It can be concluded that in all parts of the experiment within the third proposed approach, the fulfillment of the conditions is 100%, (see Table 3).

Table 3. The number of projects greater than the number of weighting factors (UCP).

Dataset		Number of projects	Experiment	Weighting factors (coefficients)	
				ANN-L16(15)	ANN-L36prim(16)
Dataset_1	UCP Benchmark Dataset	50	Training	+	+
Dataset_2	UCP Benchmark Dataset	21	Testing	+	+
Dataset_3	Combined	18	Validation1	+	+
Dataset_4	Combined Industrial projects	17	Validation2	+	+

5 Convergence rate and number of iterations performed

One of the goals of the new, improved models is to achieve the minimum number of iterations performed to reduce the time required for estimation. For each of the four listed ANN architectures. Table 4. shows the number of iterations that need to be performed to meet the set value for the GA (Gradient Descent) criterion. The minimum number of iterations required is for ANN-L27 and ANN-L36 architectures for small and medium clusters. The most significant number of iterations should be performed for the ANN-L18 architecture, and it is nine iterations. It can be concluded that the number of iterations required is minimal (equal to nine), which leads to a quick estimate using the COCOMO2000 model.

Table 4. Number of iterations performed for each ANN architecture - COCOMO2000

COCOMO2000 model												
ANN	ANN-L9			ANN-L18			ANN-L27			ANN-L36		
No. of Iterations	S	M	L	S	M	L	S	M	L	S	M	L
GA<0.01	7	6	8	7	6	9	5	5	8	5	5	7

*(S-Small cluster, M-Medium cluster, L-Large cluster)

The number of iterations required in the second COSMIC FFP approach for both proposed architectures is six, except for the ANN-L12 architecture, where there are five for Dataset_1 and Dataset_3. It can be concluded that the number of required iterations is minimal (equal to six), which leads to an even faster estimate using the COSMIC FFP model (see Table 5).

Table 5. Number of iterations performed for each ANN architecture - COSMIC FFP.

COSMIC FFP model										
ANN	ANN-L12					ANN-L36prim				
No. of Iteration	D_1	D_2	D_3	D_4	D_5	D_1	D_2	D_3	D_4	D_5
GA<0.01	5	6	5	6	6	6	6	6	6	6

*(D_1-Dataset_1, D_2 -Dataset_2, D_3-Dataset_3, D_4-Dataset_4, D_5-Dataset_5)

Four iterations are required to meet the set GA criterion in the first proposed ANN-L16 architecture for the UCP model. For the ANN-L36prim architecture, the number of iterations is equal to six. It can be concluded that the number of iterations required is the lowest compared to the other two models (equal to six), which leads to an even faster estimate using the UCP model (see Table 6).

Table 6. Number of iterations performed for each ANN architecture - UCP.

UCP model		
No. of Iteration	ANN-L16	ANN-L36prim
GA<0.01	Training	Training
	4	6

In all three proposed approaches, the minimum number of iterations required to meet the GA criteria is 4 in the UCP model for the ANN-L16 architecture. It can be concluded that this architecture converges the fastest to the minimum value of MMRE [2], [8], [9].

6 Selection of activation function and encoding/decoding method

During our research, for all experiments of all three proposed approaches, different activation functions in the hidden and output layers were used. The experiments were performed with different functions: sigmoid function, hyperbolic tangent, Gaussian function, and others. The sigmoid function gave the best results in all experiments, i.e., the lowest MMRE value, and was used in all three proposed improved models [1], [6]. Different coding methods were used to homogenize different nature of input values: fusion method, logarithmic method, combined method, etc. The best results were achieved in all experiments with the fuzzification method used in all three proposed models [2], [8], [9].

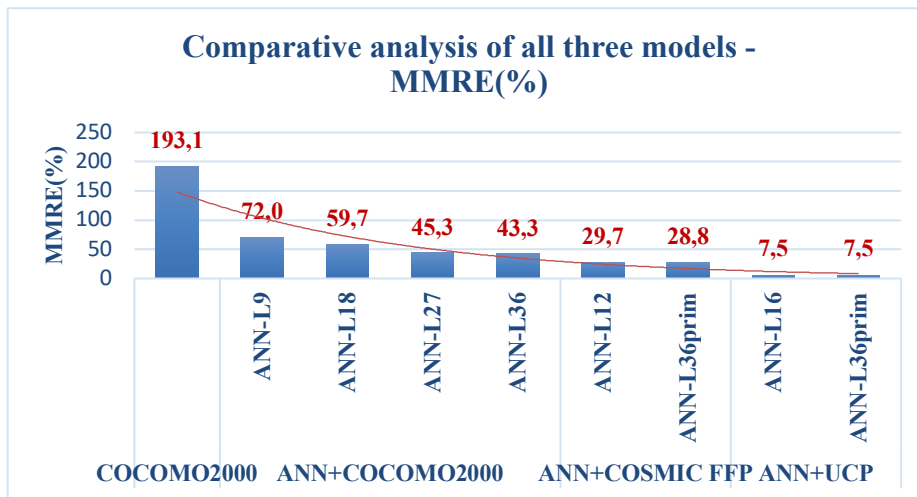


Fig 5. MMRE values in all approaches.

By comparing the obtained results using all three new proposed improved models, it can be concluded:

- The most enormous value of the error in estimating effort and cost is in the parametric COCOMO2000 methods and goes to 193.1%;
- The best result, the lowest value of MMRE in the first proposed model is 43.3%, which is 4.5 times better result compared to the parametric method;
- In the second proposed model, the error on the ANN-L36prim architecture is 28.8% which is 6.7 times better result compared to the parametric method;
- In the third model, which is the best result in all performed experiments, the MMRE value for both proposed ANN architectures is 7.5%, which is 25.7 times better than parametric methods.

Comparing the models with each other we come to the following conclusions,

- The second proposed model ANN + COSMIC FFP has 1.6 times less error than the first model ANN + COCOMO2000;

- The third proposed ANN + UCP model has a 5.8 times lower error than the first proposed model ANN + COCOMO2000;
- The third proposed ANN + UCP model has a 3.8 times lower MMRE value than the second proposed model ANN + COSMIC FFP [2], [8], [9].

7 Conclusion

Depending on the size of the project for which the effort and costs need to be assessed, the number of inputs, and other specific factors, all three new, improved models can be used. After several rounds of experiments and analysis of the obtained results, it can be concluded that the third proposed model gives the lowest value of relative error, and it is 7.5%. If we consider that the minimum number of iterations required to reach the minimum MMRE is only four, the best ANN architecture can be identified, and that is ANN-L16 within the UCP model. The analysis of effort and cost estimates presented in this paper gives better results than the previous ones and shows that better, more advanced models can be constructed. The presented applied scientific methods can help in improving this area of software engineering. The new methodology does not exclude the possibility of applying a subjective assessment of labor and costs but can help the project team, pointing out potential problems due to discrepancies in different methods. In this way, teams can perform additional analysis to better assess and correct the results in the shortest possible time. Otherwise, project teams would have to share their own risk if only their subjective opinion is taken into account. The results obtained can help project teams, software engineers, and test engineers to make short-term plans with high and long-term reliability. Therefore, teams can with great confidence realize all phases of analysis and design and complete the project on time. If the additional estimation is needed, it is possible to anticipate the effort and costs that each project task is realized in a quality and optimal way.

References

1. A. Stoica and J. Blosiu. Neural learning using Orthogonal arrays. *Advances in Intelligent Systems*, 41: 418, 1997.
2. D. Rankovic et al. Convergence rate of Artificial Neural Networks in software development projects. *Information and Software Technology Journal*, 138 (10), 2021.
3. B. W. Boehm et al. *Software cost estimation with COCOMO II*. First Edition, Prentice Hall Press, 2009.
4. J. Popović. Enhancing methods for effort estimation in software projects. Doctoral dissertation, University of Belgrade, School of Electrical Engineering,
5. R. Meli et al. On the applicability of COSMIC-FFP for measuring software throughout its life cycle. In *Proceedings of the 11th European Software Control and Metrics Conference*, pages 18-20, Springer, 2000.
6. M. A. Shah. Ensembling artificial bee colony with analogy-based estimation to improve software development effort prediction. *IEEE Access*, 8: 58402-58415, 2020.

7. A. Kaur and K. Kaur. Effort estimation for mobile applications using use case point (UCP). In *Smart Innovations in Communication and Computational Sciences*, pages 163-172, Springer, Singapore, 2019.
8. N. Rankovic et al. Improved effort and cost estimation model using Artificial Neural Networks and Taguchi method with different activation functions. *Entropy*, 23 (7), 854, 2021.
9. N. Rankovic. et al. A new approach to software effort estimation using different Artificial Neural Network architectures and Taguchi Orthogonal Arrays. *IEEE Access*, 9: 26926-26936, 2021.
10. N. Rankovic, D. Rankovic, M. Ivanovic, & L. Lazic, (2021). A Novel UCP Model Based on Artificial Neural Networks and Orthogonal Arrays. *Applied Sciences*, 11(19), 8799.
11. J. F. Khaw, B. Lim, and L. E.Lim. Optimal design of neural networks using the taguchi method. *Neurocomputing*, 7 (3): 225–245, 1995.
12. A. BaniMustafa. Predicting software effort estimation using machine learning techniques. In *8th International Conference on Computer Science and Information Technology (CSIT)*, pages 249–256, IEEE, 2018.
13. *A Guide to the Project Management Body of Knowledge (PMBOK Guide)*. Third Edition, Project Management Institute, Inc. 2004. ISBN: 1-930699-45-X.
14. M. Ayat et al. Current trends analysis and prioritization of success factors: a systematic literature review of ICT projects. *International Journal of Managing Projects in Business*, 14 (3): 652-679, 2021.
15. P. S. Kumar et al. Advancement from neural networks to deep learning in software effort estimation: Perspective of two decades. *Computer Science Review*, 28 (11):100288, 2020.
16. P. S. Kumar and H. Behera. Estimating software effort using neural network: An experimental investigation. *Computational Intelligence in Pattern Recognition*, Proceedings of CIPR, pages 165–180, Springer, 2020.
17. M. H. Sazli. A brief review of feed-forward neural networks. *Communications Faculty of Sciences University of Ankara Series A2-A3 Physical Sciences and Engineering*, 50 (01), 2006.
18. A. Jeang and C. L. Chang. Combined robust parameter and tolerance design using Orthogonal arrays. *The International Journal of Advanced Manufacturing Technology*, 19 (6), 442-447, 2002.
19. J. D. Kechagias et al. A comparative investigation of Taguchi and full factorial design for machinability prediction in turning of a titanium alloy. *Measurement*, 151: 107213, 2020.
20. M. Kassab, C. Neill, and P. Laplante. State of practice in requirements engineering: contemporary data. *Innovations in Systems and Software Engineering*, 10:(4): 235-241, 2014.
21. E. R. Carroll. Estimating software based on use case points. *Companion to the 20th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications*, pages 257-265, San Diego, CA, USA, 2005.
22. A.B. Nassif, L.F. Capretz, and D. Ho. Enhancing Use Case Points Estimation Method using Soft Computing Techniques. *Journal of Global Research in Computer Science*, 1 (4): 12-21, 2010.
23. A.B. Nassif. *Software Size and Effort Estimation from Use Case Diagrams Using Regression and Soft Computing Models*. Doctoral dissertation, Western University, London, Ontario, Canada, 2012.
24. M. Azzeh. Fuzzy Model Tree for Early Effort Estimation Machine Learning and Applications. *12th International Conference on Machine Learning and Applications*, pages 117-121, Miami, Florida, USA, IEEE, 2013.