An Evaluation of Neural Networks Approaches used for Software Effort Estimation

B.V. Ajay Prakash¹, D.V. Ashoka², V.N. Manjunath Aradhya³
¹ SJB Institute of Technology/Department of ISE, Bangalore, India
ajayprakas@gmail.com
² JSSATE, Dept. of Information Science and Engineering, Bangalore, India
der.ashok_research@hotmail.com
³ SJCE, Department of MCA, Mysore, India
aradhya1980@yahoo.co.in

Abstract—Demand for software is increasing day by day due to its more usage in IT industries. It is a mega challenge for software industry to develop very high quality software effectively within stipulated time and budget. To accomplish this challenge, the software development process needs to be effectively managed and well planned. It is very important to have good effort estimation in order to manage well budget. This paper aims to highlight the application of data mining techniques that yields high software effort estimation accuracy in contrast with other well established effort estimation models using little features. The effect of the proposed work discussed in this paper consisting of several steps. These steps can provide cost-saving project effort estimation means to identify and select only relevant and necessary features. By applying data mining techniques project managers and experts can consume less time to predict software project effort and more time on more important issues in releasing the project in time to customers. Further, this work illustrates the advantages like less computation time and effort resulting in energy saving mandates that can be easily adapted in software development process.

Index Terms—Data mining, effort estimation, neural networks, project planning.

I. INTRODUCTION

Estimating size or resources is one of the vital topics in software engineering and IT. During software development, effort estimation is the basic step taken in budgeting, planning and development of the software project. Exact estimation of software development effort, predicting and scheduling are the essential components to deliver the software product on time, to produce superior quality and contained by estimated cost. Effort estimation inaccuracy can be harmful to an IT industry’s economics and dependability due to behind schedule release, poor quality and stakeholder’s displeasure with the software product. From the Brazilian Ministry of Science and Technology-MCT research report in 2001, only 29% of the IT industry’s able to achieve size estimation and 45.7% of software effort estimation [1]. So in current years effort estimation has motivated as an extensive research. The key in component is developed kilo line of code (KLOC), which affects the software effort estimation which includes all program instructions [10]. Many software estimation models have been proposed to support project manager in accurate decision making about their projects [11, 12]. One of the known mathematical approaches for software cost estimation is the
COnstructive COst MOdel (COCOMO) model was first provided by Boehm [11], it was built based on 63 software projects. COCOMO model helps in finding the development time, the effort. Here E represents the evaluated software effort in man-months. In equation 1 a and b parameters depend mainly on the type of software project. Software projects were categorized based on the complexity of the project into three categories. They are:

\[ E = a \times \text{KLOC}^b \]

For Organic Model
\[ E = 2.4 \times \text{KLOC}^{1.03} \quad D = 2.5 \times (E^{0.36}) \]

For Semi-Detached Model
\[ E = 3.0 \times \text{KLOC}^{1.12} \quad D = 2.5 \times (E^{0.35}) \]

For Embedded Model
\[ E = 3.6 \times \text{KLOC}^{1.20} \quad D = 2.5 \times (E^{0.32}) \]

II. RELATED WORKS

Quite a lot of research works has been carried out on building efficient effort estimation using soft computing techniques [2] [3]. Kelly et al. [4] provides methodology for exploring software cost estimation using Neural Networks (NNs), Genetic Algorithms (GAs) and Genetic Programming (GP). Artificial neural network are used in effort estimation due to its ability to learn from previous data [5] [6]. It is also complex relationships between the dependent (effort) and independent variables (cost drivers). Many authors encompass neural network to effort estimation using feed-forward multi-layer Perceptron, Back propagation algorithm and sigmoid function [4]. Idri et al. [7] has made research on estimating software cost using the neural network and fuzzy logic rules on COCOMO81 dataset. Samson et al. [8] applied multilayer perception in software effort prediction for boehm’s COCOMO dataset also compares neural network with linear regression. Radlinki et al. [13], analyses the accuracy of predictions for software development effort using different machine learning techniques. Parag [14], proposed a Probabilistic Neural Networks (PNN) approach for simultaneously estimating values of software development parameters (either software size or software effort) and probability that the actual value of the parameter will be less than its estimated value. Attarzadeh et al. [15], the author’s presents that the one of the greatest challenges for software developers is predicting the development effort for a software system based on developer abilities, size, complexity and other metrics for the last decades. Stamata et al. [16], comparative research has been done by using three machine learning methods such as Artificial Neural Networks (ANNs), Case-Based Reasoning (CBR) and Rule Induction (RI) to build software effort prediction systems.

III. ESTIMATING DEVELOPMENT EFFORT USING NEURAL NETWORKS TOOLS

Different DM tools such as weka 3.6, neuro solutions evaluation version 6.29, and MATLAB neural network tool box used for software effort estimation. In this research work publicly available PROMISE repository (http://promisedata.org/data) is used. Table 2 shows the different data sets used for effort estimation.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Description</th>
<th>Features</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cocomonasa</td>
<td>NASA projects</td>
<td>17</td>
<td>60</td>
</tr>
<tr>
<td>Maxwell</td>
<td>Biggest commercial Banks projects in finland</td>
<td>27</td>
<td>62</td>
</tr>
<tr>
<td>China</td>
<td>Chinese software companies projects data</td>
<td>18</td>
<td>499</td>
</tr>
<tr>
<td>Nasa93</td>
<td>NASA projects</td>
<td>17</td>
<td>93</td>
</tr>
<tr>
<td>Desharnais</td>
<td>Canadian software projects</td>
<td>12</td>
<td>81</td>
</tr>
<tr>
<td>Cocomo81</td>
<td>Projects data from NASA</td>
<td>17</td>
<td>63</td>
</tr>
<tr>
<td>Kemerer</td>
<td>Large business applications</td>
<td>7</td>
<td>15</td>
</tr>
<tr>
<td>Telecom</td>
<td>Telecom companies projects</td>
<td>3</td>
<td>18</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td></td>
<td><strong>890</strong></td>
</tr>
</tbody>
</table>
The steps followed in calculating the software effort is as shown in Figure 2. In this work the application of neural network to effort estimation made use of feed-forward multi-layer Perceptron, Radial basis function network and Support vector machine.

![Fig.2. Steps in effort estimation](image)

![Fig.3. Multilayer Perceptron](image)

**A. Weka Tool: As a DM tool**

Weka is a tool for performing data mining tasks such as classification, association rules, clustering and visualization. Also contains many machine algorithms, data pre-processing. It can be directly applied to dataset or can be called by developing own java source code. In this experiment, used multilayer preceptron with setting the test cross validation folds to 10 and applied to different data set which is mentioned in table 2.

**B. Neuro Solutions**

Neuro Solutions is developed by NeuroDimension which provides the neural network development environment. It can be used as modula or icon based network design with user interface with implementation of different machine learning algorithms such as back propagation, Levenberg-Marquardt etc. Using Neuro Solution tool, we can design, train, test and deploy neural network (supervised learning and unsupervised learning) models to perform a wide variety of tasks such as data mining, classification, function approximation, multivariate regression and time-series prediction [17].

Figure 4, shows the neuro solution tool output. In neuro solution, performance of neural network results is evaluated using the magnitude of related error (MRE) and mean magnitude of relative error (MMRE). MRE equation 2 is defined as follows

\[
MRE = \frac{|ActualEffort - PredictedEffort|}{ActualEffort} \times 100
\]  

(2)
\[ MMRE = \frac{1}{n} \sum_{i=1}^{n} MRE_i \]  \hspace{1cm} (1)

Fig. 4. Neuro solution result

C. MATLAB

Using MATLAB R2011a, different data sets are trained and tested. China dataset is selected as input and target output is the effort. Figure 5 shows the neural network model used in the experiment and figure 6 shows the result of effort.

Fig. 5. Neural network training

Fig. 6. Result of effort

295
IV. CONCLUSIONS

Software effort estimation implementation in a large software development organization takes more than a year. Getting hold of estimation experience and integrating it into project management processes along with the consequent introduction of IT measurements for continuing improvement might require another couple of years. This paper has illustrated the application of different data mining tools for software effort estimation to reduce time and effort in software development. A neural network, Weka 3.6 tool and MATLAB neural network toolbox are used to different dataset of 890 projects in order to estimate efforts were discussed. The discussion made in this paper may use as a reference guide to software project managers in software effort estimation.

REFERENCES