Journal of Advanced Computer Science and Technology Research, Vol.7 No.2, June 2017, 42-56

The Application of Artificial Neural Networks in Software Effort Estimation

Maryam Mirzapour Moshizi, Vahid Khatibi Bardsiri

Department of Computer Science, Bardsir Branch, Islamic Azad University, Kerman, Iran

mmairr_2007@yahoo.com

ISSN: 2231-8852

ABSTRACT

Estimate the cost and time required to build the software system is one of the most important aspects of software project management. Estimation or effort required to develop a software system is one of the major concerns of the project manager. Activity and time required to complete the process and the cost of each process will be studied in the context of estimation. There have been a lot of patterns for effort and cost estimates up to now. This is beneficial and necessary that in the early stages of construction with a minimum of information of the project, estimate effort and cost of development of a software system. Learning methods, such as neural networks, is one of these models. In this paper, the software effort estimation by the neural network has been studied and their assessment criteria have been compared.

Keywords: Software, Estimation, Cost, Effort, Artificial Neural Networks

1. Introduction

In the past, the cost of software included a small percentage of the total cost of the computer system and in software cost estimation the error rate was relatively low. Today the software is the most expensive element of all computer systems and major error costs can be the difference between profit and loss. Now, the ability to more accurately estimate a critical is a factor for many of the major software successes and estimation of cost and activity of software known as a science. Many variables, such as human, technical, environmental, policy and management software can have an impact on the estimated cost and efforts for its development.

The software has a dual role: Software is a product and is a tool for the construction and delivery of the product. The software is a logical element rather than a physical system and has a characteristic that is so much different from hardware characteristics (Roger & Bruce, 2014).

The difference between software engineering and other engineerings can be seen in the following (Ian, 2010):

1- The product is intangible

2- There is no standard software process

3- Large software projects are often unique projects.

Computer technology and its applications based on three factors: hardware, software and manpower that in a form system it has the ability to design, plan and implement. One of the factors in the development of society is access to advanced technology in the field of software and related sciences. Software and software applications are the links to use the hardware (Igor, 2000). Due to these factors, we can understand why estimating software effort in the early stages of software development is one of the most important challenges that software developers and project managers are faced with it. Software effort or cost estimation is one of the important and influencing processes in software engineering that can have an important role in the success or failure of the project. Suitable and correct software effort or cost estimations make the project manager in the software life cycle have strong support to make different decisions and project manager, analyst, designer, programmer and software development team members know how much effort and time needs to make a good product. Cost or effort estimation models in the early stages of construction, with a minimum of information on the project cost or effort estimate system, are useful and necessary. Right effort estimation method provides the possibility of controlling time and cost of the system effectively. The accuracy of effort is an important factor in the success of the project (Kim et. al, 2004). Without a proper estimate of the cost required, the project manager can determine how much time and how many people and other resources needed for the project and in the case of error, the project will move in the direction of inevitable defeat.

According to reports published by the Standish Group (chaos report) in 2014, only 16.2% of projects in the world have successfully completed, while 52.7% of them have failed and 31.1% of them have cancelled. Software failure can destroy a company's reputation and cost and it shows the importance of accurate estimates of software effort and cost. Pressman recommends that use at least two different methods to estimate (Roger & Bruce, 2014).

2. Common Problems of Software Project Cost or Effort Estimation

Despite the different techniques and tools for estimating project costs or efforts, many software project estimations have less accuracy. T. DeMarco provides 4 recommendations for such carelessness and how to overcome them:

a. Providing an estimate for a large software project is complex and time-consuming activity. Many estimates must be made quickly and before the completion of the requirements of users.
b. People who provide software development effort or cost estimation don’t have enough experience in this field, especially in big projects. The use of professionals in this field and also keep records of previous project cost or effort estimates is a good solution for this problem.
c. Humans tend to have lower estimates for this reason sometimes estimators forget some additional costs such as test and Integration costs.
d. However, administrators want to provide an estimate but in fact, they want to win a contract with an estimate. In this situation, providing time and cost of the software application is a good solution (DeMarco, 1986).

3- Software Effort Estimation Methods

Several methods are provided for estimating the effort or cost of software that each has its advantages and disadvantages and according to the conditions of the problem are selected and used. In general, the estimation methods are divided into two categories: algorithmic and non-algorithmic. In the following, each category is explained briefly and the most important of them expressed.

3.1. Algorithmic methods

Algorithmic techniques use mathematical models to estimate the cost or effort of the project. Each algorithmic model is defined as a function of cost factors. Algorithmic methods are different in two aspects, one is the selection of cost factors and the other is defining of the cost function. First, we consider the cost factors and then describe the methods.

3.1.1. Cost factors

a. Product Factors: having reliability, complexity, database size, reusability, consistency of documentation with the requirement of the project life cycle.
b. Computer Factors: limitation of the runtime system, limitation of storage space, limitation of restarting the computer and variety of platforms.
c. Personnel Factors: skills of analysis team, skills of programmers, dominance on the platform, dominance on programming language and its tools, coordination of the team.
d. Project Factors: multisite development, using software tools.

3.1.2. Examples of algorithmic methods:

- **SLOC[1]:**

One way to estimate the size of a project is based on the number of lines and a comparison with other programs that have already been calculated SLOC. However, this estimate is simple but it is difficult at the beginning of the project because as long as the requirements are not fully predicted, the number of lines of the program not a precise prediction. Meanwhile, the number of lines of the program varies according to the programming language.

- **FP[2]:**

This method is based on the premise that the team members in any software project and its cost, depending on the scale of the project. Functional Points is calculated as follows:

Formula1: $FP = UFC * TC$

UFC is the number of points the net. That is obtained from the total number of inputs, outputs, logical files, Interface and query. TCF is the technical complexity factor, which is estimated between 0.65 and 1.35.

1 Source Line Of Code

2 Function Point

- **COCOMO[3]:**

This method was presented in 1981 by B.W.Bohem. Bohem in your model, consider the following factors in the cost of a software project:

1. Product reliability, 2. Product Complexity, 3. Time limits, 4. Limitations of main memory, 5. The availability of machine, 6. The ability of the analysis team, 7. Applications software development experience, 8. The ability of programming, 9. The use of modern design tools and 10. The use of modern programming techniques.

In this way, the effectiveness of each of factors on the projects will be ranked low (LOW) up to very high (EXTRA HIGH) and they are given weight. In this way, a matrix obtained by the rows of factors and the columns of the degree of impact of each factor on the project. This estimate considers many factors, and therefore the probability of error is high.

- **FPROM:**

This method is an estimating stage model and for each stage, when we can estimate human resources that the previous stage is performed. The difference between this method and FP is in estimates model (stage of it).

- **Seer-Sem:**

This method was introduced in 1980 by Galorath and more used to estimate commercial projects. The size of the project is the main factor in this estimate.

- **Linear Models:**

In this way, the simple structure is made which is calculated according to the formula:

Formula2: $$\text{Effort} = a_0 + \sum_{i=1}^{n} x_i a_i$$

Here, a_1 and a_2, …, and a_n, based on the project are determined.

- **Multiplicative models:**

In this method, the following formula is used:

Formula3: $$\text{Effort} = a_0 \prod_{i=1}^{n} a_i^{x_i}$$

Here, a_1 and a_2, ..., and a_n, based on the project are determined and the values of x_i are just - 1, 0 and 1.

- **Putman's model:**

This method was developed by Putman and is used for many projects. Equation method is presented below:

Formula 4: $$S = E \times (\text{Effort})^{1/5} t_d^{4/5}$$

E is characteristic of the environment and the response time is T_d.

3 CONSTRUCTION COST MODEL

3.2. Non-algorithmic models

- **Analogy Costing:**

In this way, based on a previous project, the new project cost estimate is done. This method can be used in the entire project or used in subsystems. In the first case, all the components of costs checked and in the second case, additional assessment of the similarities and differences between the current system and the previous system are done so it would be a more accurate estimate.

The advantage of this method is that it is based on real experience and disadvantage is that the former systems are not consistent with the current system and compare them are ignorant and wrong and can distort estimates.

- **Expert Judgment:**

In this method, costs estimate is based on personal experience of experts in the field of software development. To solve the possible inconsistencies in the estimates provided by different people, the techniques that the consensus estimates, such as Delphi and PERT will be used.

- **Machine learning Models:**

Most methods of software estimation used the methods and techniques of this model. Estimation Accuracy is increasing because of the learning ability and the ability to run multiple times of these models. This method can be divided into two main categories:

a) Neural networks
b) Fuzzy Method

These procedures are consistent with most projects and widely used (Khatibi & Jawawi, 2010).

- **Parkinson:**

In this method, software costs are not estimated, but according to available resources (regardless of project goals) are determined. Although this method in some cases provides a reasonable estimate the technique is not appropriate for the estimation of project costs.

- **Price-to-Win:**

In this method, instead of the software, features and applications, the cost is estimated based on the client's budget.

- **Button-up:**

In this way, each of the system components is estimated separately and then sum these estimates, the overall cost estimate of the project will be considered and the collection of them will be considered as total project costs estimation. To use this method, first, it is necessary to do a preliminary design of the system to get the structural components.

- **Top-Down:**

In this method, the total project costs are estimated based on general criteria. In the next step, the cost can be distributed between the different system components.

Accurate estimates of a project at least use two or three techniques (Roger & Bruce, 2014).

4. Artificial Neural Network

An artificial neural network is inspired by the biological nervous system and such as the brain, process information. The system consists of many processing elements called neurones that work together to solve a problem. ANNs, such as humans, learn by example and by processing the experimental data, pass the knowledge to the network structure. Thus, these systems are called smart because by calculating on numerical data or examples, learns the general rules. Figure 1 shows the general structure of an artificial neural network

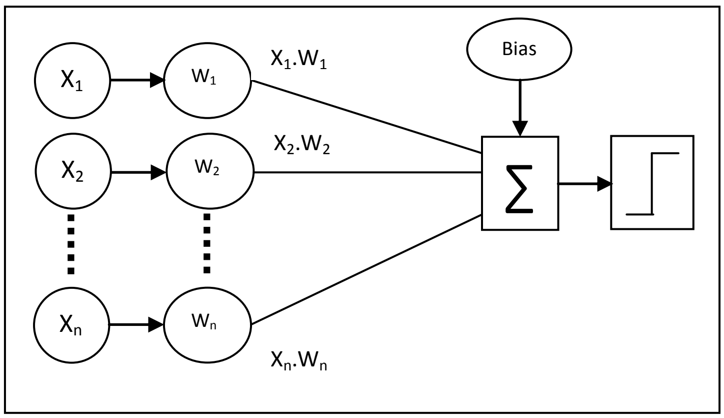

Figure1: Structure of an artificial neural network.

4.1. Why ANNs are worth reading?

- Neural networks, due to parallel processing, have high speed.
- Neural networks have the potential to solve the problems that are difficult or impossible to simulate by logic or other methods.
- Neural networks, such as the human brain is continuously learning and adapting to the environment. This means that if the network was trained for a situation and a small change in environment occurred, by a little training, are also used for the new situation.
- In a neural network, the wrong performance of part of the brain's neurones may not be a complete failure and there is also the possibility of making the correct decision.
- This method can provide a logical answer for data in uncertainty conditions.

4.2. The Reasons for using a Neural Network in Software Effort Estimation

There are many different methods for software effort estimation and Artificial Intelligence techniques have been used in this field to enhance the accuracy and reliability. One of the best artificial intelligence models that are used to estimate the software effort is an artificial neural network. These networks by using the technique of training assess data with minimal error. Evaluation criteria for estimating the effort/cost of software are errors. Previous studies have shown the error of neural networks is lower than the algorithmic effort estimations. In addition, in the process of software development, software factors and information about effort estimates are low and there is no possibility of using the algorithm with good approximation. Also, because of parallel processing of neural networks, the processing speed is high. Therefore, the use of neural networks to estimate the effort in the early stages of software development is useful. The neural network is made of two main computational parts. Part I: neurones, which are the nodes of ANN and part II: synapses, which are the weights and connections of ANN (Dave & Dutta, 2014).

5. Assessment Articles

Many articles about software effort estimation by using an artificial neural network were performed. In this paper 35 articles of the 2004 and 2015 selected. Among the articles were studied, 21 of them are journals and 14 of them are conferences which are listed in the following tables and charts.

Table1: Assessment Article and their Properties

Purposes and Results	Algorithm to compare	Parameters for assessment	Database used	The neural network used	Algorithm
The third model has a better performance at estimating software effort	Three neural networks were compared	MRE	148 (IT Project)	1: Neural network model (FP only). 2: Neural network model (six variables only). 3: Neural network model (FP + six variables).	Park & Baek, 2008
The third model has a better performance at estimating software effort	Three neural networks were compared	MRE BRE MMRE Pred	41 Lopez-Martin	1: Cascaded Feed Forward Back Propagation Neural Network model, Elman 2: Back Propagation Neural Network model, Layer Recurrent 3: Neural Network model & Generalised Regression Neural	Ghose et al, 2011
Neural networks have provided a better performance in estimating software effort	Regression Analysis		41 Lopez-Martin	Feed Forward-Back Propagation Neural Network	Bhatnagar, et al, 2010
Improvements of 3.27% in software effort estimation	COCOMO and typical neural network	MRE MMRE Pred	COCOMO With 63 project With 93 NASA 93 project	PSO-ANN-COCOMO II	Dan, 2013
Improvements of 8.36% in software effort estimation	COCOMO II	MRE MMRE Pred	COCOMO With 63 project With 93 NASA 93 project	ANN-COCOMO II	Attarzadeh, et al, 2012
According to MMRE parameter, FFNN performance is better than RBFNN model but based on RSD parameters RBFNN model provides a more accurate estimate	Regression Analysis Model And with each other	MMRE RSD	60 projects from and COCOMO NASA	1: Feed- Forward Neural Network (FFNN) 2: Radial Basis Functional Neural Network (RBFNN)	Dave & Dutta, 2011
Improvements in software effort estimation	Compare with databases	MMRE PRED	240	Cascade Correlation Neural Network (CCNN	Nassif, et al, 2012 (a)
Improve efforts	Use Case Point	MMRE	240	Feed- Forward Neural Network	Nassif,

compared to UCP estimates model	(UCP)	PRED		(FFNN)	et al, 2012 (b)
Reducing the time and effort required to estimate the cost of the software in the early stages of the project	Two neural networks were compare	MMRE	530	BPN with GA BPN with trial and error	Kim, et al, 2004
Improvements in software effort estimation	FFNN	MMRE MdMRE PRED	NASA Maxwell COCOMO81	PSO - Feed forward Link Artificial Neural Networks (PSO-FLANN)	Benal, et al, 2013
More accurate estimates	COCOMO	MRE	COCOMO81 With 63 project	Feed- Forward Neural Network (FFNN)	Mukherjee & Malu, 2014
Improvements in software estimation	Compare with data bases	MRE	NASA	Feed Forward-Back Propagation Neural Network	Shukla, et al, 2014
Improvements in software estimation	Compare with data bases	MMRE MRE PRED	COCOMO81	Feed Forward Neural Network (FFNN)	Subitsha Rajan, & 2014
Neural network model has a better performance at estimating software effort	Regression Analysis	MMRE	COCOMO With 63 project	Feed Forward Neural Network (FFNN)	Fabiana, et al, 2007
Fuzzy Neural Network Model has better performance than neural network model and COCOMO	COCOMO and neural network	MMRE MRE PRED	COCOMO With 63 project	Fuzzy Neural Network	Huang & Chiu, 2009
ANN model has a better performance at estimating software effort	APF, SLIM, COCOMO, Regression Analysis	MMRE PRED	COCOMO With 63 project	Feed Forward Neural Network (FFNN)	Fabiana, et al, 2008
Objective: To solve the problem of neural network learning The result: better estimates	FFNN	MMRE MdMRE PRED	COCOMO NASA With 60 project NASA93 With 93 project USC With 63 project	Multilayer Perceptrons (MLP)	Kultur, et al, 2009
Objective: To determine the scope and boundaries of the neural network	FFNN	MMRE PRED	COCOMO’81 With 63 project	FFNN+ COCOMO + K-mean	Sarac & Duru, 2013
RBFNN model has a better performance at estimating software effort	three neural networks were compared	BRE MRE MIBRE	With ISBSG 5052 project	1: General Regression Neural Network (GRNN) 2: Radial Basis Function Neural Networks (RBFNN) 3: Multilayer Perceptrons (MLP)	Lopez-Martin, 2015
The preferred method is determined	SLIM, COCOMO And with each other	MMRE PRED	NASA COCOMO	1: Feed-forward Neural Network 2: Recurrent Neural Networks 3: Radial Basis Function (RBF)	Hmaza & Kamel,

according to the terms of issue				Network 4: Neuro-Fuzzy Networks	2013
For small database, using MLP and for large database using the linear regression model offers a better estimate	Compared with databases	MMRE MRE PRED MdMRE	ISBSG With 223 project Western University Canadaب With 65 project CompuTop With 45 project	Multilayer Perceptrons (MLP)	Nassif et al, 2013
Reduce the difference between actual costs and estimated amounts	Neural network and COCOMO II	MRE	5 project	Feed-forward Neural Network + COCOMO II	Patil et al, 2014
Improving accuracy in estimating software effort	Neural network and COCOMO II	MMRE PRED	COCOMO 81 With 63 project	Feed-forward Neural Network + GA	Li, 2010
ANN model has a better performance at estimating software effort	Halstead, Walston-Felix Bailey-Basili Doty (for KLOC > 9)	MMRE MRE RMSSE	With 18 NASA project	Feed Forward-Back Propagation Neural Network	Kaur et al, 2010
	Compare with databases	MRE MdMRE	ISBSG With 4106 project NASA93 With 93 project	Multilayer Perceptrons (MLP)	Mittas et al, 2015
More accurate estimates	COCOMO	MMRE MRE	COCOMO With 63 project	Feed Forward-Back Propagation Neural Network	Soleimanian, 2011
Improvements in software effort estimation	Compare with databases	MMRE MRE PRED	COCOMO81 With 63 project NASA60 With 60 project NASA93با With 93 project Albrec With 24 project CF With 21 project Desharnais With 77 project	Feed Forward Neural Network	Jodpimai et al,2010
Improvements in software effort estimation	ABE and CART and MLR and SWR and ANN and ANN_C-means and ABE_PSO and ABE_GA and ABE_ANN and ABE_Grey	MMRE MRE PRED MdMRE	COCOMO With 63 project Maxwell With 62 project ISBSG With 5052 project	Clustering the project and choose the best method for each cluster Feed Forward Neural Network	Khatibi et al,2013
In projects with small size, offers more accurate estimates	statistical regression	MMRE MRE	132 project	Feed Forward Neural Network	Ivica & Lopez-Martin, 2010
Improvements in software cost	COCOMO		COCOMO II	Feed Forward-Back Propagation Neural Network	Tadayon 2005

estimation					
Improvements in software cost and effort estimation	Compare with databases	MMRE,MRE, MdMre,PRED,CC,MAE, RMSE	COCOMO 81 With 63 project Kermerer With 15 project IBMDPS With 24 project Hallmark With 28 project NASA96 With 93 project NASA63 With 63 project Maxwellبا With 100 project	1- Multi-Layer Perceptron (MLP) 2- Radial Basis Function (RBF) 3- Wavelet Neural Network (WNN) 4- Functional Link Artificial Neural Network (FLANN) 1- Generalised Regression Neural Network (GRNN)	Soleimanian et al, 2014
The best values in the China database is made when the learning rate is 0.9, and the number of neurones in the hidden layer is 15 and in the Maxwell database is made when the learning rate is 0.9, and the number of neurones in the hidden layer is 17	Apply different learning rates and compare with each other	MMRE و MSE وMAE و RMSE	China With 499 project Maxwellا With 62 project	Error Back Propagation Network (EBPN)	Hota et al, 2015
Improvements in software cost estimation	COCOMO	MMRE MRE PRED	69 project	Neuro-Fuzzy- COCOMO	Huang et al, 2007
2.54% improvement by applying PCA technique in GRNN	- M5 - Linear regression - SMO polykernel - SMO RBF kernel - GRNN - M5 + PCA - Linear regression +PCA - SMO polykernel + PCA - SMO RBF kernel + PCA - GRNN + PCA	MMRE MdMRE	COCOMO With 63 project	General Regression Neural Network (GRNN)	Sankara & Kumar, 2015
The proposed method has less error than the other two methods	COCOMO and typical neural network	MMRE MRE PRED	NASA with 60 project	MLP + ICA	Soleimanian & Maroufi, 2014

In Table 1 the literature review and the neural networks and the databases for each article are shown. Also, the evaluation parameters and algorithms which were compared with the model and the advantages and purposes of the proposed model are expressed.

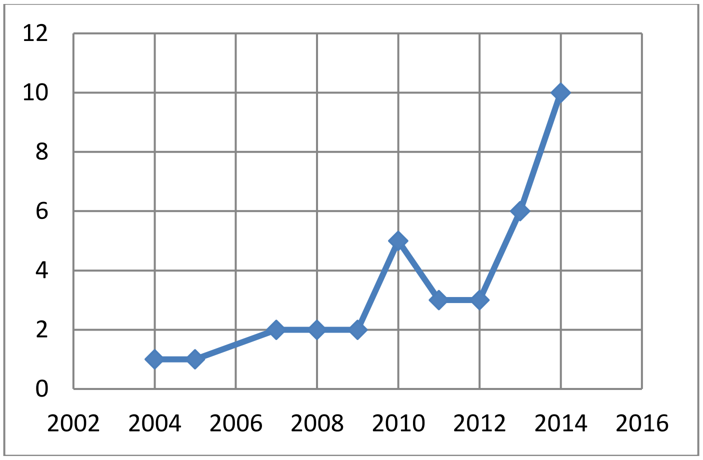

Figure 2: The Articles studied in the period 2004-2015

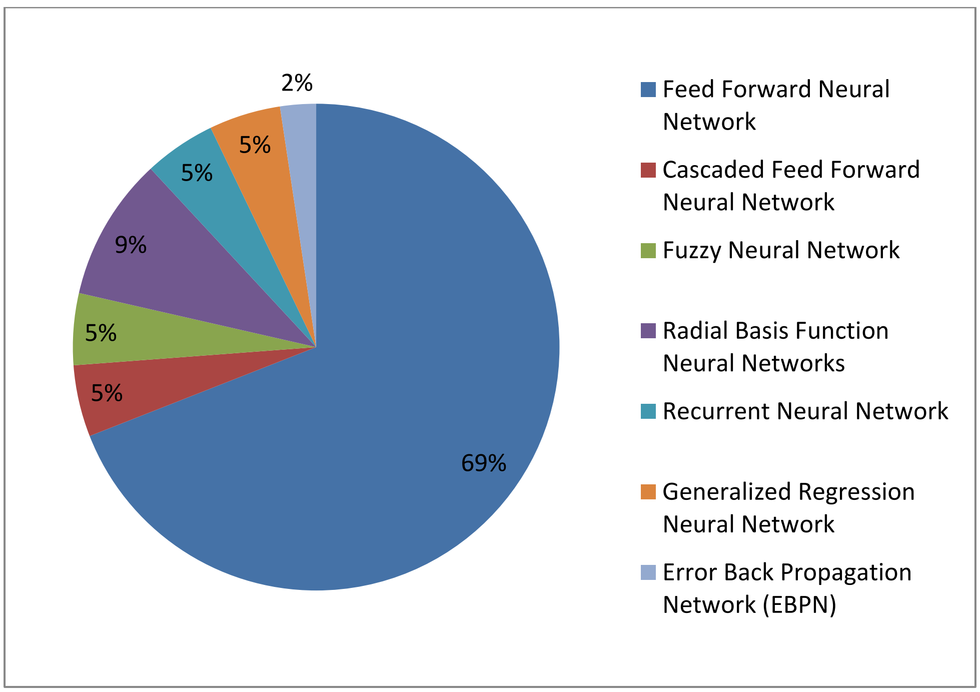

Figure 3: The amount and Neural Network Models used to Estimate Effort or Cost

In Figure 3, a variety of neural networks used in various articles and the amount of their use in the form of a graph are shown. As can be seen, in most papers, the typical neural network is used while the other models of them have been used less. According to this chart suggests researchers who are interested in estimates by using neural networks topics use other models of neural networks.

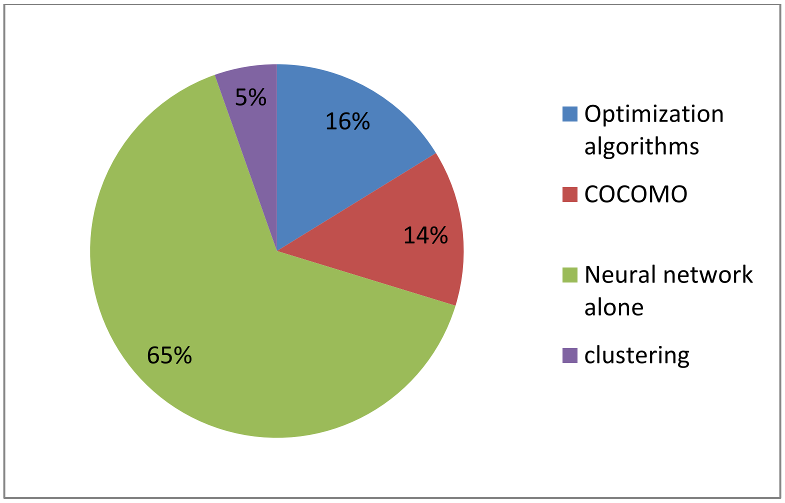

Figure 4: The use of other techniques with neural network in estimating the effort/cost software

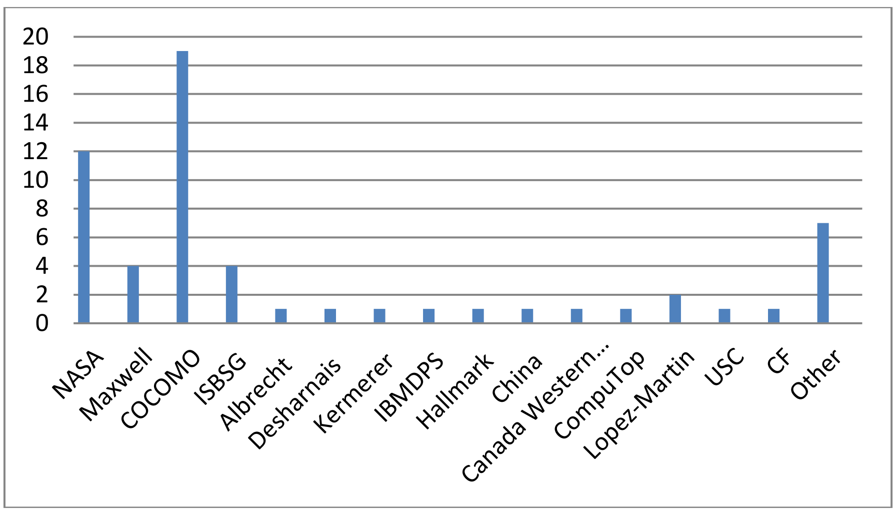

Figure 5: The use of different databases the articles

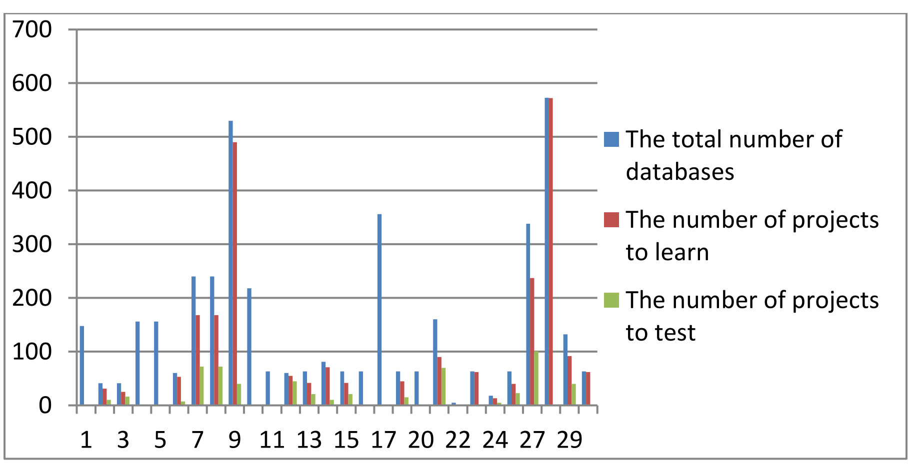

Figure 6: The total number of projects and the number of projects to learn and test used in each article

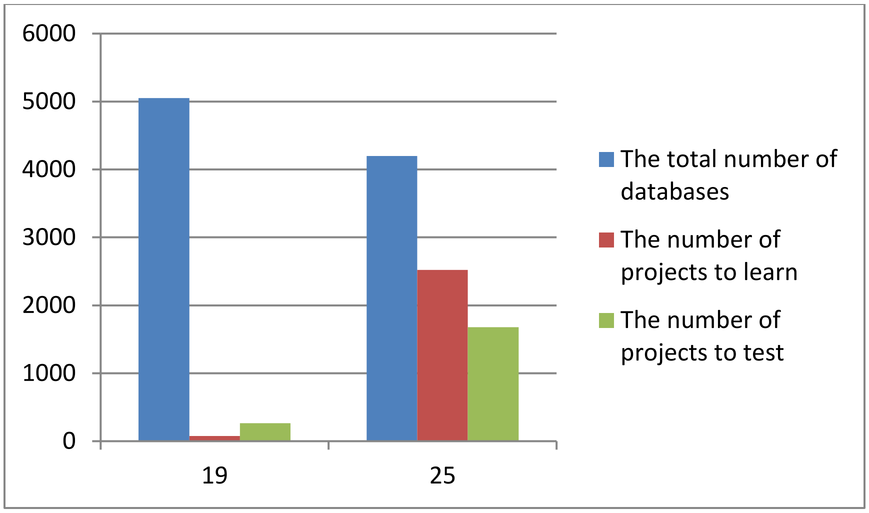

Figure 7: The total number of projects and the number of projects for learning and testing used in the articles of Shukla et al, 2014 and Sarac & Duru, 2013

6. Conclusion

Creative and abstract nature of software projects makes extremely difficult to estimate software cost and time. A successful software project is a project that is done in terms of specific and pre-determined cost and time. Project estimation is a method that before the start of the project.

we realise it can be possible. This estimate can be the difference between profit and loss. One of the methods used to estimate software projects is neural networks. Neural networks because of learning ability can be provided a more accurate estimation. In this case, 35 articles of the 2004 to 2015 assessment that all of them improve the software effort or cost estimation. Finally, the drawn graphs and tables can be a good guide to future works.

REFERENCES

1. Ali Bou Nassif, Danny Ho, Luiz Fernando Capretz “Towards an early software estimation using log-linear regression and a multilayer perceptron model” The Journal of Systems and Software, 2013, pp 144– 160
2. Attarzadeh, I, Mehranzadeh, A. Barati, A. “Proposing an Enhanced Artificial Neural Network Prediction Model to Improve the Accuracy in Software Effort Estimation” Fourth International Conference on Computational Intelligence, Communication Systems and Networks (CICSyN), 2012, pp 167-172
3. Cuauhtémoc López-Martín, “Predictive accuracy comparison between neural networks and statistical regression for development effort of software projects” Applied Soft Computing, Volume 27, 2015, pp434–449
4. Dave, V.S., Dutta, K. “Neural network based software effort estimation & evaluation criterion MMRE” 2nd International Conference on Computer and Communication Technology (ICCCT), 2011, pp 347-351
5. Dr. Haitham Hamza, Dr. Amr Kamel “Software Effort Estimation using Artificial Neural Networks: A Survey of the Current Practices” 10th International Conference on Information Technology: New

Generations, 2013, pp731-733
6. F. Soleimanian, A. Maroufi "Approach of software cost estimation with hybrid of imperialist competitive and artificial neural network algorithms" Journal of Scientific Research and Development, 2014, pp 50-57
7. F.Soleimanian, I. Maleki, S.Sadouni "Artificial Neural Networks Based Analysis of Software Cost Estimation Models" MAGNT Research Report, Volume 2, 2014, PP 597-505
8. Farhad Soleimanian "Neural Networks Application in Software CostEstimation: A Case Study", 2011, pp 69-73
9. G Kim,et al, "Comparison Of Construction Cost Estimating Models Based On Regression Analysis, Neural Networks, And Case-Based Reasoning," Elsevier 2004
10. GH Kim, JE Yoon, SH An, HH Cho, KI Kang "Neural network model incorporating a genetic algorithm in estimating construction costs" Building and Environment, Volume 39, Issue 11, 2004, pp 1333–1340
11. Ghose, M.K, Bhatnagar, R.; Bhattacharjee, V, "Comparing some neural network models for software development effort prediction" 2nd National Conference on Emerging Trends and Applications in Computer Science (NCETACS), 2011, pp 1-4
12. H Park, S Baek, "An empirical validation of a neural network model for software effort estimation" Expert Systems with Applications, 2008, Volume 35, Issue 3, pp 929–937
13. H.S. Hota, Ragini Shukla and S. Singhai "Predicting Software Development Effort Using Tuned Artificial Neural Network" Smart Innovation, Systems and Technologies Volume 33, 2015, pp 195-203
14. Ian Sommerville, "Software Engineering" 2010, Edition 9th, ISBN-13: 978-0137035151
15. Igor Hawryszkiewycz, "Introduction to Systems Analysis and Design"2000, Edition 5th, ISBN-13: 978-1740092807
16. Iris Fabiana de Barcelos Tronto, Jose Demisio Simoes da Silva, Nilson Sant'Anna "Comparison of Artificial Neural Network and Regression Models in Software Effort Estimation" Proceedings of International Joint Conference on Neural Networks,2007,pp12-17
17. Iris Fabiana de Barcelos Tronto, Jose´ Demı´sio Simo˜ es da Silva, Nilson Sant'Anna "An investigation of artificial neural networks based prediction systems in software project management" The Journal of Systems and Software, 2008, pp 356–367
18. Ivica Kalichanin-Balich, Cuauhtemoc Lopez-Martin "Applying a feed forward neural network for predicting software development effort of short-scale projects" Eighth ACIS International Conference on Software Engineering Research, Management and Applications, 2010, pp 269-275
19. Jaswinder Kaur, Satwinder Singh, Dr. Karanjeet Singh Kahlon, Pourush Bassi "Neural Network-A Novel Technique for Software Effort Estimation" International Journal of Computer Theory and Engineering, Vol. 2, No. 1, 2010,pp17-19
20. Lalit V. Patil, Rina M. Waghmode, S. D. Joshi, V. Khanna "Generic Model Of Software Cost Estimation: A Hybrid Approach", International Advance Computing Conference (IACC), 2014, pp 1379-1384
21. Nasser Tadayon "Neural Network Approach for Software Cost Estimation" Proceedings of the International Conference on Information Technology: Coding and Computing, Volume 2, 2005, pp 815 – 818
22. Nassif, A.B. ; Capretz, L.F. ; Ho, D. "Estimating Software Effort Using an ANN Model Based on Use Case Points" 11th International Conference on Machine Learning and Applications (ICMLA), Volume 2, 2012(b), pp 42-47
23. Nassif, A.B. ; Capretz, L.F. ; Ho, D. "Software Effort Estimation in the Early Stages of the Software Life Cycle Using a Cascade Correlation Neural Network Model" 13th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel & Distributed Computing (SNPD), 2012(a), pp 589-594
24. Nikolaos Mittas, Efi Papatheocharous, Lefteris Angelis, Andreas S. Andreou "Integrating non-parametric models with linear components for producing software cost estimations" The Journal of

Systems and Software, 2015, pp 120–134

25. Ömer Faruk Saraç, Nevcihan Duru "A Novel Method for Software Effort Estimation: Estimating with Boundaries" International Symposium on Innovations in Intelligent Systems and Applications (INISTA), 2013 pp1-5
26. P.Subitsha, J.Kowski Rajan "Artificial Neural Network Models For Software Effort Estimation" International Journal Of Technology Enhancements And Emerging Engineering Research, Vol 2, Issue 4, 2014, Pp 76-80
27. Parasana Sankara Rao, Reddi Kiran Kumar "Software Effort Estimation through a Generalised Regression Neural Network" Advances in Intelligent Systems and Computing, Volume 337, 2015, pp 19-30
28. Pichai Jodpimai, Peraphon Sophatsathit, and Chidchanok Lursinsap "Estimating Software Effort with Minimum Features using Neural Functional Approximation" International Conference on Computational Science and Its Applications, 2010, pp 266-273
29. R Shukla, M Shukla, T Marwala "Neural Network and Statistical Modeling of Software Development Effort" Advances in Intelligent Systems and Computing, Volume 236, 2014, pp 189-198
30. Roger Pressman, Bruce Maxim, "Software Engineering: A Practitioner's Approach" 2014, Edition 8th, ISBN-13: 978-0078022128
31. Roheet Bhatnagar, Vandana Bhattacharjee, Mrinal Kanti Ghose, "Software Development Effort Estimation – Neural Network Vs. Regression Modeling Approach" International Journal of Engineering Science and Technology, Volume 2(7), 2010, pp 2950-2956
32. S Mukherjee, RK Malu "Optimisation of Project Effort Estimate Using Neural Network" International Conference on Advanced Communication Control and Computing Technologies (ICACCCT), 2014, pp406-410
33. Sun-Jen Huang, Nan-Hsing Chiu "Applying fuzzy neural network to estimate software development effort" Appl Intell, 2009, pp73–83
34. T.DeMarco,"Controlling Software Projects: Management, Measurement, and Estimates" 1986, Edition 1st, ISBN-13:007 – 6092031185
35. TR Benala, R Mall, S Dehuri "Software Effort Estimation Using Functional Link Neural Networks Optimised by Improved Particle Swarm Optimisation" Swarm, Evolutionary, and Memetic Computing Lecture Notes in Computer Science, Volume 8298, 2013, pp 205-213
36. V. Khatibi Bardsiri, Dayang Norhayati Abang Jawawi, A. Khatibi Bardsiri, E. Khatibi "LMES: A localised multi-estimator model to estimate software development effort" Engineering Applications of Artificial Intelligence, 2013, pp 2624–2640
37. Vahid Khatibi, Dayang N. A. Jawawi," Software Cost Estimation Methods: A Review" Journal of Emerging Trends in Computing and Information Sciences, Volume 2 No. 1, 2010, pp 21-29
38. VS Dave, K Dutta, "Neural network based models for software effort estimation: a review", Artificial Intelligence Review, 2014, Volume 42, Issue 2, pp 295-307
39. X.Huang, D.Ho, J.Ren, L.F. Capretz "Improving the COCOMO model using a neuro-fuzzy approach" Applied Soft Computing, vol 7, 2007, pp 29–40
40. Yigit Kultur, Burak Turhan, Ayse Bener "Ensemble of neural networks with associative memory (ENNA) for estimating software development costs" Knowledge-Based Systems, 2009, pp 395–402
41. Zhang Dan, "Improving the accuracy in software effort estimation: Using artificial neural network model based on particle swarm optimisation" International Conference on Service Operations and Logistics and Informatics (SOLI), 2013, pp 180-185
42. Zhenyou LI "Intelligently Predict Project Effort by Reduced Models" International Conference on E-Business and E-Government, 2010, pp 1536-1542