Neural Networks based Software Development Effort Estimation: A Systematic Mapping Study

Fatima Ezzahra Boujida[1][a], Fatima Azzahra Amazal[1][b] and Ali Idri[2][c]

[1]LabSIV, Department of Computer Science, Faculty of Science, Ibn Zohr University, BP 8106, 80000 Agadir, Morocco

[2]Software Projects Management Research Team, ENSIAS, Mohammed V University, Madinate Al Irfane, 10100 Rabat, Morocco

Keywords: Systematic Mapping Study, Software Development Effort Estimation, Artificial Neural Networks.

Abstract: Developing an efficient model that accurately predicts the development effort of a software project is an important task in software project management. Artificial neural networks (ANNs) are promising for building predictive models since their ability to learn from previous data, adapt and produce more accurate results. In this paper, we conducted a systematic mapping study of papers dealing with the estimation of software development effort based on artificial neural networks. In total, 80 relevant studies were identified between 1993 and 2020 and classified with respect to five criteria: publication source, research approach, contribution type, techniques used in combination with ANN models and type of the neural network used. The results showed that, most ANN-based software development effort estimation (SDEE) studies applied the history-based evaluation (HE) and solution proposal (SP) approaches. Besides, the feedforward neural network was the most frequently used ANN type among SDEE researchers. To improve the performance of ANN models, most papers employed optimization methods such as Genetic Algorithms (GA) and Particle Swarm Optimization (PSO) in combination with ANN models.

1 INTRODUCTION

Over the last decades, software effort estimation techniques have seen increasing demand among researchers and practitioners. The researchers working in the software development effort estimation (SDEE) field are facing greater challenges in order to produce a measurement tool that accurately estimates the development effort. In this regard, a reliable estimate of software effort is crucial to ensure that time and budget constraints are met (Sommerville, 2010). Erroneous estimates can lead to situations where the software cannot be produced on time and within the budget set in the initial planning, which in turn may result in loss of contracts (Jones, 2007).

To get accurate estimates, several SDEE models have been developed. They can be grouped into three main categories (de Barcelos Tronto et al., 2008): (1) Parametric models (Boehm, 2000; Mendes, 2008) which presume that the function expressing the Relationship between effort and software attributes has a well-defined form; (2) Machine learning (ML) models (Wen et al., 2012; Huang et al., 2008; Kumar et al., 2008; Elish, 2009; Shepperd and Schofield, 1997; Ahmed and Muzaffar, 2009) which rely on the use of artificial intelligence (AI) techniques such as case-based reasoning (CBR) (Idri et al., 2015; Idri et al., 2002a; Idri and Zahi, 2013; Idri et al., 2019; Idri, 2002), decision trees (DT)(Idri and Elyassami, 2011), genetic algorithms (GA) and artificial neural networks (ANN) (Idri et al., 2002b; Idri et al., 2007); and (3) Expert judgment (Hughes, 1996) which is purely based on the experience of one or more experts in previously completed projects to derive estimates. Machine learning (ML) techniques have recently received special attention from SDEE community. Wen et al. (Wen et al., 2012) conducted a systematic literature review (SLR) on the use of ML models in SDEE. Their SLR revealed that ANN and CBR were the most frequently used techniques by SDEE researchers. Further, among the eight ML techniques that were identified in the study, ANN models were the most accurate in terms of arithmetic mean of Preds (25) and arithmetic mean MMREs (mPred(25)= 64% and mMMRE = 37%).

[a] https://orcid.org/0000-0001-8733-0085
[b] https://orcid.org/0000-0002-9008-656X
[c] https://orcid.org/0000-0002-4586-4158

Boujida, F., Amazal, F. and Idri, A.
Neural Networks based Software Development Effort Estimation: A Systematic Mapping Study.
DOI: 10.5220/0010603701020110
In *Proceedings of the 16th International Conference on Software Technologies (ICSOFT 2021)*, pages 102-110
ISBN: 978-989-758-523-4

ANN models have been widely used among software researchers since they have several advantages. First, they are able to learn from historical data (Attarzadeh and OW, 2014) and model complex relationships between effort and cost drivers (Iwata et al., 2010). Second, it is possible to obtain more accurate estimates with the correct configuration of the weights (Huang and Chiu, 2007). Third, they can be adapted to the environmental changes through free parameters (synapses) (Kamlesh et al., 2019; Kaur and Singh Salaria, 2013).

As the research area related to the use of ANN in SDEE has deepened, the number of ANN-based SDEE papers has increased. Therefore, it becomes important to summarize the existing works and provide an overall view. Consequently, it is of crucial importance to construct a classification scheme and structure the reported studies on ANN-based SDEE, in order to understand and facilitate their application.

To the best of the authors' knowledge, no systematic mapping study has been conducted on the use of ANN in SDEE. Therefore, in this paper, a systematic mapping study (SMS) is conducted to investigate the use of artificial neural networks in SDEE. As stated in (Kitchenham, 2010), a mapping study aims to find and classify primary studies in a specific thematic area. It can be used to identify available literature.

The purpose of this SMS is to: 1) identify the existing Neural Network-based SDEE papers published until 2020; and 2) classify and evaluate the selected studies with respect to five criteria: publication source, research approach, contribution type, techniques used in combination with ANN models and type of the ANN used.

This paper is organized as follows: Section II reports the research methodology used to conduct our SMS. Section III presents the results of the mapping study. Section IV shows the implications for research and practice. Conclusions and future works are presented in Section V.

2 RESEARCH METHODOLOGY

To conduct our study, we adopted the mapping process suggested by Kitchenham and Charters (Kitchenham, 2007). According to Ref. (Kitchenham, 2007), the purpose of a mapping study is to find and classify primary studies related to a specific thematic area. This process is based on five steps: (1) define the mapping questions,(2) conduct an exhaustive search for primary studies, (3) select studies, (4) extract data, and (5) synthesize data. The description of each of these steps is given below.

2.1 Mapping Questions

The first step of the mapping process consists on defining the set of the mapping questions (MQs) to be addressed. Five MQs were defined. The MQs and their main motivations are listed in Table 1. These MQs are related to the properties and categories presented in Table 2.

2.2 Search Strategy

The objective of this step is to identify the relevant ANN-based SDEE papers that treat the MQs of Table 1. To carry out the search, we used four digital libraries: IEEE Xplore, ACM Digital library, Science Direct and Google Scholar. The IEEE, ACM and Science Direct were chosen to provide full-text access to the highest quality engineering and technical literature. Google Scholar was used to find additional relevant studies since it explores other electronic databases. Note that, these four databases were used in our previous mapping and review studies (Idri et al., 2015; Amazal and Idri, 2019; Idri et al., 2016a; Idri et al., 2016b). They were also adopted by other researchers to conduct their SMS and SLR such as Wen et al. (Wen et al., 2012). All searches were limited to the papers published in the 1993-2020 period. To conduct the search using the above-mentioned digital libraries, a search string was constructed. To this end, we identified the major terms related to our MQs as well as their synonyms and alternative spellings. Then, we used the Boolean operators OR and AND to join synonymous and main terms (Idri et al., 2015; Amazal and Idri, 2019). The constructed search string was as follows: (”neural network” OR ”ANN” OR ”MLP” OR ”multi-layer perceptron”) AND (”software” OR ”system” OR ”application” OR ”project”) AND (”cost” OR ”effort”) AND (estimate* OR predict*).

To make sure that all papers that address the MQs of Table 1 were retrieved, we divided the search process into two phases. In the first phase, we applied the search string on the four digital libraries to retrieve the set of candidate studies. In the second phase, we evaluated each of the candidate papers using a set of inclusion and exclusion criteria to decide whether it should be included or rejected. The evaluation was based on title, abstract and keywords. In case of doubt, the full text was examined. The reference lists of all retained papers (papers that satisfy the inclusion and exclusion criteria) were checked to ensure that no ANN-based SDEE paper was missed in the first phase.

Table 1: Mapping questions.

ID	Mapping Question	Motivation
MQ1	Which sources are the main targets for ANN based SDEE papers?	To identify the main publication channels targeted by ANN based SDEE studies.
MQ2	What research approaches are applied in ANN based SDEE papers?	To investigate the research approaches most applied in SDEE studies using ANN models.
MQ3	In which contribution types are ANN based SDEE papers classified?	To discover the different contribution types of ANN based SDEE studies.
MQ4	What are the most frequently used techniques and models in combination with ANN based SDEE models?	To identify the techniques and models that are combined with ANN based SDEE models to improve their performance.
MQ5	What are the main types of ANNs used in SDEE studies?	To identify the most used types of ANNs in SDEE papers.

Table 2: Classification criteria.

Research approach	History-based evaluation (HE), solution proposal (SP), case study (CS), review (-RV), survey (SV).
Contribution type	Technique, comparison, validation, metric, model.
Techniques used in combination with ANN models	Optimization method (Opt), Constructive Cost Model (COCOMO),Clustering techniques (CT), Use Case Points (UCP), Functional Point (FP), Class Point Analysis (CPA), K-Nearest Neighbors (KNN), Case Based Reasoning (CBR), Bayesian Regularization (BR), Morphological operator (Mor), Associative Memory Technique (AMT).
Neural network used	Feed-Forward Neural Network (FFNN), General Regression Neural Network (GRNN), Functional Artificial Neural Network (FLANN),Cascade Correlation Neural Network (CCNN), Adaptive Neuro Fuzzy Inference System (ANFIS), Wavelet Neural Network (WNN), Radial Basis Function Neural Network (RBFNN), Recurrent Neural Networks (RNN), Resilient Back Propagation Neural (RBPNN), Feedforward back-propagation (FBNN), Elman Neural Network (ENN), Multi-Layer Perceptron (MLP), Hybrid Neural Network (HNN)

2.3 Study Selection

This step aims to select the relevant studies that answer the mapping questions posed in Table 1. To this end, we defined a set of inclusion and exclusion criteria to evaluate each of the candidate studies and decide on its relevance to our SMS and determine whether it should be included or discarded. These criteria are linked by the Boolean operator OR and are given bellow.

- Inclusion criteria
  - Develop, improve or use an ANN-based model to predict software effort.
  - Evaluate and/or compare the performance of an ANN-based SDEE model with that of other SDEE techniques/models (e.g. linear regression, decision tree, etc.).
  - Propose a hybrid model that uses a combination of neural networks and other techniques to predict software effort.
- Exclusion criteria
  - Same study with duplicate publications (only the most complete paper is taken into account).
  - Studies with focus on how to predict maintenance or testing effort.
  - Studies with focus on how to predict software size or time.
  - Studies estimating effort for construction projects.

2.4 Data Extraction Strategy and Synthesis Method

Once the studies that are relevant to our SMS have been selected, the data necessary to address the MQs were collected. To facilitate data extraction, we used a data extraction form that we filled out for each of the selected studies. Table 3 shows the data extracted from each study. To aggregate evidence and synthesize the extracted data with respect to each of the mapping questions, we used a narrative synthesis approach. Further, we used some visualization tools such as bar graphs and pie charts to facilitate the analysis of the results.

Table 3: Data extraction form.

Data extractor
Paper identifier
Author(s) name(s)
Article title
Publication year
Data checker
(MQ1) Publication source
(MQ2) Research approach (History-based evaluation, solution proposal, case study, review, survey)
(MQ3) Contribution type (Technique, comparison , validation, metric, model)
(MQ4) Techniques used in combination with ANN
(MQ5) Type of the ANN used

3 RESULTS AND DISCUSSION

This section presents and discusses the results obtained from the 80 selected studies with respect to the five mapping questions listed in Table 1.

3.1 Overview of the Selected Studies

Figure 1 shows the results of the selection process. Conducting the search using our search string and the four digital databases returned 1817 candidate studies. Since many of them would not be useful to answer the MQs, the inclusion and exclusion criteria were applied to decide on their relevance to our SMS. As motioned earlier, the evaluation of the candidate studies was performed based on keywords, title, abstract and full text. This resulted in 80 selected studies. No extra relevant papers were found by scanning the reference lists of the selected studies.

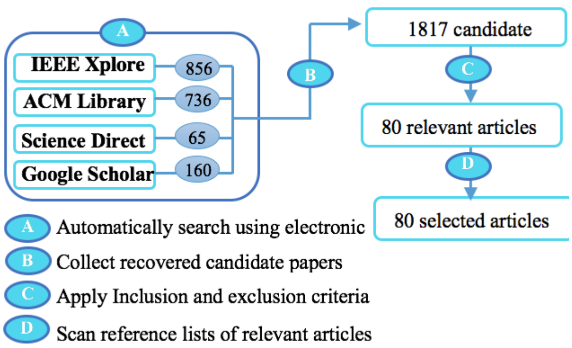

Figure 1: Results of selection process.

The list of selected articles can be sent upon request to researchers for further research. In addition, the list will be available in our next systematic literature review paper.

3.2 Publications Sources and Trends (MQ1)

Two main publication sources were targeted by the selected studies: journals and conferences. Specifically, 46 (57.5%) papers came from journals and 34 (42.5%) papers were published in conferences. Table 4 indicates the publication sources of the selected papers with at least 2 papers on the use of ANN in SDEE. Four journals were identified with 2 studies addressing the use of ANN in SDEE: Expert Systems with Applications (ESWA), International Journal of Information Technology (IJIT), Global Journal of Computers and Technology (GJCT), and Software Engineering Notes (SEN). Two conferences were identified with 2 papers: International Joint Conference on Neural Networks (IJCNN), and World Congress on Services (SERVICES). The other publication sources were not listed in the table since they were used only once to publish ANN-based SDEE studies.

Table 4: Publication sources of the selected studies.

Publication venue	Type	# of studies
Expert Systems with Applications (ESWA)	Journal	2
International Journal of Information Technology (IJIT)	Journal	2
Global Journal of Computers and Technology (GJCT)	Journal	2
Software Engineering Notes (SEN)	Journal	2
International Joint Conference on Neural Networks (IJCNN)	Conference	2
World Congress on Services (SERVICES)	Conference	2

To investigate the publication trends of ANN-based SDEE studies, we analyzed the number of published papers over the years. Figure 2 shows the distribution of the number of papers over the 1993-2020 period. As can be noticed, no ANN-based SDEE paper was found in 1997, 1999, 2002, 2003, and 2006. Besides, the use of ANN in SDEE has gained research interest between 2011 and 2019 (74% of the selected papers).

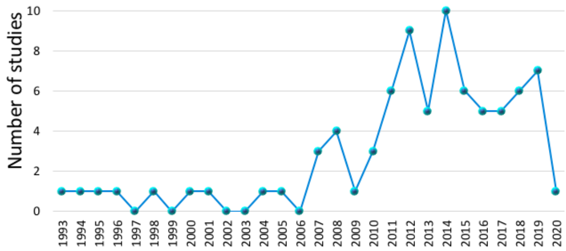

Figure 2: Publication trends of the selected studies.

3.3 Research Approaches (MQ2)

Five main research approaches were adopted by the authors of the selected studies as shown in Table 5: Solution Proposal (SP), History-based Evaluation (HE), Case Study (CS), Review (RV) and Survey (SV). As can be seen from Table 5, , HE was the most frequently used approach (89% of papers) followed by SP (76% of papers). Besides, only 5% (4 out of 80) of the selected papers were reviews or surveys. It can also be seen that, most of the selected papers used historical datasets (89%) or Case Studies (4%) to empirically validate their works.

When investigating the use of the HE approach in the selected studies, we found that most papers (81%) used historical datasets to evaluate the performance of their proposed ANN-based SDEE model or to perform a comparison with other SDEE techniques. Historical datasets were also employed to study the effect of some dataset properties on the prediction accuracy of ANN-based SDEE models.

As for the datasets used to evaluate ANN-based SDEE models, the selected papers used various datasets with different sizes and characteristics. Figure 3 shows the datasets used as well as the number of papers using these datasets. It can be noticed that, COCOMO was the most frequently used dataset (27 studies) followed by Nasa (20 studies) and Desharnais (10 studies). Of the 71 studies using the HE approach, 38 datasets were employed in 108 evaluations. Note that, each study may perform experiments using more than one dataset.

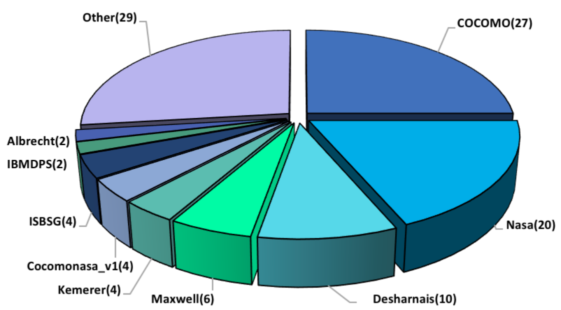

Figure 3: Distribution of studies using the HE research approach over the datasets.

3.4 Contribution Type (MQ3)

By analyzing the contribution types of the selected papers, five main contribution types were identified: Comparison, model, technique, validation and metric. Figure 4 shows the number of studies per contribution type. As can be seen, most studies are included in the Comparison contribution type (87%). These studies either compared various configurations of the same ANN-based SDEE model or performed comparisons with other SDEE models. Researchers were also interested on developing new ANN-based SDEE models or improving existing ones (66%). Note that, no tool was developed to estimate software effort using ANN. This lack of ANN-based SDEE Tools may limit the use of ANN models to estimate software effort in industry.

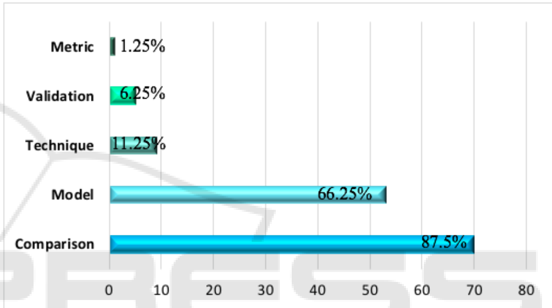

Figure 4: Number of studies per contribution type.

3.5 Techniques Used in Combination with ANN Models (MQ4)

Different techniques were used in combination with ANN-based SDEE models to overcome a set of difficulties related to (1) the selection of the optimal parameters; (2) the reduction of the size of the input characteristics; and (3) the optimization of the number of neurons in the hidden layer. Figure 5 shows the techniques used in combination with ANN models and the number of studies in which they were applied. It can be seen from Figure 5 that neural networks were most often combined with optimization techniques (Opt) (15%), followed by clustering techniques (CT) and COCOMO model (11% for each). The most frequently used optimization techniques were genetic algorithms (GA) and particle swarm optimization (PSO). The former was used to optimize the number of neurons in the hidden layers, reduce the dimensions of the set of features, or reduce the complexity of neural network models (Bisi and Kumar Goyal, 2016; Goyal and Bhatia, 2019; Tirimula et al., 2012; K.S., 2000; Oda and Nakazato, 2016). The latter was applied for its global classification ca-

Table 5: Distribution of ANN-based SDEE research approaches over the years.

Research Approach	1993 - 1999	2000 - 2006	2007 - 2013	2014 - 2020	Total
HE	3	4	27	37	71
SP	4	3	23	31	61
CS	1	0	2	0	3
RV	0	0	1	2	3
SV	0	0	1	0	1

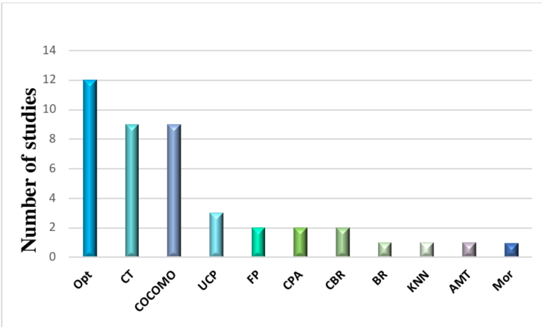

Figure 5: Techniques used in combination with ANN-based SDEE models.

pabilities and to adjust the parameters of the adhesion function (Bisi and Kumar Goyal, 2016; Tirimula et al., 2013; Suharjito et al., 2016).

We investigated the use of CT and COCOMO in combination with ANN models for SDEE. The main goals for using CT and COCOMO were the following:

- CT: (1) to aggregate datasets and make data as normal as possible; (2) to increase the training efficiency of ANN models; and (3) to improve the convergence speed of back-propagation and deal with imprecise and uncertain data (Azath et al., 2018; Dasheng and Shenglan, 2012; Praynlin and Latha, 2018; Hassankashi and Hanchate, 2017; Huang and Chiu, 2007; Kanmani et al., 2008; Kanmani et al., 2008; Anita et al., 1998).
- COCOMO: for its characteristics and capabilities such as the use of a clear definition of the attributes for software projects (Attarzadeh and OW, 2014; Kamlesh et al., 2019; Kaur and Singh Salaria, 2013; Kumar and Kumar, 2014; Sivakumar, 2014; Sarno et al., 2015; Satyananda Reddy and Raju, 2009; Satyananda Reddy and Raju, 2010; Tadayon, 2005).

3.6 Types of Neural Networks Used (MQ5)

Various neural network models were developed in the last years. Figure 6 shows the types of the ANN models used in the selected papers as well as the number of studies using each type. As can be seen, the Feedforward neural network was the most frequently used ANN type in the selected studies (36% for FFBN and 21% for FFNN), followed by Multilayer Perceptron (MLP) and Adaptive Neuro Fuzzy Inference System(ANFIS) (15% and 9% respectively). The widespread use of these ANN models among researchers may be due to the fact that they are simpler to use than other ANN models and more suited to the problem of software effort estimation. Other ANN types were rarely used, such as Resilient Back Propagation Neural Network (RBPNN) (1%), wavelet neural network (WNN) and Functional Link Artificial Neural Network (FLANN) (2% each).

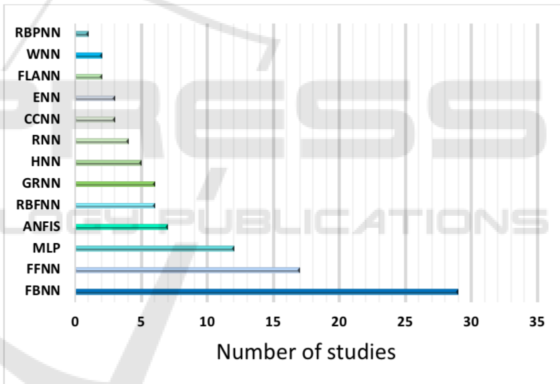

Figure 6: Number of studies per ANN type.

4 IMPLICATION FOR RESEARCH AND PRACTICE

This study aims to present an overview of the use of ANNs in software development effort estimation. In this section we provide some recommendations to researchers and practitioners based on the findings of our mapping study.

This study revealed a lack of research on how to evaluate ANN-based SDEE models in real-life contexts. In fact, only one case study was identified among the 80 ANN-based SDEE selected papers. Therefore, we recommend for researchers to cooperate with practitioners in order to investigate in depth the use of ANN models in industry to estimate software effort. Besides, most studies used historical datasets to evaluate or compare the performance of their proposed ANN-based SDEE models. The datasets used in these studies are not large enough to get good results. It is therefore recommended for practitioners to provide researchers with larger datasets.

No tool was developed to encourage the use of ANN models among SDEE practitioners. This implies that, researchers should develop tools that implement their ANN-based SDEE models and facilitate their use among practitioners and researchers. Furthermore, this study found that optimization and clustering techniques are the most frequently used techniques in combination with ANN-based SDEE studies. Other techniques such as Bayesian Regularization (BR) and K-Nearest Neighbors (KNN) were rarely used. Therefore, researchers are encouraged to conduct further research works using other techniques in combination with ANN.

We noticed that, some types of ANNs were rarely used, such as Resilient Back Propagation Neural Networks (RBPNN), wavelet neural networks (WNN) and Functional Link Artificial Neural Net- works (FLANN) while others were used more often such as Feedforward neural networks, Multilayer Perceptron (MLP) and Adaptive Neuro Fuzzy Inference System (ANFIS). Therefore, it is suggested to SDEE researchers to explore the use of other types of ANNs such as Long short-term memory (LSTM) to improve prediction accuracy.

5 CONCLUSION AND FUTURE WORK

The aim of this systematic mapping study was to identify and classify the existing works on ANN-based SDEE. The paper identified 80 relevant ANN-based SDEE studies and classified them according to publication source, research approach, contribution type, techniques used in combination with ANN models and type of the neural network used. The main findings of our SMS are the following.

(MQ1): Journals and conferences were the main publication sources of ANN-based SDEE papers. Besides, the use of ANN in SDEE has gained research interest between 2011 and 2019.

(MQ2): History-based evaluation was the most frequently used research approach followed by solution proposal. The use of both approaches by researchers is increasing over time.

(MQ3): Most of the selected studies focused on developing a new ANN model or evaluating and comparing the performance of their ANN model with other SDEE models.

(MQ4): Optimization methods and clustering techniques were the most frequently used techniques in combination with artificial neural networks, followed by COCOMO.

(MQ5): Most papers used feedforward neural networks followed by multilayer perceptron and the adaptive neuro fuzzy inference system.

Conducting this SMS allowed us to build a classification scheme of ANN-based SDEE research area. However, many issues related to the performance of ANN-based SDEE models need to be investigated in depth.

Therefore, we see a need to systematically analyze and summarize the evidence of ANN-based SDEE models performance by conducting a systematic literature review that aggregates the results of SDEE studies proposing new or modified ANN models.

To this end, a systematic literature review is ongoing to analyze the use of ANN in SDEE by taking into consideration the findings of this SMS.

Another important issue that should be addressed when dealing with ANN-based SDEE models consists on how to interpret ANNs to gain practitioners acceptance (Idri et al., 2002b; Idri et al., 2004; Idri et al., 2010). In fact, ANNs are viewed as black boxes which may prevent them from being widely used by SDEE researches and practitioners.

REFERENCES

Ahmed, M. and Muzaffar, Z. (2009). Handling imprecision and uncertainty in software development effort prediction: a type-2 fuzzy logic based framework. *Information and Software Technology*, 51(3):640–654.

Amazal, F. and Idri, A. (2019). "handling of categorical data in software development effort estimation: A systematic mapping study". In *Proceedings of the 2019 Federated Conference on Computer Science and Information Systems, FedCSIS 2019, Leipzig, Germany, September 1-4, 2019*, volume 18 of *Annals of Computer Science and Information Systems*, pages 763–770.

Anita, L., Cheng, C., and Balakrishnan, J. (1998). Software development cost estimation: Integrating neural network with cluster analysis. *Inf. Manag.*, 34(1):1–9.

Attarzadeh, I. and OW, S. (2014). Proposing an effective artificial neural network architecture to improve the precision of software cost estimation model. *Int. J. Softw. Eng. Knowl. Eng.*, 24(6):935–954.

Azath, H., Mohanapriya, M., and Rajalakshmi, S. (2018). Software effort estimation using modified fuzzy C means clustering and hybrid ABC-MCS optimization in neural network. *J. Intell. Syst.*, 29(1):251–263.

Bisi, M. and Kumar Goyal, N. (2016). Software development efforts prediction using artificial neural network. *IET Softw.*, 10(3):63–71.

Boehm, B. (2000). Software cost estimation with cocomoii. *NJ: Prentice-Hall*.

Dasheng, X. and Shenglan, H. (2012). Estimation of project costs based on fuzzy neural network. In *2012 WICT World Congress on Information and Communication Technologies. Trivandrum, India*. IEEE.

de Barcelos Tronto, I., da Silva, J. S., and Sant'Anna, N. (2008). An investigation of artificial neural networks based prediction systems in software project management. *The Journal of Systems and Software*, 81(3):356–367.

Elish, M. (2009). Improved estimation of software project effort using multiple additive regression trees. *Expert Systems with Applications*, 36(7):10774–10778.

Goyal, S. and Bhatia, P. (2019). Ga based dimensionality reduction for effective software effort estimation using ann. *Advances and Applications in Mathematical Sciences*, 18(8):637–649.

Hassankashi, M. and Hanchate, D. (2017). Role of ann and fuzzy in software cost estimation. *Journal of Basic and Applied Research International*, 21(1):1–9.

Huang, S. and Chiu, N. (2007). Applying fuzzy neural network to estimate software development effort. *International Journal of Research on Intelligent Systems for Real Life Complex Problems*, 30(2):73–83.

Huang, S.-J., Chiu, N.-H., and Chen, L.-W. (2008). Integration of the grey relational analysis with genetic algorithm for software effort estimation. *European Journal of Operational Research*, 188(3):898–909.

Hughes, R. (1996). Expert judgment as an estimating method. *Information and Software Technology*, 38:67–75.

Idri, A., Abnane, I., Hosni, M., and Abran, A. (2019). Analogy software effort estimation using ensemble KNN imputation. In *45th Euromicro Conference on Software Engineering and Advanced Applications, SEAA 2019, Kallithea-Chalkidiki, Greece, August 28-30, 2019*, pages 228–235. IEEE.

Idri, A., Abran, A., and Khoshgoftaar, T. (2002a). Estimating software project effort by analogy based on linguistic values. In *Proceedings Eighth IEEE Symposium on Software Metrics*, pages 21–30.

Idri, A., Amazal, F., and Abran, A. (2015). "analogy-based software development effort estimation: A systematic mapping and review". *Information and Software Technology*, 58.

Idri, A. and Elyassami, S. (2011). Applying fuzzy ID3 decision tree for software effort estimation. *CoRR*, abs/1111.0158.

Idri, A., Hosni, M., and Abran, A. (2016a). Systematic literature review of ensemble effort estimation. *J. Syst. Softw.*, 118:151–175.

Idri, A., Hosni, M., and Abran, A. (2016b). "systematic mapping study of ensemble effort estimation". *in Proc. 11th International Conference on Evaluation of Novel Software Approaches to Software Engineering*.

Idri, A., Khoshgoftaar, T., and Abran, A. (2002b). Can neural networks be easily interpreted in software cost estimation? In *2002 IEEE World Congress on Computational Intelligence. 2002 IEEE International Conference on Fuzzy Systems. FUZZ-IEEE'02. Proceedings (Cat. No.02CH37291)*, volume 2, pages 1162–1167.

Idri, A., Mbarki, S., and Abran, A. (2004). Validating and understanding software cost estimation models based on neural networks. In *Proceedings. 2004 International Conference on Information and Communication Technologies: From Theory to Applications, 2004.*, pages 433–434.

Idri, A. and Zahi, A. (2013). Software cost estimation by classical and fuzzy analogy for web hypermedia applications: A replicated study. In *Proceedings of the 2013 IEEE Symposium on Computational Intelligence and Data Mining, CIDM 2013 - 2013 IEEE Symposium Series on Computational Intelligence, SSCI 2013*, pages 207–213. cited By 1.

Idri, A., Zahi, A., Mendes, E., and Zakrani, A. (2007). Software cost estimation models using radial basis function neural networks. In *Software Process and Product Measurement, International Conference, IWSM-Mensura 2007, Palma de Mallorca, Spain, November 5-8, 2007. Revised Papers*, volume 4895 of *Lecture Notes in Computer Science*, pages 21–31. Springer.

Idri, A., Zakrani, A., and Zahi, A. (2010). Design of radial basis function neural networks for software effort estimation. *International Journal of Computer Science*, 7(3).

Idri, A. e. a. (2002). Estimating software project effort by analogy based on linguistic values. In *8th IEEE International Software Metrics Symposium (METRICS 2002), 4-7 June 2002, Ottawa, Canada*, page 21. IEEE Computer Society.

Iwata, K., Nakashima, T., Anan, Y., and Ishii, N. (2010). Applying an artificial neural network to predicting effort and errors for embedded predicting effort and errors for embedded software development projects software development projects. *IEEJ Transactions on Electronics, Information and Systems*, 130(12):2167–2173.

Jones, C. (2007). Estimating software costs: bringing realism to estimating. In *McGraw-Hill, 2nd edn*.

Kamlesh, D., Varun, G., and S.D., V. (2019). Analysis and comparison of neural network models for software development effort estimation. *J. Cases Inf. Technol.*, 21(2):88–112.

Kanmani, S., Kathiravan, J., Senthil Kumar, S., and Shanmugam, M. (2008). Class point based effort estimation of OO systems using fuzzy subtractive clustering and artificial neural networks. In *Proceeding of the 1st Annual India Software Engineering Conference, ISEC 2008, Hyderabad, India, February 19-22, 2008*, pages 141–142. ACM.

Kaur, H. and Singh Salaria, D. (2013). Bayesian regularization based neural network tool for software effort estimation. *Global Journal of Computer Science and Technology*, 13(2).

Kitchenham, B. e. a. (2010). "systematic literature reviews in software engineering–a tertiary study". In *Inf. Softw. Technol.*

Kitchenham, B. et Charters, S. (2007). "guidelines for performing systematic literature reviews in software engineering". Technical report.

K.S., K. (2000). Neuro-genetic prediction of software development effort. *Inf. Softw. Technol.*, 42(10):701–713.

Kumar, G. and Kumar, B. (2014). Automation of software cost estimation using neural network technique. *International Journal of Computer Applications*, 98(20):11–17.

Kumar, K., Ravi, V., Carr, M., and Kiran, N. (2008). Software development cost estimation using wavelet neural networks. *Journal of Systems and Software*, 81:1853–1867.

Mendes, E. (2008). The use of bayesian networks for web effort estimation: further investigation. *In Proc. 8th Int Conf on Web Engineering, New York*, pages 203–216.

Oda, S. and Nakazato, O. (2016). An improved multilayer perceptron artificial neural network with genetic algorithm for software cost estimation. *International Journal of Academic Research in Computer Engineering*, 1(1):40–46.

Praynlin, E. and Latha, M. (2018). Performance analysis of FCM based ANFIS and ELMAN neural network in software effort estimation. *Int. Arab J. Inf. Technol.*, 15(1):94–102.

Sarno, R., Sidabutar, J., and Sarwosri (2015). Comparison of different neural network architectures for software cost estimation. In *IC3INA 2015 International Conference on Computer, Control, Informatics and Its Applications, Bandung, Indonesia*.

Satyananda Reddy, C. and Raju, K. (2009). A concise neural network model for estimating software effort. *International Journal of Recent Trends in Engineering*, 1(1).

Satyananda Reddy, C. and Raju, K. (2010). An optimal neural network model for software effort estimation. *Int. J. of Software Engineering*.

Shepperd, M. and Schofield, C. (1997). Estimating software project effort using analogies. *IEEE Transactions on Software Engineering*, 23(12):736–743.

Sivakumar, M. (2014). Enhancement of prediction accuracy in cocomo model for software project using neural network. In *Fifth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*.

Sommerville, I. (2010). Software engineering - 9th edition. In *Addison-Wesley*.

Suharjito, Nanda, S., and Soewito, B. (2016). Modeling software effort estimation using hybrid pso-anfis. In *iSITIAInternational Seminar on Intelligent Technology and Its Application*.

Tadayon, N. (2005). Neural network approach for software cost estimation. In *International Symposium on Information Technology: Coding and Computing (ITCC 2005), Volume 2, 4-6 April 2005, Las Vegas, Nevada, USA*, pages 815–818. IEEE Computer Society.

Tirimula, R., Chinnababu, K., Mall, R., and Satchidananda, D. (2013). A particle swarm optimized functional link artificial neural network (pso-flann) in software cost estimation. *FICTAProceedings of the International Conference on Frontiers of Intelligent Computing: Theory and Applications*, 199:59–66.

Tirimula, R., Satchidananda, D., Suresh, C., and Madhurakshara, S. (2012). Genetic algorithm for optimizing functional link artificial neural network based software cost estimation. In *Proceedings of the International Conference on Information Systems Design and Intelligent Applications (INDIA 2012)*, pages 75–82.

Wen, J., Li, S., Lin, Z., Huc, Y., and Huang, C. (2012). "systematic literature review of machine learning based software development effort estimation models". *Information and Software Technology*, 1(54).