We are IntechOpen, the world’s leading publisher of Open Access books
Built by scientists, for scientists

3,800
Open access books available

116,000
International authors and editors

120M
Downloads

154
Countries delivered to

TOP 1%
Our authors are among the most cited scientists

12.2%
Contributors from top 500 universities

WEB OF SCIENCE™
Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com
1. Introduction

Recently, growth in the information industry has caused a wide range of uses for information devices, and the associated need for more complex embedded software, that provides these devices with the latest performance and function enhancements (Hirayama (2004); Nakamoto et al. (1997)). Consequently, it is increasingly important for embedded software-development corporations to ascertain how to develop software efficiently, whilst guaranteeing delivery time and quality, and keeping development low costs (Boehm (1976); Tamaru (2004); Watanabe (2004)). Hence, companies and divisions involved in the development of such software are focusing on various types of improvement, particularly process improvement. Predicting effort requirements of new projects and guaranteeing quality of software are especially important, because the prediction relates directly to costs, while the quality reflects on the reliability of the corporation Komiyama (2003); N. (2004); Nakashima (2004); Ogasawara & Kojima (2003); Takagi (2003). In the field of embedded software, development techniques, management techniques, tools, testing techniques, reuse techniques, real-time operating systems and so on, have already been studied. However, there is little research on the relationship between the scale of the development and the number of errors, based of data accumulated from past projects. As a result, previously we described the prediction of the total scale using multiple regression analysis (Iwata et al. (2006b); Nakashima et al. (2006)) and collaborative filtering (Iwata et al. (2006a)). In this Chapter we therefore, propose a method for creating effort and errors prediction model using an Artificial Neural Network (ANN) for complementing missing values (Iwata et al. (2006a)). The proposed method calculates the amount of effort and the number of errors by the following 3 steps. The first step, the similarity between the complementary project data, which include missing values, and the complete project data is calculated. Next, applies collaborative filtering using the method Tsunoda et al. (Tsunoda et al. (2005)) to complement missing values in the data and thus produce sufficient amount of data. In the final step, the prediction target project effort (or errors) is calculated.
using the model that is derived from the ANN and with this data. However, the ANN has a large margin of errors for some projects. We therefore, propose a method to reduce the margin of errors model. Finally, we also compare the accuracy of the proposed ANN model with that of a multiple regression analysis model using Welch’s t-test (Student (1908); Welch (1947)).

The rest of the Chapter is organized as follows. In Section 2, we explain software development management and discuss current problems and the objectives which this study is trying to achieve, and illustrate software development process and selection of data to establish the model in Section 3. Then, Section 4 explains a collaborative filtering to complement missing values. In Section 5 we expound models to predict effort and errors. In section 6 describes evaluation experiment. Section 7 concludes.

2. Software project management and issues

The embedded software for financial institutions developed by “OMRON software Co.” is based on the basic software customized for an individual customer’s need to install it at various sites. To minimize the customization needed during the development of basic software, parameters are embedded to control the system. This engineering technique assures productivity and quality. This type of approach is actively taken during the process of software development. While using this type of technique, the pressures related to delivery time and quality are more and more intense. This requires improving further the quality and cost during software development. Hence, we have already studied costs of the processes by using analysis(Iwata et al. (2006b); Nakashima et al. (2006)) and collaborative filtering(Iwata et al. (2006a)). To cope with this situation, the tools that can manage the progress status or results in the database are used to improve the quality and productivity. However, the more the volume of software development project increases the more the errors increase. By analyzing the database, we determine the relationship between the volume of software development project and the errors.

3. Software development processes and selection of data

In software development division of “OMRON software Co.”, the waterfall model(Boehm (1976)) is used as the basic development-process model. A general description of this model is given in Table 1.

Regarding the data related to project productivity or quality, data for such things as internal effort and size information etc. are recorded as shown in Table 2 and Table 3, for 3 points in time:

1. At the beginning of the project.
2. During the project.
3. At the end of the project.

Before analyzing data, we examined the data and decided which data should be selected to make the model. Table 2 and Table 3 show the latter data and the results of the selection.

3.1 Data sets for creating models

Using the following data, we create models to predict both the planning effort ($Eff$) and errors ($Err$).

$Eff$: “The amount of effort”, which needs be predicted.
1 Conceptual design (CD)  
This is so-called "system engineering work". They analyze customer requirements and detail the areas to be addressed as development factors.

2 Design  
According to the development factors defined in CD process, designing of software functionality, combining of software modules, and writing of source code are performed.

3 Debugging  
Verify the outcome of the design process with the actual machine to see if it is designed according to the design. The same designer in design process is assigned to debug.

4 Test  
After finishing debugging, double-check the software to confirm that it satisfies customer's requirements. A different person (not those assigned to design and debug) is assigned to do this.

Table 1. Software Development Process

<table>
<thead>
<tr>
<th>Process</th>
<th>Contents of work</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conceptual design (CD)</td>
<td>This is so-called &quot;system engineering work&quot;. They analyze customer requirements and detail the areas to be addressed as development factors.</td>
</tr>
<tr>
<td>Design</td>
<td>According to the development factors defined in CD process, designing of software functionality, combining of software modules, and writing of source code are performed.</td>
</tr>
<tr>
<td>Debugging</td>
<td>Verify the outcome of the design process with the actual machine to see if it is designed according to the design. The same designer in design process is assigned to debug.</td>
</tr>
<tr>
<td>Test</td>
<td>After finishing debugging, double-check the software to confirm that it satisfies customer's requirements. A different person (not those assigned to design and debug) is assigned to do this.</td>
</tr>
</tbody>
</table>

Table 2. Classifications and Selection of Data

<table>
<thead>
<tr>
<th>Items</th>
<th>Data</th>
<th>Selection</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal effort</td>
<td>Effort for each planned process and actual performance</td>
<td>Selected</td>
<td>The data are acquired by effort management system. These data are quantitative and accuracy is good.</td>
</tr>
<tr>
<td>Project-scale information</td>
<td>Number of lines in new, modified, original and reused software</td>
<td>Selected</td>
<td>These data are quantitative and accuracy is good.</td>
</tr>
<tr>
<td>Effort information</td>
<td>Actual effort in each process</td>
<td>Rejected</td>
<td>The data are acquired at the end of the project. Hence, they can not use to predict effort and errors at the beginning of the project.</td>
</tr>
<tr>
<td></td>
<td>Effort for redo in each process</td>
<td>Rejected</td>
<td>The definition of redo is not consistent. The accuracy is poor.</td>
</tr>
<tr>
<td>Items</td>
<td>Data</td>
<td>Selection</td>
<td>Reason</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-------------------------------------------</td>
<td>-----------</td>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Products information</td>
<td>Product classification and product models</td>
<td>Selected</td>
<td>It is necessary to make characteristics of the products and development process be reflected in the model.</td>
</tr>
<tr>
<td>Customer name and sub project name</td>
<td></td>
<td>Rejected</td>
<td>The data is qualitative and it is difficult to obtain accurate data.</td>
</tr>
<tr>
<td>Development type(new or modification)</td>
<td></td>
<td>Rejected</td>
<td>Because there are only two types, it is not appropriate as parameter for the model.</td>
</tr>
<tr>
<td>Delivery time</td>
<td></td>
<td>Rejected</td>
<td>Because the delivery time is seldom changed, this is not selected.</td>
</tr>
<tr>
<td>Outsourcing</td>
<td>The estimation of outsourcing amount and actual situation</td>
<td>Rejected</td>
<td>Because outsourcing amount includes sales aspects, this is not appropriate for actual project error status.</td>
</tr>
<tr>
<td>Quality information</td>
<td>The number of problems in each process</td>
<td>Selected</td>
<td>It is necessary to find relationship between a project and errors. These data consist of “Total Error”, “Error in CD and Design”, “Error in Debugging” and “Error in Test”.</td>
</tr>
</tbody>
</table>
Err: “The number of errors” in a project.

V\textsubscript{new}: “Volume of newly added”, which denotes the number of steps in the newly generated functions of the target project.

V\textsubscript{modify}: “Volume of modification”, which denotes the number of steps modifying and adding to existing functions to use the target project.

V\textsubscript{survey}: “Volume of original project”, which denotes the original number of steps in the modified functions, and the number of steps deleted from the functions.

V\textsubscript{reuse}: “Volume of reuse”, which denotes the number of steps in functions of which only an external method of has been confirmed and which are applied to the target project design without confirming the internal contents.

4. Collaborative filtering

4.1 Conventional collaborative filtering and complementing values

Collaborative filtering is used as a basic technique in a system (here referred to as a “recommendation system”), that recommends items from a number of available options by matching user preferences (Breese et al. (2000); Tsunoda et al. (2005)). The items are any objects, for which the degree of preference changes according to the user, such as articles, web pages, books, songs, movies, and so on. A recommendation system based on collaborative filtering includes the following two steps:

1. Calculating similarity among users (user evaluation values are used in the calculation).
2. Determining items to be recommended (calculated values for the items are based on similarity). Herein, it is assumed that the preferences of users that are highly similar, will also be similar, and that new items are recommended to users.

This Chapter applies the method of conceptualizing from a recommendation system based on collaborative filtering to complement missing values, and references the effort and errors prediction method proposed by Tsunoda et al., (Tsunoda et al. (2005)). In other words, in this Chapter we calculate missing values based on the assumption that “if a project has any missing values, the values are similar to those of other projects that show striking similarities, because highly similar projects output similar values for each item.”. We use metrics to calculate the similarity among projects instead of user evaluation values. However, the range of a metric is different for each class, in contrast to user evaluation values that are all within a fixed range. Hence, the values by each metric are normalized to set its range. Moreover, the values by each metric rely on the scale of the project, and the dispersion of the average values by each metric is extremely large. Because of this, errors will be magnified if the scale of the project is not considered when calculating missing values. Therefore, to calculate missing values, we use revised values corresponding to the scale of the project, and do not use those projects that are too far apart on the scale, even if the similarity is high.

4.2 Complement method for missing value

In this Chapter, the matrix $m \times n$ denotes a data set including missing values. $p_i \in \{p_1, p_2, \ldots, p_m\}$ indicates the $i$th project, and $m_j \in \{m_1, m_2, \ldots, m_n\}$ indicates the $j$th metric. $v_{ij} \in \{v_{1,1}, v_{1,2}, \ldots, v_{m,n}\}$ means the value of the measurement in the $j$th matrix $m_j$ in the $i$th project $p_i$. When $v_{ij}$ is the missing value, it is denoted as $v_{ij} = \phi$.
Let the value of the bth metric \( m_b \) in the ath project \( p_a \) be a missing value \( v_{a,b} = \phi \) and \( \bar{v}_{a,b} \) mean the prediction value for the metric value \( v_{a,b} \). In calculating \( \bar{v}_{a,b} \) the following three steps are processed.

1. Metric normalization. The range of every metric range is \([0, 1]\) via normalization.
2. Calculation of similarity among projects.
3. Missing value calculation.

### 4.2.1 Metric normalization

The values of each metric are normalized to set the range for the metric. The range of every metric range is \([0, 1]\) via normalization. The normalized value for a metric of value \( v_{i,j} \) is denoted as \( f_{n}(v_{i,j}) \), and \( f_{n}(v_{i,j}) \) is calculated by the Eq. (1).

\[
f_{n}(v_{i,j}) = \frac{v_{i,j} - \min(P_j)}{\max(P_j) - \min(P_j)} \tag{1}
\]

where, \( P_j \) is the set of projects able to measure the value of the metric \( m_j \) and \( \max(P_j) \), \( \min(P_j) \) are the maximum and minimum values of \( \{v_{k,j} | p_k \in P_j\} \), respectively.

### 4.2.2 Calculation of similarity among projects

The similarity between the missing value prediction target project \( p_a \) and another project \( p_i \) is described as \( f_{sim}(p_a, p_i) \). The similarity calculation uses a vector calculation algorithm (Breese et al. (2000)).

The algorithm for calculating similarity is usually used to calculate the similarity between two documents (Salton & MacGill (1983)). In the algorithm, each vectors contain the frequency of words appearing in each document, and similarity is calculated using the cosine of the angles created by the vectors. Breese et al. (Breese et al. (2000)) proposed a recommendation system based on this algorithm, in which they equate a document with a user, the words with items, and the word frequency with the item evaluation value. In this Chapter, we calculate the similarity of projects by equating the user with the project, the item with the metric, and the item prediction value with the metric value similar to the method of Tsunoda et al. (Tsunoda et al. (2005)).

The similarity \( f_{sim}(p_a, p_i) \) between the prediction target project \( p_a \) and the another project \( p_i \) is calculated as in Eq. (2).
Applying an Artificial Neural Network to Predicting Effort and Errors for Embedded Software Development Projects

\[ f_{\text{sim}}(p_a, p_i) = \frac{\sum_{j \in M_a \cap M_i} f_n(v_{a,j}) \times f_n(v_{i,j})}{\sqrt{\sum_{j \in M_a \cap M_i} f_a(v_{a,j})^2} \sqrt{\sum_{j \in M_a \cap M_i} f_n(v_{i,j})^2}} \]  

where, \( M_a \) and \( M_i \) are the set of non missing measured in projects \( p_a \) and \( p_i \) respectively. The value range for the similarity \( f_{\text{sim}}(p_a, p_i) \) is [0, 1].

4.2.3 Missing value calculation

The prediction value \( \bar{v}_{a,b} \) of a missing value \( v_{a,b} \) is calculated using the similarity \( f_{\text{sim}}(p_a, p_i) \). It is necessary to consider the scale of projects in higher similarity to \( p_a \) to calculate \( \bar{v}_{a,b} \) because only the vector angles are used in the similarity calculation and vector size is not taken into account (Tsunoda et al. (2005)). Hence, for the calculation of missing values, the project scale reviser \( \text{amplifier} : f_{\text{amp}}(p_a, p_i) \) is used as a weight. Furthermore, if \( f_{\text{amp}}(p_a, p_i) \) exceeds the constant value \( \text{ampmax} \), the project \( p_i \) is not used in the calculation of \( \bar{v}_{a,b} \), since the project scale is considered too different even if the similarity is high. \( \bar{v}_{a,b} \) is calculated from Eq. (3).

\[ \bar{v}_{a,b} = \frac{\sum_{i \in \text{knP}} v_{i,b} \times f_{\text{amp}}(p_a, p_i) \times f_{\text{sim}}(p_a, p_i)}{\sum_{i \in \text{knP}} f_{\text{sim}}(p_a, p_i)} \]  

where, \( \text{knP} \) means a set that has the \( k \) projects with a high similarity to project \( p_a \) without the value of \( \text{amplifier} \) exceeding \( \text{ampmax} \). \( f_{\text{amp}}(p_a, p_i) \) is calculated from Eq. (4).

\[ f_{\text{amp}}(p_a, p_i) = \begin{cases} r_n & \text{if } h = 2n-1 \\ \frac{r_n + r_{n+1}}{2} & \text{if } h = 2n \end{cases} \]  

where, \( h \) is the number of the product set of \( M_a \) and \( M_i \), and \( r_j \) equals \( \frac{f_n(v_{a,j})}{f_n(v_{i,j})} \), that is, the ratio of the values of the metric \( m_j \) in projects \( p_a \) and \( p_i \).

5. Effort and error prediction models

5.1 An artificial neural network model

Artificial Neural Networks (ANNs) are essentially simple mathematical models defining function.

\[ f : X \rightarrow Y \]  

where \( X = \{ x_i | 0 \leq x_i \leq 1, i \geq 1 \} \) and \( Y = \{ y_i | 0 \leq y_i \leq 1, i \geq 1 \} \).

ANNs are non-linear statistical data modeling tools and that can be used to model complex relationships between inputs and outputs. The basic model is illustrated in Fig. 2, in which the output is calculated as follows.
1. Calculating values for hidden nodes. The value of Hidden Node\(_j\) is calculated using the following equation:

\[
\text{Hidden Node}_j = f \left( \sum_i (w_{i,j} \times \text{Input}_i) \right)
\]  

(6)

where \(f(x) = \frac{1}{1 + \exp(-x)}\) and the \(w_{i,j}\) is weight calculated by the learning algorithm.

2. Calculating Output using Hidden Node\(_j\) as follows:

\[
\text{Output} = f \left( \sum_k (w'_{k,j} \times \text{Hidden Node}_k) \right)
\]  

(7)

where \(f(x) = \frac{1}{1 + \exp(-x)}\) and the \(w'_{k,j}\) is weight calculated by the learning algorithm.

We can use an ANN to create effort and error prediction models.

---

**Fig. 2. Basic Artificial Neural Network**

---

www.intechopen.com
5.1.1 Normalization of data

In an ANN, a range of input values or output values is usually less than or equal to 1 and greater than or equal to 0. However, most selected data are greater than 1. Each data range is, therefore, converted to [0, 1] by normalization. The normalized value for \( t_{kind} \) is expressed as \( f_n(t_{kind}) \) (where \( kind \) denotes \( Eff, Err, V_{new}, V_{modify}, V_{survey} \) and \( V_{reuse} \)). The normalized value \( f_n(t_{kind}) \) is calculated using Eq. (8), which is the same as Eq. (1).

\[
 f_n(t_{kind}) = \frac{t_{kind} - \min(T_{kind})}{\max(T_{kind}) - \min(T_{kind})} \tag{8}
\]

where \( T_{kind} \) denotes the set of \( t_{kind} \) and \( \max(T_{kind}) \) and \( \min(T_{kind}) \) denote the maximum and minimum values, respectively, of \( T_{kind} \).

The normalization is flat and smooth, then, a small change in a normalized value influences a small-scale project to a greater degree than a large scale project. For example, let \( \min(T_{Eff}) = 10, \max(T_{Eff}) = 300, t_{Eff1} = 15, t_{Eff2} = 250 \), predicted value for \( t_{Eff1} \) be \( \hat{t}_{Eff1} \) and \( t_{Eff2} \) be \( \hat{t}_{Eff2} \). If the prediction model has +0.01 error, then \( f_n^{-1}(0.01) = 2.90 \). The predicted values result in \( \hat{t}_{Eff1} = 17.90 \) and \( \hat{t}_{Eff2} = 252.90 \). Both cases have same errors, but their absolute of the relative errors (ARE) are follows:

\[
 ARE_{Eff1} = \left| \frac{\hat{t}_{Eff1} - t_{Eff1}}{t_{Eff1}} \right| = \left| \frac{17.90 - 15}{15} \right| = 0.1933
\]

\[
 ARE_{Eff2} = \left| \frac{\hat{t}_{Eff2} - t_{Eff2}}{t_{Eff2}} \right| = \left| \frac{252.90 - 250}{250} \right| = 0.0116
\]

The results indicate the absolute of the relative errors of former is greater than that of the latter. These distributions for the amount of effort and the number of errors indicate the small-scale projects are major and more than the large scale projects. Therefore, in order to improve prediction accuracy, it is important to reconstruct the normalizing way.

5.2 New normalization of data

In order to solve the problem, we adopt new normalizing way in the following equation:

\[
 f_n(t) = \sqrt{1 - (f_n(t) - 1)^2} \tag{9}
\]

The comparison between Eq. (8) and Eq. (9) is shown in Figure 3 and 4. The Eq. (9) has a sharp inclination at the lower original data, then a small change at the lower original data get magnified.

Using the same assumption, the predicted values result in \( \hat{t}_{Eff1} = 15.56 \) and \( \hat{t}_{Eff2} = 271.11 \). Their absolute of the relative errors are in Eq. (10) and Eq. (11).
The results show the absolute of the relative errors for the small-scale project is smaller than that of old normalization method and, in contrast, that for the large scale project is slightly larger than that of old normalization method. The more detailed comparison analyses are in Section 6.

5.2.1 Structure of model
In a feed-forward ANN, the information is moved from input nodes, through the hidden nodes to the output nodes. The number of hidden nodes is important, because if the number is too large, the network will over-training. The number of hidden nodes is, generally 2/3 of the number of input nodes or twice the number of input nodes. In this Chapter, we use 8 hidden nodes in our model which is illustrated in Fig. 5.
5.3 Multiple regression analysis model
The multiple regression analysis (MRA) model is derived from Eq. (12), in which the notation adheres to the meanings defined in Subsection 3.1.

\[ D = \alpha_1 \times S^2 + \alpha_2 \times S + \beta \] (12)

where, \( D \) indicates \( \text{Eff or Err} \), and \( S \) is calculated by Eq. (13).

\[ S = V_{\text{new}} + V_{\text{modify}} + \theta_1 \times V_{\text{survey}} + \theta_2 \times V_{\text{reuse}} \] (13)

where, \( \theta_1 \) and \( \theta_2 \) are less than 1, thus Eq. (13) emphasizes \( V_{\text{new}} \) and \( V_{\text{modify}} \).

6. Evaluation experiment
6.1 Evaluation criteria
Equations (14) to (16) are used as evaluation criteria for the effort and errors prediction models. The smaller the value of each evaluation criterion, the higher is the relative accuracy in Eqs. (14) to (16). The accuracy value is expressed as \( X \), and the predicted value as \( \bar{X} \). Also, the number of data is expressed as \( n \).

1. Mean of Absolute Errors (MAE).
2. Standard Deviation of Absolute Errors (SDAE).
3. Mean of Relative Errors (MRE).

\[
MAE = \frac{1}{n} \sum |\hat{X} - X| \tag{14}
\]

\[
SDAE = \sqrt{\frac{1}{n-1} \sum \left( |\hat{X} - X| - MAE \right)^2} \tag{15}
\]

\[
MRE = \frac{1}{n} \sum \frac{|\hat{X} - X|}{X} \tag{16}
\]

\[
SDRE = \sqrt{\frac{1}{n-1} \sum \left( \frac{\hat{X} - X}{X} - MRE \right)^2} \tag{17}
\]

6.2 Data used in evaluation experiment
The evaluation experiment uses the complemented data for projects including missing values by the method described in Subsection 4.2. This complemented project data is divided into
two random sets. One of the two sets is used as training data, while the other is test data. The training data is used the generation of the effort (or errors) prediction model generation, which is used to predict the effort (or errors) of the projects in the test data. The prediction criteria presented in Subsection 6.1 are then used to confirm whether the effort were accurately predicted or not by. Both data sets, that is, the training data and test data, are divided into 5 sections and these are used to repeat the experiment 5 times.

6.3 Results and discussion
A total of 73 projects were used in the experiment, of which 53 included missing values. Missing values were added to these 53 projects. For each method, averages of the experiments results for the 5 experiments are shown in Table 5.

<table>
<thead>
<tr>
<th></th>
<th>MAE</th>
<th>SDAE</th>
<th>MRE</th>
<th>SDRE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANN</td>
<td>17.440</td>
<td>37.679</td>
<td>0.69892</td>
<td>0.67254</td>
</tr>
<tr>
<td>MRA</td>
<td>30.825</td>
<td>55.643</td>
<td>0.98884</td>
<td>0.98928</td>
</tr>
</tbody>
</table>

Table 4. Experimental Results for Errors Prediction

<table>
<thead>
<tr>
<th></th>
<th>MAE</th>
<th>SDAE</th>
<th>MRE</th>
<th>SDRE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANN</td>
<td>11.345</td>
<td>17.403</td>
<td>0.25536</td>
<td>0.33830</td>
</tr>
<tr>
<td>MRA</td>
<td>24.962</td>
<td>11.941</td>
<td>0.89056</td>
<td>0.91893</td>
</tr>
</tbody>
</table>

Table 5. Experimental Results for Efforts Prediction

6.3.1 Validation analysis of the accuracy of the models
We compare the accuracy of the ANN model with that of the regression analysis model using Welch’s t-test (Welch (1947)). The t-test (called Student’s t-test)(Student (1908)) is used as a test of the null hypothesis that the means of two normally distributed populations are equal. Welch’s t-test is used when the variances of two samples are assumed to be different to test the null hypothesis that the means of non two normally distributed populations are equal if the two sample sizes are equal (Aoki (n.d.)). The t statistic to test whether the means are different is calculated as follows:

\[
t_0 = \frac{\overline{X} - \overline{Y}}{\sqrt{\frac{s_x^2}{n_x} + \frac{s_y^2}{n_y}}}
\]

where \(\overline{X}\) and \(\overline{Y}\) are the sample means, \(s_x\) and \(s_y\) are the sample standard deviations and \(n_x\) and \(n_y\) are the sample sizes. For use in significance testing, the distribution of the test statistic is approximated as an ordinary Student’s t-distribution with the following degrees of freedom:

\[
v = \frac{(\frac{\overline{X} - \overline{Y}}{\sqrt{\frac{s_x^2}{n_x} + \frac{s_y^2}{n_y}}})^2}{\frac{s_x^2}{n_x(n_x-1)} + \frac{s_y^2}{n_y(n_y-1)}}
\]

Thus once the a t-value and degrees of freedom have been determined, a p-value can be found using a table of values from the Student’s t-distribution. If the p-value is smaller than or equal
to the significance level, then the null hypothesis is rejected. The significance levels are usually 0.05 and 0.01, are represented by the Greek symbol, $\alpha$.

The null hypothesis, in these cases, is “there is no difference between the means of the prediction errors for the ANN model and the MRA model”. The results of the $t$-test for absolute errors and relative errors are given in Tables 6 and 7, are given in Tables 8 and 9, respectively.

<table>
<thead>
<tr>
<th></th>
<th>ANN Model</th>
<th>MRA Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean ($\bar{X}$)</td>
<td>17.440</td>
<td>30.825</td>
</tr>
<tr>
<td>Standard deviation(s)</td>
<td>37.679</td>
<td>55.643</td>
</tr>
<tr>
<td>Sample size ($n$)</td>
<td>189</td>
<td>189</td>
</tr>
<tr>
<td>Degrees of freedom ($\nu$)</td>
<td>362.565</td>
<td>362.565</td>
</tr>
<tr>
<td>$t$ value ($t_0$)</td>
<td>19.0483</td>
<td></td>
</tr>
<tr>
<td>$p$ value</td>
<td>$2.2 \times 10^{-16}$</td>
<td></td>
</tr>
</tbody>
</table>

Table 6. Results of $t$-test for MAE for Effort

<table>
<thead>
<tr>
<th></th>
<th>ANN Model</th>
<th>MRA Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean ($\bar{X}$)</td>
<td>0.69892</td>
<td>0.98884</td>
</tr>
<tr>
<td>Standard deviation(s)</td>
<td>0.67254</td>
<td>0.98928</td>
</tr>
<tr>
<td>Sample size ($n$)</td>
<td>189</td>
<td>189</td>
</tr>
<tr>
<td>Degrees of freedom ($\nu$)</td>
<td>362.82</td>
<td></td>
</tr>
<tr>
<td>$t$ value ($t_0$)</td>
<td>3.0918</td>
<td></td>
</tr>
<tr>
<td>$p$ value</td>
<td>0.002143</td>
<td></td>
</tr>
</tbody>
</table>

Table 7. Results of $t$-test for MRE for Effort

<table>
<thead>
<tr>
<th></th>
<th>ANN Model</th>
<th>MRA Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean ($\bar{X}$)</td>
<td>11.345</td>
<td>24.962</td>
</tr>
<tr>
<td>Standard deviation(s)</td>
<td>17.403</td>
<td>11.941</td>
</tr>
<tr>
<td>Sample size ($n$)</td>
<td>189</td>
<td>189</td>
</tr>
<tr>
<td>Degrees of freedom ($\nu$)</td>
<td>363.409</td>
<td></td>
</tr>
<tr>
<td>$t$ value ($t_0$)</td>
<td>34.5583</td>
<td></td>
</tr>
<tr>
<td>$p$ value</td>
<td>$2.2 \times 10^{-16}$</td>
<td></td>
</tr>
</tbody>
</table>

Table 8. Results of $t$-test for MAE for Errors

<table>
<thead>
<tr>
<th></th>
<th>ANN Model</th>
<th>MRA Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean ($\bar{X}$)</td>
<td>0.25536</td>
<td>0.89056</td>
</tr>
<tr>
<td>Standard deviation(s)</td>
<td>0.33830</td>
<td>0.91893</td>
</tr>
<tr>
<td>Sample size ($n$)</td>
<td>189</td>
<td>189</td>
</tr>
<tr>
<td>Degrees of freedom ($\nu$)</td>
<td>309.901</td>
<td></td>
</tr>
<tr>
<td>$t$ value ($t_0$)</td>
<td>7.7881</td>
<td></td>
</tr>
<tr>
<td>$p$ value</td>
<td>$1.031 \times 10^{-13}$</td>
<td></td>
</tr>
</tbody>
</table>

Table 9. Results of $t$-test for MRE for Errors
The results indicate that the means of the absolute (or relative) errors between ANN models and MRA model shows a statistically significant difference, because the $p$-values are less than 0.01.

7. Conclusion

In this Chapter, we have established effort and errors prediction models using artificial neural networks for complementing missing values. The proposed method calculated the amount of effort and the number of errors by the following 3 steps.

1. Calculating the similarity between the complementary projects data (which include missing values) and the complete project data,
2. Applying collaborative filtering to complement missing values in the data and thus produce sufficient amount of data,
3. Creating models to predict target project effort (or errors) by the ANN and with this data.

In addition, we carried out an evaluation experiment that compared the accuracy of the ANN model with that of the MRA model using Welch’s $t$-test. The results of the comparison indicate that the ANN model is more accurate than the MRA model, because the mean errors of the ANN are statistically significantly lower.

Our future works are the following:

1. In this study, we used a basic artificial neural network. More complex models need to be considered to improve the accuracy by avoiding over-training.
2. We implemented a model to predict the final amount of effort and number of errors in new projects. It is also important to predict effort and errors mid-way in the development process of a project.
3. We used all the data in implementing the model. However, the data include exceptions and there are harmful to the model. Data needs to be clustered in order to identify these exceptions.
4. Finally, more data needs to be collected from completed projects.

8. References


Welch, B. L. (1947). The generalization of student’s problem when several different population variances are involved, *Biometrika* 34(28).
This book covers 27 articles in the applications of artificial neural networks (ANN) in various disciplines which include business, chemical technology, computing, engineering, environmental science, science and nanotechnology. They modeled the ANN with verification in different areas. They demonstrated that the ANN is very useful model and the ANN could be applied in problem solving and machine learning. This book is suitable for all professionals and scientists in understanding how ANN is applied in various areas.

How to reference
In order to correctly reference this scholarly work, feel free to copy and paste the following:
