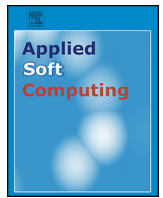




Contents lists available at [ScienceDirect](#)

Applied Soft Computing

journal homepage: [www.elsevier.com/locate/asoc](http://www.elsevier.com/locate/asoc)



# Predictive accuracy comparison between neural networks and statistical regression for development effort of software projects

Cuauhtémoc López-Martín\*

Information Systems Department, Universidad de Guadalajara, P.O. Box 45100, Jalisco, Mexico

## ARTICLE INFO

### Article history:

Received 5 July 2013  
Received in revised form 6 August 2014  
Accepted 27 October 2014  
Available online xxx

### Keywords:

Software development effort prediction  
Radial Basis Function Neural Network  
Feedforward multilayer perceptron  
General regression neural network  
Statistical regression  
ISBSG data set

## ABSTRACT

To get a better prediction of costs, schedule, and the risks of a software project, it is necessary to have a more accurate prediction of its development effort. Among the main prediction techniques are those based on mathematical models, such as statistical regressions or machine learning (ML). The ML models applied to predicting the development effort have mainly based their conclusions on the following weaknesses: (1) using an accuracy criterion which leads to asymmetry, (2) applying a validation method that causes a conclusion instability by randomly selecting the samples for training and testing the models, (3) omitting the explanation of how the parameters for the neural networks were determined, (4) generating conclusions from models that were not trained and tested from mutually exclusive data sets, (5) omitting an analysis of the dependence, variance and normality of data for selecting the suitable statistical test for comparing the accuracies among models, and (6) reporting results without showing a statistically significant difference. In this study, these six issues are addressed when comparing the prediction accuracy of a radial Basis Function Neural Network (RBFNN) with that of a regression statistical (the model most frequently compared with ML models), to feedforward multilayer perceptron (MLP, the most commonly used in the effort prediction of software projects), and to general regression neural network (GRNN, a RBFNN variant). The hypothesis tested is the following: the accuracy of effort prediction for RBFNN is statistically better than the accuracy obtained from a simple linear regression (SLR), MLP and GRNN when adjusted function points data, obtained from software projects, is used as the independent variable. Samples obtained from the International Software Benchmarking Standards Group (ISBSG) Release 11 related to new and enhanced projects were used. The models were trained and tested from a leave-one-out cross-validation method. The criteria for evaluating the models were based on Absolute Residuals and by a Friedman statistical test. The results showed that there was a statistically significant difference in the accuracy among the four models for new projects, but not for enhanced projects. Regarding new projects, the accuracy for RBFNN was better than for a SLR at the 99% confidence level, whereas the MLP and GRNN were better than for a SLR at the 90% confidence level.

© 2014 Elsevier B.V. All rights reserved.

## 1. Introduction

Since two of the three most important causes of Information Technology projects failure have been related to a poor resource prediction [1], the creation of accurate models to Software Development Effort Prediction (SDEP) represents a continuous activity of researchers and software managers [2]. In average, software developers expend from 30% to 40% more effort than is predicted [1]. Underpredicting software project effort causes schedule delays and cost over-runs, which may address to project failure and affect the

reputation and competitiveness of a company; on the other hand, overpredicting software project effort may address to an ineffective use of software development resources, which can result in missed opportunities to fund in other projects and therefore loss of project tenders [3,4]. These scenarios have motivated researchers in directing their efforts to determine which technique is more accurate for effort prediction, or to propose new or combined techniques that could provide better predictions [5]. These techniques can be classified into the following two general categories:

- 1) Expert judgment that implies a lack of analytical argumentation and aims to derive estimates based on experience of experts on similar projects. This technique is based on a tacit (intuition-based) quantification step [6].

\* Tel.: +52 3337703352.  
E-mail address: [cuauhtemoc@ucea.udg.mx](mailto:cuauhtemoc@ucea.udg.mx)

2) Model-based techniques which are based on a deliberate (mechanical) quantification step, and they can be divided into models based on Statistics (whose general form is a statistical regression model [7]) and models based on machine learning (ML) such as case-based reasoning, artificial neural networks, decision trees, Bayesian networks, support vector regression, genetic algorithms, genetic programming, and association rules [5].

These techniques have been applied to SDEP of projects developed at individual level in academic environments [8,9] or by teams [1,5]. This study involves projects developed by teams of practitioners in enterprises environments.

A systematic literature review of 84 primary studies specifically regarding the application of ML techniques to SDEP was structured from journals (70%), conferences (29%) and a book chapter [5]. This review revealed the following issues:

- 1) ML techniques have mainly been used in two forms between the years 1991 and 2010: either alone or as combination of two or more ML techniques, or as a combination with non-ML techniques.
- 2) Neural networks represent the second most used technique with 26%, after the case based reasoning technique with 37%.
- 3) Genetic algorithms have only been used in combination with other techniques such as case-based reasoning, artificial neural networks, and support vector regression. In the mentioned cases, genetic algorithms have been used for either feature weighting or feature selection.
- 4) As for weaknesses of techniques, decision trees is prone to overfitting on small training data set; case-based reasoning and neural networks cannot deal with categorical features in their standard forms; case-based reasoning, neural networks and decision trees cannot deal with missing values in their standard forms; and finally, neural networks are not easy to understand by practitioners.
- 5) As to strengths of techniques, case-based reasoning and decision trees are intuitive and are easy to understand; whereas neural networks have the ability to learn complex (non-linear) functions.
- 6) Regarding strengths and weaknesses of Bayesian networks, support vector regression, genetic programming, and association rules were not listed since none of them was mentioned by more than one study. That is, their application in the software development prediction is still scarce.

The SDEP accuracy results of ML techniques, so far, have been inconclusive and inconsistent. The SDEP accuracy varies under the same model when it is constructed with different historical project data sets or different experimental designs. This assertion was attributed to the limited number of primary studies [5]. Therefore, this research is also motivated by the suggestion generated from Ref. [5], in the sense of conducting more empirical studies to obtain stronger in evidence related to the performance of techniques.

With regard to neural networks, several kinds of them have been applied to SDEP. Until the year 2008, the feedforward multilayer perceptron (MLP) with back propagation learning algorithm was the neural network most commonly used in the SDEP [10], and this kind of neural networks have kept its use until the year 2009 [2], 2012 [11], and 2013 [12]. In this research, the application of an artificial neural network named Radial Basis Function Neural Network (RBFNN) is proposed. The RBFNN prediction accuracy is compared with that of a MLP and a GRNN (a variant of the RBFNN recently used for predicting the effort). The proposal of the RBFNN is based on the following arguments:

- 1) A survey of 96 studies comparing the performance between neural networks and statistical regression models in several fields, showed that neural networks outperformed the regression models in about 58% of the cases, whereas in 24% of the cases, the performance of the statistical models were equivalent to the neural networks [13].
- 2) The non-linear relationships are common among development effort and independent variables in software projects [14].
- 3) Neural networks are the most accurate models among ML techniques when they have been applied to SDEP [5].
- 4) In spite of the fact that neural networks have been shown to have a better prediction accuracy than regression models when they are applied to SDEP, the application of ML techniques in the industry is still limited [5].
- 5) The procedure for training a RBFNN is faster than that one used for training a MLP, due to the internal representation formed by the RBFNN hidden neurons which lead to a two-stage training procedure [15], whereas a RBFNN requires less substantial computation than a GRNN to evaluate new points [38].

There have been identified ten studies where a RBFNN was applied to SDEP. Six of them compared a RBFNN with a statistical regression [22–27], whereas the other four did not compare the accuracy of the RBFNN with any other technique [28–31]. In this research, the SDEP accuracy of the RBFNN is compared with the accuracy of a statistical regression. There are the following reasons for this comparison:

- 1) The use of non-ML (such as statistical regression) and ML techniques in parallel at the early stage of development is recommended [5].
- 2) A regression analysis allows selecting the statistically significant independent variables [16] that explain the dependent variable (development effort).
- 3) Statistical regressions are the models most frequently compared with ML models [5,13].

Neural networks and statistical models have been applied in fields such as accounting, finance, health, medicine, engineering, manufacturing or marketing [13] and specifically to SDEP [5]. The following six weaknesses have been identified in studies where the accuracy in these two models was compared with each other:

1. In spite of the Mean Magnitude of Relative Error (MMRE) is a biased predictor of central tendency of the residuals because of it is an asymmetric measure that leads to asymmetry [17], the 89% of the studies has used to the MMRE as criterion for evaluating the prediction accuracy of ML models [5].
2. Among the three dominant validation methods used for evaluating the precision accuracy of ML models are *Holdout* (38%), *Leave-one-out cross-validation* or LOOCV (37%), and *k-fold cross-validation* (19%) [5]. The *Holdout* and *k-fold cross-validation* methods use mainly a random selection for training and testing the models, which introduces the problem of conclusion instability of different studies. Therefore, methods that use a random selection have been deprecated for assessing SDEP models [18].
3. The determination of parameters for the neural networks like the number of hidden layers or number of neurons in each hidden layer is usually omitted [13].
4. Results obtained from the trained models were not tested on a new data set that was not used for training the models. The reason for testing on a separate data set is that otherwise, it would not have provided an unbiased prediction of the generalization error [15].
5. Most of the studies did not use the statistical techniques for analyzing the dependence, normality and variance of data. They

Download English Version:

<https://daneshyari.com/en/article/6905466>

Download Persian Version:

<https://daneshyari.com/article/6905466>

[Daneshyari.com](https://daneshyari.com)