*Journal of Universal Computer Science, vol. 18, no. 14 (2012), 2041-2070*
*submitted: 20/4/11, accepted: 26/7/12, appeared: 28/7/12* © *J.UCS*

# Software Cost Modelling and Estimation Using Artificial Neural Networks Enhanced by Input Sensitivity Analysis

**Efi Papatheocharous**
(University of Cyprus, Nicosia, Cyprus
efi.papatheocharous@cs.ucy.ac.cy)

**Andreas S. Andreou**
(Cyprus University of Technology, Lemesos, Cyprus
andreas.andreou@cut.ac.cy)

**Abstract:** This paper addresses the issue of Software Cost Estimation (SCE) providing an alternative approach to modelling and prediction using Artificial Neural Networks (ANN) and Input Sensitivity Analysis (ISA). The overall aim is to identify and investigate the effect of the leading factors in SCE, through ISA. The factors identified decisively influence software effort in the models examined and their ability to provide sufficiently accurate SCEs is examined. ANN of variable topologies are trained to predict effort devoted to software development based on past (finished) projects recorded in two publicly available historical datasets. The main difference with relevant studies is that the proposed approach extracts the most influential cost drivers that describe best the effort devoted to development activities using the weights of the network connections. The approach is validated on known software cost data and the results obtained are assessed and compared. The ANN constructed generalise efficiently the knowledge acquired during training providing accurate effort predictions. The validation process included predictions with only the most highly ranked attributes among the original cost attributes of the datasets and revealed that accuracy performance was maintained at same levels. The results showed that the combination of ANN and ISA is an effective method for evaluating the contribution of cost factors, whereas the subsets of factors selected did not compromise the accuracy of the prediction results.

**Keywords:** Software Cost Estimation, Artificial Neural Networks, Input Sensitivity Analysis
**Categories:** D.2.8, D.2.9

## 1 Introduction

Over the last four decades a plethora of Software Cost Estimation (SCE) methods and modelling techniques has been proposed in the international literature [Boehm, 00; Moløkken, 03; Jørgensen, 07]. The vast variety of such techniques on one hand, and their inconclusive results so far on the other, reveal the complexity of the development process and highlight the difficulties of producing accurate and reliable cost approximations. SCE involves the activity to calculate, with certain confidence, the resources required to develop software systems which are associated with the total person-months of the effort required. For estimating software effort various project parameters, usually called software cost drivers, need to be considered. However, the definitions of many of the cost drivers are not easy to define and sometimes are regarded as highly ambiguous and difficult to measure due to their dependence on

subjective notions and due to the intangible nature of software [Sommerville, 07].

In fact, numerous difficulties have been identified in the process of accurate and reliable SCE. Most difficulties concern practical measurement and modelling issues [Briand, 00] while the high complexity and uniqueness of the software engineering process is being the leading obstacle for achieving consistently successful estimations. In addition, two software systems are never identical; they may need to run on unfamiliar platforms, or use new technologies. Moreover, their development undergoes new processes and usually different people are involved. All the above cause high uncertainty during the initial project phases as many of the project parameters are undefined or unknown. The process of software development involves many inter-twined factors, which affect development effort, quality and productivity and whose relationships are not well understood or easily studied. Also, there is lack of trained estimators with the necessary expertise and knowledge to support the estimation process [Leung, 02], whereas the number of active researchers with long-term interest on SCE is low compared to other research topics and approaches within the software engineering discipline [Jørgensen, 07]. Practically, this suggests that extensive research of the many correlated factors contributing to software effort and their inter-relations is a very tedious set of tasks, while many researchers emphasise the need of automating such a process.

Several models and tools developed for SCE use a set of measures that describe a software project and provide an estimation of the associated effort. However, since the influencing factors of cost and their relationships are not well understood, many existing models aim to improve forecasting ability without conducting any form of analysis of the input variables [Park, 08]. Furthermore, most of the SCE models encounter the following difficulty: In order to make a reliable estimation they require information that is not known at the initiation of the project, while project managers and engineers try to specify exact values for the metrics used as inputs [MacDonell, 97]. Since for many of these metrics the actual values are never known with certainty until the project is completed, managers often assume values they anticipate [Jørgensen, 04]. As an alternative, cost data values may be collected from past completed projects and be utilised as future cases in an analogy-based method [Chiu, 07] according to a set of project characteristics. While the former situation suffers from subjectivity, the latter does not guarantee that a project under development will require the same amount of effort with that of a 'similar' project with respect to specific characteristics.

Taking the above problems into consideration, this paper focuses on the combination of Computational Intelligent methods, such as Artificial Neural Networks (ANN) [McCulloch, 43; Haykin, 99] with Input Sensitivity Analysis (ISA) to assess and rank the significance of a set of attributes used as inputs in the models based on the internal weight values assigned to ANN after training is concluded. The aim is to capture and examine the interactions between the influencing cost factors and effort and utilise the input's degree of influence built within the network to extract the most influential cost factors. In the use of ANN for SCE an important step is to identify the dominant factors, or attributes, that affect development effort [Park, 08]. Even though a number of measures have been reported to determine the significance of ANN input attributes [Garson, 91; Belue, 95; Glorfeld, 95; Satizábal, 07] they have never been employed on software cost drivers.

In this work the proposed process of selecting the most significant attributes, meaning those that most highly affect development costs, and then, using them to investigate the behaviour of ANN in SCE is proven a practical way to reduce the model's input space (and thus computational complexity and human effort) while maintaining the same levels of effort prediction accuracy. The approach offers a simple and reliable automatic method to reach to a usable influential subset of project attributes which may be more feasible to measure, collect and maintain. Another contribution of this work lies with the revelation and exploitation of strong correlations among a selected set of cost attributes and development effort recorded in past projects. Numerical and ordinal values from completed project data are used and their performance on accurate SCE is evaluated based on specific evaluation criteria. The core of the proposed approach is a series of empirical experiments employing ANN with different feedforward Multi-Layer Perceptron (MLP) topologies [McCulloch, 43; Karray, 04] aiming at improving the internal and external validity (generalisation) of the model. The prediction ability of ANN is also compared to a Multiple Linear Regression (MLR) model.

The rest of the paper is organised as follows: Section 2 presents a review on SCE research and techniques from the area of Computational Intelligence. Section 3 provides a description of the datasets used and explains the stages of the proposed methodology for creating Computational Intelligent SCE, along with the design principles of the experimental process. Section 4 presents the experimental results and summarises the main findings of this work. The section closes with an outline of possible threads to the validity of the proposed approach. Section 5 draws the concluding remarks and suggests some directions for future research.

## 2 Computational Intelligence Techniques in SCE Research

During the last decades, extensive research has been conducted in SCE resulting in the development of various estimation techniques and models. Several data-driven techniques in the area of Artificial Intelligence and Soft Computing, such as Artificial Neural Networks (ANN) and Evolutionary Algorithms (EA), have been investigated, as they presented several advantages over other, parametric approaches like Regression. The main advantage is that they usually make minimal or no assumptions at all regarding the mathematical function for describing the behaviour of effort in relation to a set of cost attributes and present high adaptability within the environments examined. A lot of studies (presented in a subsequent section) rely on non-parametric methods, such as ANN, and present comparative or improved results to traditional methods. Nevertheless, to the best of our knowledge, none of these studies addresses the issue of analysing how ANN store the knowledge gained through the iterative mapping of input patterns to the output samples. This knowledge is represented mainly by the synaptic weights of the individual neurons. Thus, it would be very interesting to analyse the 'black box' nature of ANN and the contribution of the independent variables to the prediction process, by examining the strength of the connections between the neurons, which originate from the inputs and propagate to the output, revealing their overall significance. The following section presents a brief literature overview related to studies that employ ANN for SCE. The advantages of using ANN include the ability to deal with domain complexity, noisy or

distorted data and generalise the knowledge gained, along with adaptability, flexibility and parallel processing [Haykin, 99].

### 2.1 Artificial Neural Networks and SCE: A Literature Overview

A wide range of studies have conducted research to approximate effort and compare various techniques. We begin our review with some of the early studies: The work of Serluca [Serluca, 95] compared the results of three methods, regression, analogy and ANN, using the MERMAID-2 dataset for effort estimation. The ANN achieved far more superior results compared to regression and marginally better than analogy when the dataset was fully used. However, when the dataset was separated into two more homogenous and therefore smaller clumps, the ANN performed very poorly, while the other two methods improved considerably. This led the author to conclude that ANN require large training sets before they can provide accurate predictions.

Srinivasan and Fisher [Srinivasan, 95] compared ANN and regression trees for predicting effort reported in the Kemerer dataset, using the COCOMO dataset for training. The results of the experiments were in favour of ANN.

Jørgensen [Jørgensen, 95] reported four modelling approaches to estimate maintenance effort: Regression, ANN, a form of pattern recognition and a simple baseline rule of thumb model according to which, "effort is equal to size divided by the mean productivity". The study used a MLP with a back-propagation training algorithm on the Jørgensen95 dataset and the ANN was found to perform worse than the best regression model in terms of the *MMRE*, but very successfully in terms of the *Pred(0.25)* metric (for the description of evaluation metrics refer to section 3.3). This leads to the conclusion that the selection of evaluation criteria is very important in the assessment of SCE models.

Wittig and Finnie [Wittig, 97] compared a back-propagation MLP ANN with CBR (Case Based Reasoning or analogy) using the Desharnais dataset [Desharnais, 89] and 136 sample observations from the Australian Software Metrics Association (ASMA) to estimate effort. In this work the ANN yielded very encouraging results, but only the attribute of system size was utilised to provide predictions of effort. Trials conducted to test the model combining other attributes resulted in reduced prediction errors, which suggested that there is room for further investigation and improvement through a more systematic study of the development characteristics.

Samson et al. [Samson, 97] developed an Albus MLP to predict software effort, which operates in a similar way to a lookup table, using a generalisation mechanism so that a solution learned at one point in the input space influenced solutions at neighbouring points. Different ANN were then compared with linear regression. Although predictions made by the ANN outperformed those produced by linear regression using the COCOMO dataset, for some projects both techniques performed poorly. Thus, accurately performing SCE in every single case is usually not feasible.

Hughes [Hughes, 97] compared a wide range of approaches for effort estimation including analogy, regression, and ANN, using the WSD1 dataset. The dataset was initially divided into two homogenous groups. When the two groups were merged the *MMRE* was improved, reinforcing the fact that ANN can perform well when presented with larger datasets, while, at the same time, performance of other techniques, including analogy and regression, deteriorated.

Mair et al. [Mair, 00] evaluated predictions of effort using regression, rule induction, CBR and ANN models, which showed considerable variations, but concluded that ANN was the most accurate model. Although the datasets utilised had different characteristics, like the number of features and the number of projects, and, additionally, the dataset presented outliers, collinearity, total convergence was finally obtained.

MacDonell and Gray [MacDonell, 97] compared the FP (Function Points) method, Least Squares (LS) regression and ANN for effort prediction on the Desharnais dataset and indicated that the most accurate model was the ANN. The authors attribute this success to the non-linearity and interactions within the data.

Heiat [Heiat, 02] compared the effort prediction performance of a MLP and Radial Basis Function Networks (RBFN) to that of regression analysis and found that when a set of project data implemented with a third generation language was used the ANN performed equally well with regression. However, when a combined third and fourth generation languages dataset was used ANN outperformed regression.

Idri et al. [Idri, 02] conducted two experiments for effort estimation using a back-propagation trained MLP on the COCOMO ’81 dataset, the outputs of which were mapped to a fuzzy rule-based system. Their results indicated poor accuracy performance, while an important issue was ANN overfitting which was not sufficiently handled since 300,000 iterations were executed on just a small set of samples (63 samples). The same authors investigated the use and interpretation of RBFN in SCE by mapping the ANN to a fuzzy rule-based system [Idri, 04]. Results on the COCOMO ’81 dataset indicated that the accuracy of the ANN depended heavily on the parameters of the middle layer and more specifically on the number of hidden neurons and the weight values.

Kumar et al. [Kumar, 08] used Wavelet Neural Networks (WNN) for SCE and compared the effectiveness with MLP, RBFN, Multiple Linear Regression (MLR), Dynamic Evolving Neuro-Fuzzy Inference System (DENFIS) and Support Vector Machines (SVM) in terms of the *MMRE*. WNN seemed to outperform all other techniques.

Tronto et al. [Tronto, 08] investigated the application of ANN and stepwise regression for SCE. The experiments were conducted on the COCOMO dataset employing categorical variables whose impact was identified based on the work of Angelis et al. [Angelis, 01] forming new categorical values. It was observed that there is a strong relationship between the success of a technique and the size of the learning dataset, the nature of the function for cost and other dataset characteristics (such as existence of outliers, collinearity and number of attributes).

Park and Baek [Park, 08] built and evaluated ANN effort estimation models by using regression analysis and expert interviews to select the input variables. The ANN model was compared to expert judgement and two traditional regressions. The authors found ANN to yield the most accurate predictions. They also emphasised how most existing studies focus on selecting the best estimation method without mentioning how variables are being selected, but usually refine the set of factors by a trial and error approach. In such an approach the different sets of factors are then tested repeatedly until the evaluation criteria are met. The authors also added that a method to define which factors to use as inputs in ANN does not exist yet and underlined that

it is critically important to identify dominant factors that should be used in these models.

In [Azzeh, 10] the impact of Grey Relational Analysis (GRA) integrated with Fuzzy set theory for a by-analogy effort estimation model was investigated and also compared to ANN, CBR and MLR models using several public datasets, i.e., ISBSG, Desharnais, COCOMO, Albrecht and Kemerer. The Fuzzy GRA produced statistically more significant results than the rest of the models. Moreover, it effectively reduced the uncertainty of attribute measurement between two software projects and improved the way to handle both numerical and categorical data in similarity measurements.

Summarising the above, the current literature is quite rich in studies reporting the use and comparison of ANN with other techniques which have offered valuable lessons. In several cases the ANN were found to outperform the techniques compared to; in other cases they performed similarly with other techniques, while it was not the most appropriate technique to use for specific datasets, for example if too few data samples were available [Serluca, 95; Hughes, 97; Tronto, 08]. The majority of the studies suggested that ANN present high data and parameters dependence (such as internal layer nodes and weight values) [Mair, 00; MacDonell, 97; Idri, 04; Tronto, 08], factors that were taken into consideration in the experiments conducted in our work. Also, the issues of overfitting [Idri, 02] and selecting the appropriate inputs for the models were raised [Wittig, 97; Park, 08]. Although in many cases ANN were recognised to produce reasonably accurate predictions when complex relationships between inputs and outputs existed, or in the presence of outlying data, it is quite possible that in some cases they may fail to generalise when conditions change or they may not accurately predict every project [Tronto, 08; Samson, 97]. In a parallel context, this work provides an assessment of computational models applied for SCE. The methodology proposed combines ANN and a simpler yet rigorous Input Sensitivity Analysis (ISA) technique than the relative techniques found in [Refenes, 95; Belue, 95; Glorfeld, 96; Olden, 02] with the objective to eliminate the less influential input parameters by computing the sensitivity level of each connection (weight) to the internal structure and address the open issue identified in [Wittig, 97; Chulani, 99; Park, 08].

## 3 ANN Modelling and Estimation Methodology using ISA

### 3.1 Datasets Description

Two datasets were used in this work, the Desharnais [Desharnais, 89] and the ISBSG Release 9 (obtained from http://www.isbsg.org/) [ISBSG, 05] containing historical samples of past software projects. The Desharnais (1989) dataset includes observations for 81 systems developed by Canadian Software Development Houses. The second dataset is provided by the International Software Benchmarking Standards Group (ISBSG) and contains an analysis of the cost and functional size measurements for a large group of software projects, approximately 3,024. The projects come from a broad cross section of industry and range in size, effort, development platform and language. These projects underwent a series of quality checks and pre-processing to create filtered versions of the datasets that do not

contain null values and conform to the standards we set for homogeneity and integrity before feeding them as inputs to the ANN. The list of the Desharnais and ISBSG attributes selected and used in this work, along with their abbreviations are summarised in Table 1. The attributes include numerical and ordinal values and are related to people, schedule, function points and size metrics as we assumed that these attributes are the most valuable descriptors of development effort. The term ordinal attribute is used for any variable with values ordered in categories (such as low, medium, high) represented by incremental integers so that the lowest values correspond to the first category and the highest to the last.

| Dataset | Attributes | Abbreviation |
|---|---|---|
| **Desharnais** | Team Experience (years) | TE |
| | Manager Experience (years) | ME |
| | Duration (months) | DU |
| | Transactions | TR |
| | Entities | EN |
| | Points Adjusted | PA |
| | Scope | SC |
| | Points Non Adjusted | PNA |
| **ISBSG** | Functional Size | FS |
| | Adjusted Function Points | AFP |
| | Project Elapsed time | PET |
| | Project Inactive time | PIT |
| | Resource Level (ordinal) | RL |
| | Maximum Team Size | MTS |
| | Input count | INC |
| | Output count | OC |
| | Enquiry count | EC |
| | File count | FC |
| | Interface count | IFC |
| | Added count | AC |
| | Changed count | CC |
| | Deleted count | DC |

*Table 1: Dataset Attributes and Abbreviations*

According to [Desharnais, 89] TR in the Desharnais dataset is defined by the number of inputs, outputs and enquiries, that is, the logical transactions in the system, while EN is defined as the number of entities in the system's data model. PA is calculated adding the number of TR and EN (as specified by Albrecht's approach) according to their identification as external or internal and with respect to a complexity level [Albrecht, 79; Albrecht, 83]. This calculation also takes into account various technical and quality characteristics called the General System Characteristics. SC represents the function point complexity adjustment factor (the total processing complexity), whereas PNA represents the Function Points (FP) adjusted by an adjustment factor and is equal to 0.65+(0.01*PA). According to the

aforementioned definitions variables PA and PNA seem to have erroneously switched labels from the original source [Desharnais, 89] an observation also reported in [Port, 08]. The output variable is the actual development effort measured in person-hours. In the Desharnais dataset even though DU may be considered as a dependent variable we consider it among the independent variables because usually the project schedule (or duration) is known (planned) at the initiation of a project and is considered highly correlated to effort.

For the ISBSG dataset, FS is equal to the unadjusted form of the FP count. The AFP stands for the adjusted form of the FS of the project at the final count using a Value Adjustment Factor (VAF), if such a factor was used in the measurements. The VAF is provided by the developer and takes into account various technical and quality characteristics, like data communications and user efficiency. AFP is dependent on the counting approach used. PET and PIT correspond to the project schedule and are measured in calendar months. RL involves data about the level of people whose time is included in the work effort data recorded and MTS is the maximum number of people that worked at any time on the project (peak team size). The rest of the variables reported in Table 1 for the ISBSG dataset comprise the basic constituents of the Function Points measurements. Therefore, one may consider the inclusion of these elements as a repetition of the same size-related information. Nevertheless, this is not actually the case as each attribute is rather unique, i.e. it carries its own part of size description which is not repeated in the others. The output attribute is the newly formed Full-Cycle Work Effort containing only the projects that report actual work effort for the full development life-cycle thus avoiding possible bias inserted by the normalisation of the summary work effort. This is because the initial variables included in the ISBSG dataset, namely Normalized Work Effort and Summary Work Effort, represent an estimate of the full development life-cycle effort for projects covering less than a full development life-cycle and the actual effort reported respectively. It should be mentioned here that both of the forms of the Function Points metric are utilised in each dataset during the experiments firstly to exploit the benefits of their raw (unadjusted) form (PA and FS respectively) and secondly to investigate whether the transformations made from the unadjusted to the adjusted form (PNA and AFP) added any subjectivity or bias. The rationale of the latter lies with the possibility that this bias may become evident when assessing the significance of the input variables: One expects that if no bias is inserted in the adjusted form of a variable then its significance and explanatory power over the dependent variable (i.e. the effort) will be the same as that of its unadjusted. Prior studies on the aforementioned relationships among the components of Function Point Analysis may be found in [Jeffery, 96; Lokan, 99].

### 3.2 The Methodology

The methodology of this paper (presented in Figure 1) employs Computational Intelligent models (Artificial Neural Networks (ANN)) and examines their forecasting ability for the estimation of development effort (or Software Cost Estimation (SCE)). The main issues examined are the following: (i) Can we isolate a set of ANN which are well-trained in terms of accurately estimating software development effort (low prediction error and consistent performance)? (ii) Can we identify a set of attributes that influence software effort more than the rest through Input Sensitivity Analysis

(ISA)? (iii) Can we reduce the size of input dimension of the ANN and not compromise the accuracy of the results?

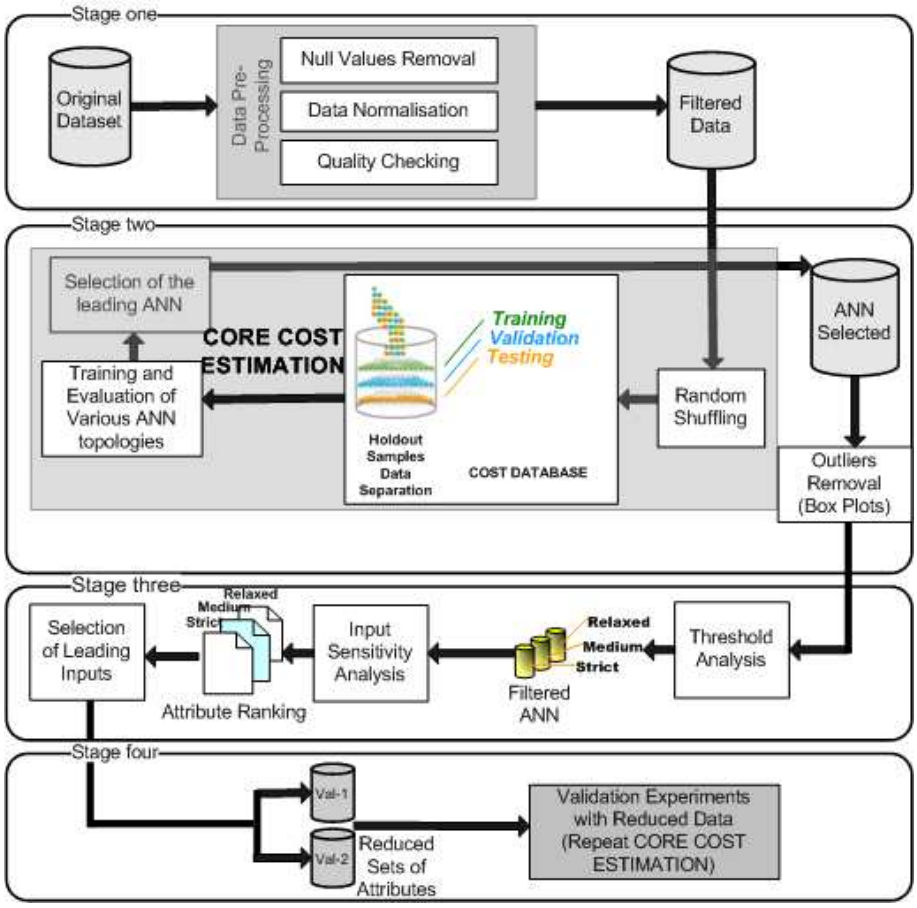


*Figure 1: The stages of the methodology*

In particular, the following stages were carried out:

***Stage one:*** The following pre-processing tasks were performed: In both datasets categorical attributes were removed, whereas ordinal attributes were kept. Then, irrelevant attributes with software size, complexity, productivity, or derived attributes (obtained from transformations of other attributes) were excluded. After that, the ISBSG dataset was processed based on some guidelines provided by the ISBSG. The projects kept were only those measured under the same counting approach (i.e., IFPUG) and with data quality and Unadjusted Function Points ratings equal to 'A' or 'B'. Variables with more than 40% null values were removed (as keeping them in the dataset would cause a substantial reduction in the sample size after deleting rows containing null values among the samples of those variables). In addition, only projects with the same value in Summary Work Effort and Normalized Work Effort

were kept as they represent projects with values of effort for the whole software life-cycle. Finally, the data were normalised in the range [-1, 1] so that the attributes would have the same effect.

***Stage two:*** The stage includes the Core Cost Estimation (CCE) module which was iterated 250 times and each time the input samples were shuffled. Thus, no specific order was given to the projects. The prediction ability of ANN was assessed using holdout samples, meaning that different parts of the input data were used for training, validation and testing. The different parts were again random (with 70% of the data samples being used for training, 10% for validation (i.e. taking into account the errors of this set to adjust the weights) and 20% for testing ('unseen' during training)). This repetitive approach led to investigating the performance and robustness of ANN using various parts of the datasets. The results reported concern the best performing and the average performance of the ANN constructed.

In addition, different ANN architectures were utilised which were analogous to the number of inputs. The ANN architectures variations were produced by modifying the number of neurons in the internal hidden layers empirically, starting from the number of inputs for each dataset and increasing by 1 in each step until it reached to twice the number of inputs, so that overfitting due to too many neurons would not occur.

The hyperbolic tangent sigmoid transfer function was used in the input and hidden layers and the pure linear function was used in the output layer. The ANN were trained with the gradient descent back-propagation algorithm. The number of training epochs was set to 100 and the training function updated the weight values according to the scaled conjugate gradient method.

The output of each ANN was the development effort. The process of training continued until no improvement on the learning ability of the network was observed, as measured by the error figures on the validation data. The performance function used was the *Mean Squared Error* (*MSE*), the learning rate and mutation constant were set to 0.3 and 0.6, whereas the Marquardt adjustment parameter, Marquardt decrease and Marquardt increase factors were set to 1, 0.8 and 1.5 respectively. Next, the generalisation ability of the trained network was assessed by testing its forecasting performance on the set of totally new to the network data samples (testing set). The 10% of the best performing ANN were selected for further experimentation and analysis by assessing the several metrics between the estimated and actual effort values in the testing set (as explained in a subsequent section).

In the final step of this stage, Box Plots were produced to examine the overall performance of the population of ANN using the metric *MMRE*, which is a scale independent metric, and also remove any extreme networks. Thus, we are mostly interested in assessing the overall accuracy quality of ANN.

***Stage three:*** The identification of an order of significance for the input attributes in ANN was examined in this stage. More specifically, the scale describing the degree of influence of each attribute on the predicted effort was investigated. This task was performed using notions of Input Sensitivity Analysis (ISA), described extensively in [Refenes, 95] and [Azoff, 94]. Even though a number of measures have been proposed to determine the significance of ANN input attributes [Garson, 91; Belue, 95; Glorfeld, 95; Satizábal, 07] the method adopted is simple to follow and able to effectively reflect the impact of each input variable to the output [Belue, 95].

According to ISA, one can sum up the absolute values of the weights fanning from each input attribute to all nodes in the successive hidden layer, thus estimating the overall connection strength of this attribute. Unlike other techniques (e.g., Garson's algorithm which make use of the entire hidden structure for calculating the effect of a certain input on the output) the adopted ISA takes into consideration only the first level of neurons without loss of generality, as demonstrated by [Azoff, 94]. The final outcome of this task was a selection of attributes which was assessed in contributing to the effort forecasting process. More specifically, the selected attributes formed two new validation datasets (Val-1/2) - used in the validation experiments of Stage four.

***Stage four:*** The two final subsets of project attributes were used to examine the accuracy performance of the ANN using the same projects and repeating the CCE module of Stage two.

### 3.3 Evaluation Metrics

This section describes the performance evaluation metrics utilised [Conte, 86] for the experimental process. A combination of three common metrics in the SCE literature was used, namely the *Mean Magnitude of Relative Error* (*MMRE*), the *Correlation Coefficient* (*CC*) and the *Normalized Root Mean Squared Error* (*NRMSE*). These error metrics were employed to validate the model's estimation ability considering the difference between the actual and the predicted cost samples and their ascendant or descendant progression in relation to the actual values.

The *MMRE*, given in equation (1), shows the prediction error for the sample being predicted. $x_{act}(i)$ is the actual and $x_{pred}(i)$ the predicted effort value of the $i^{th}$ project.

$$MMRE(n) = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{x_{act}(i) - x_{pred}(i)}{x_{act}(i)}\right| \tag{1}$$

The *CC* (Pearson's correlation measure) between the actual and predicted values, described by equation (2), indicates whether the actual and the predicted samples move in the same direction. An absolute *CC* value equal or near 1 is interpreted as a perfect follow up of the original values by the forecasted one. A negative *CC* sign indicates that the forecasting values follow the same direction of the original with negative mirroring, that is, with a $180°$ rotation about the time-axis. A *CC* value close to zero signifies poor performance on behalf of predictions in capturing the evolution of the original values.

$$CC(n) = \frac{\sum_{i=1}^{n}[(x_{act}(i) - \bar{x}_{act,n})(x_{pred}(i) - \bar{x}_{pred,n})]}{\sqrt{\left[\sum_{i=1}^{n}(x_{act}(i) - \bar{x}_{act,n})^2\right]\left[\sum_{i=1}^{n}(x_{pred}(i) - \bar{x}_{pred,n})^2\right]}} \tag{2}$$

The *NRMSE* assesses the quality of predictions and is calculated using the *Root Mean Squared Error* (*RMSE*) as follows:

$$NRMSE(n) = \frac{RMSE(n)}{\sigma_{\Delta}} = \frac{RMSE(n)}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_{act}(i) - \bar{x}_n)^2}} \tag{3}$$

$$RMSE(n) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}[x_{pred}(i) - x_{act}(i)]^2} \quad (4)$$

If *NRMSE*=0 then predictions are perfect; if *NRMSE*=1 the prediction is no better than taking $x_{pred}$ equal to the mean value of *n* samples.

In addition, the evaluation metric *Prediction of specific Level l* (*Pred(l)*) was used to evaluate the ANN performance, which is specified in equation (5). Essentially, equation (5) defines the ratio of the accurate data predictions *k* to the total number of data points predicted *n.* This accuracy is measured by the *RE* metric given in equation (6) which must be lower than level *l*. In our experiments parameter *l* was set equal to 0.25. Finally, the *RE* metric is also used in conducting the statistical comparative tests of section 4.

$$pred(l) = \frac{k}{n} \quad (5)$$

$$RE(i) = \left| \frac{x_{act}(i) - x_{pred}(i)}{x_{act}(i)} \right|, \ i = 1...n \quad (6)$$

## 4 Experimental Results

The experiments were conducted on the pre-processed datasets containing 77 projects described by 8 attributes for the Desharnais and 113 projects consisting of 14 attributes for the ISBSG dataset. We will refer to these new datasets as the "filtered data" since immense data reduction was performed. The filtered data include predictor variables and one response variable (effort), with the latter ranging from 546 to 23,940 person-hours in the Desharnais dataset and from 140 to 36,046 person-hours in the ISBSG dataset. Tables 2 and 3 summarise the descriptive statistics of the Desharnais and ISBSG datasets respectively.

The attributes indicate small differences in the central tendency and large differences in the deviations indicating that their values are spread. In the Desharnais attributes there are large variations in the distributions of DU, TR, EN, PA, PNA and Effort, they are highly skewed and not normally distributed (the Shapiro-Wilk test confirmed the non-normality since the significance value is less than 0.05). Also, in the ISBSG case most of the attributes are not normally distributed based on the kurtosis, skewness and significance values reported performing the Shapiro-Wilk test assessing normality. The two datasets, although align in terms of their general statistical form (spread, distribution), they differ in terms of values which stems from the fact that the Desharnais projects come from a single country in contrast to the cross- cultural/national origin of the ISBSG projects. This means that a logarithmic transformation of the data samples should be performed before attempting to fit a linear model (like in regression approaches).

| Attribute | Mean | Median | Standard Deviation | Sample Variance | Kurtosis | Skewness | Min | Max |
|---|---|---|---|---|---|---|---|---|
| **TE** | 2.30 | 2 | 1.33 | 1.76 | -1.28 | -0.05 | 0 | 4 |
| **ME** | 2.65 | 3 | 1.52 | 2.31 | 0.09 | 0.23 | 0 | 7 |
| **DU** | 11.30 | 10 | 6.79 | 46.05 | 2.74 | 1.45 | 1 | 36 |
| **TR** | 177.47 | 134 | 146.08 | 21339.57 | 7.66 | 2.39 | 9 | 886 |
| **EN** | 120.55 | 96 | 86.11 | 7414.62 | 1.54 | 1.39 | 7 | 387 |
| **PA** | 298.01 | 258 | 182.26 | 33219.86 | 5.07 | 1.84 | 73 | 1127 |
| **SC** | 27.45 | 28 | 10.53 | 110.88 | -0.37 | -0.19 | 5 | 52 |
| **PNA** | 282.39 | 247 | 186.36 | 34730.00 | 4.44 | 1.74 | 62 | 1116 |
| **Effort** | 4833.91 | 3542 | 4188.19 | 17540894.50 | 5.30 | 2.04 | 546 | 23940 |

*Table 2: Descriptive Statistics for the Desharnais dataset*

| Attribute | Mean | Median | Standard Deviation | Sample Variance | Kurtosis | Skewness | Min | Max |
|---|---|---|---|---|---|---|---|---|
| **FS** | 436.27 | 252 | 522.80 | 273320.38 | 7.72 | 2.59 | 42 | 3155 |
| **AFP** | 455.27 | 264 | 558.85 | 312316.79 | 8.45 | 2.65 | 39 | 3471 |
| **PET** | 9.50 | 8 | 8.89 | 79.03 | 44.01 | 5.53 | 1 | 84 |
| **PIT** | 1.34 | 0 | 4.31 | 18.55 | 72.25 | 7.86 | 0 | 42 |
| **MTS** | 6.05 | 4 | 7.75 | 60.06 | 29.39 | 4.32 | 0 | 65 |
| **INC** | 141.70 | 66 | 219.33 | 48106.82 | 9.54 | 2.90 | 0 | 1327 |
| **OC** | 87.73 | 57 | 105.03 | 11031.36 | 9.20 | 2.73 | 0 | 620 |
| **EC** | 63.13 | 30 | 88.70 | 7867.35 | 11.10 | 2.96 | 0 | 534 |
| **FC** | 111.08 | 56 | 164.46 | 27047.18 | 11.42 | 3.11 | 0 | 995 |
| **IFC** | 32.62 | 10 | 61.60 | 3794.36 | 10.71 | 3.19 | 0 | 329 |
| **AC** | 395.28 | 213 | 523.12 | 273654.37 | 8.18 | 2.63 | 0 | 3155 |
| **CC** | 39.41 | 0 | 109.80 | 12056.42 | 26.96 | 4.54 | 0 | 844 |
| **DC** | 1.58 | 0 | 12.41 | 153.89 | 98.80 | 9.74 | 0 | 128 |
| **Effort** | 4674.91 | 1974 | 6659.42 | 44347932.67 | 8.43 | 2.75 | 140 | 36046 |

*Table 3: Descriptive Statistics for the ISBSG dataset*

The next three sub-sections present the results obtained through the experiments executed in Matlab R2009a, following the stages of the methodology as previously explained. The main results of the proposed approach include: (i) the prediction performance of the selected ANN from the Core Cost Estimation (CCE) component (also free of outliers); (ii) the attribute rankings according to ISA and threshold analyses, and (iii) the validation results using the reduced sets of attributes (Val-1/2). The last two sub-sections present an analysis with regression comparing a set of models created with the original data values and the obtained subsets, and a brief discussion of some threats to the validity of the results.

### 4.1 Artificial Neural Network Results and Box Plots

Table 4 lists the top 5 ANN obtained in terms of prediction accuracy (*MMRE*) using the transformed values for each dataset which are yielded during the testing phase (prediction). The first column of Table 4 specifies the dataset and the second column reports the corresponding ANN architecture; where in all respective Tables 4, 7, 8 and 10 the column reporting a topology "*x-y-1*$_z$" refers to an ANN architecture with *x* nodes in the input layer, *y* nodes in the hidden layer and *1* output node. The *z* subscript indexing scheme is used to differentiate experiments performed in the respective experiment repetition of the CCE component for the ANN topologies examined but for different training and testing sets. The rest of the table columns report the testing phase errors and the correlation metric (*CC*). The last row of the table presents the obtained median values for the total set of experiments performed with each dataset.

| Dataset | ANN Topology | MMRE | CC | NRMSE | Pred(l) |
|---|---|---|---|---|---|
| **Desharnais** | 8-13-$1_{74}$ | 0.105 | 0.987 | 0.158 | 0.933 |
| | 8-15-$1_{199}$ | 0.085 | 0.992 | 0.132 | 1.000 |
| | 8-15-$1_{87}$ | 0.062 | 0.995 | 0.107 | 1.000 |
| | 8-17-$1_{58}$ | 0.111 | 0.989 | 0.145 | 1.000 |
| | 8-18-$1_{190}$ | 0.111 | 0.986 | 0.196 | 1.000 |
| | **Median** | **0.214** | **0.931** | **0.389** | **0.933** |
| **ISBSG** | 14-16-$1_{22}$ | 0.052 | 0.989 | 0.152 | 1.000 |
| | 14-19-$1_{172}$ | 0.060 | 0.990 | 0.137 | 1.000 |
| | 14-22-$1_{218}$ | 0.087 | 0.981 | 0.196 | 0.955 |
| | 14-26-$1_{145}$ | 0.059 | 0.986 | 0.166 | 1.000 |
| | 14-30-$1_{58}$ | 0.067 | 0.865 | 0.509 | 1.000 |
| | **Median** | **0.150** | **0.955** | **0.303** | **1.000** |

*Table 4: Indicative Effort Prediction Assessment of the best performed ANN*

Overall the results obtained show robustness in terms of performance and positive correlation between the actual and predicted values. A more detailed analysis of the results per dataset follows:

***Desharnais case:*** The error figures obtained among the best ANN were very close with respect to the *MMRE*, *NRMSE* and *Pred(0.25)* while the *CC* indicates a positive relationship between the actual and predicted values. In particular, the

optimal testing figures obtained are: *MMRE*=0.062, *CC*=0.995, *NRMSE*=0.107 and *Pred(0.25)*=1 (with a 15 hidden neurons' topology).

***ISBSG case:*** The low *MMRE* and *NRMSE* values obtained show that the models are able both to learn and generalise the knowledge even though the dataset contains heterogeneous projects from various industries and countries. In particular, the optimal testing figures obtained are: *MMRE*=0.052, *CC*=0.989, *NRMSE*=0.152 and *Pred(0.25)*=1 (with a 16 hidden neurons' topology).

Since ANN's performance usually depends on a set of parameters (and to alleviate the possibility of utilising networks with variable performance spread), in our approach, we have used Box Plots on the testing errors to remove any outlying ANN - yielding extreme and mild accuracy values.

Figure 2 depicts the Box Plots of the accuracy of the ANN built based on the *MMRE* figures in the Desharnais and the ISBSG cases. These outlying networks (marked as crosses in Figure 2) were excluded using Box Plots with a maximum length of each whisker set to 1.5 times the inter-quartile range. Particularly, 16 and 6 networks were considered outliers in the Desharnais and ISBSG cases respectively, leaving 234 and 244 ANN for the analysis that follows. We decided to use only the *MMRE* obtained during testing to filter-out potential outliers, as it is the most popular metric in the SCE literature and it is scale independent. One may observe that the spread of the *MMRE* is small in both cases and thus on average the performance of the ANN is consistent. Also, the small number of outlying ANN found during this process, which concludes Stage two, shows that the approach produces consistent results.

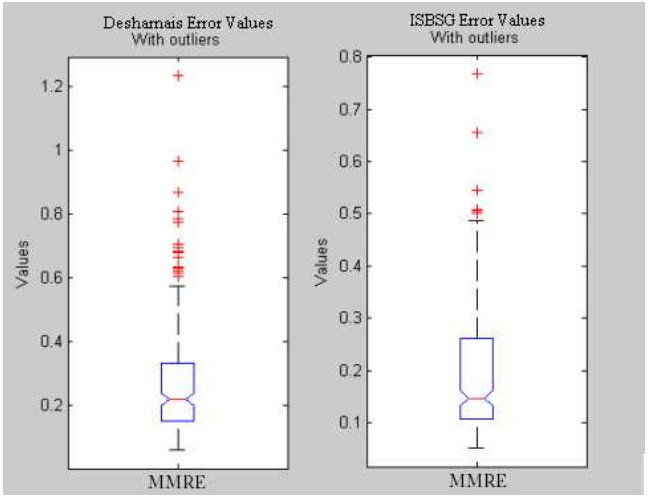


*Figure 2: Box Plots of Outliers from the Desharnais and ISBSG datasets*

### 4.2 Input Sensitivity Analysis Results

Stage three of the experimental procedure aimed to identify the leading factors affecting software effort using Threshold Analysis and Input Sensitivity Analysis

(ISA) on the remaining set of ANN created for each dataset. First, we used an empirical Threshold Analysis based on three filtering levels to separate various subsets of the ANN based on their performance, i.e., the top performing, the lower-than-top performing and the medium performing ANN. To perform this separation, three empirical thresholds were decided having the target to isolate and study the top 15%, 20% and 25% of the total ANN produced in terms of prediction accuracy, i.e., with respect to the metrics *MMRE*, *CC* and *NRMSE* and as specified in Table 5.

| | **Strict** | **Medium** | **Relaxed** |
|---|---|---|---|
| ***MMRE*** | $\leq 0.15$ | $\leq 0.20$ | $\leq 0.25$ |
| ***CC*** | $\geq 0.85$ | $\geq 0.80$ | $\geq 0.75$ |
| ***NRMSE*** | $\leq 0.15$ | $\leq 0.20$ | $\leq 0.25$ |

*Table 5: Values for Evaluating ANN in Threshold Analysis*

The above metrics have been selected, although previously reported studies [Kitchenham, 01] have criticised the problematic accuracy of measures such as the *MRE* to select optimal models, because they have been widely used in the SCE literature. In addition, their values were defined from suggestions in the relevant literature, for example [Conte, 86] consider *MMRE* and *RE* lower or equal to 0.25 as an acceptable level of performance for effort prediction models, and *CC* values greater than or equal to 0.75 exhibits a linear positive relationship of the predicted and the actual effort values.

The Threshold Analysis led to the selection of the following ANN in each case:

***For the Desharnais dataset***: the *Strict* filtering level retained a total of 5 ANN, the *Medium* level 14 ANN and finally the *Relaxed* level 35 ANN.

***For the ISBSG dataset***: the *Strict* filtering level kept 5 ANN, the *Medium* level 28 ANN and lastly, the *Relaxed* level 69 ANN.

Each of these ANN subsets were used for ISA. ISA aimed to derive the significance rank of the ANN inputs for each level as listed in Table 6. The analysis explained below aimed firstly to assess the inputs' importance for describing effort and secondly derive two validation sets that are characterised with over than 50% reduction in the number of attributes. The validation sets will then be examined in terms of accuracy performance using ANN in the next stage. The following process was executed to achieve the abovementioned targets:

i. The leading *trunc[n/2]* inputs according to their weights' values were isolated for each filtering level, where *n=8* for the Desharnais, *n=14* for the ISBSG and *trunc[]* denoting the integer part of the quotient.
ii. The attributes that were ranked the leading ones by all filtering levels were placed in the so-called first evaluation set (Val-1).
iii. If the number of elements in the evaluation set was equal to *trunc[n/2]*, which essentially means that each filtering level indicated the same leading inputs, the process was terminated. Otherwise, the process continued with step (iv).
iv. The rest of the attributes that were promoted by some of the filtering levels were further examined to create the second evaluation set (Val-2) as follows:

a. One by one the attributes promoted by the *Strict* level were examined first, followed by those of the *Medium* and finally by those suggested by the *Relaxed* level. For each attribute the following two steps were executed.
   a.1. If the attribute was suggested also by at least one of the other two filtering levels then it was placed in the second evaluation set, together with the attributes of Val-1, forming Val-2.
   a.2. If the elements in the evaluation set reached *trunc[n/2]* the process was terminated. Otherwise, it continued with the rest of the attributes of step (a) above.

Table 6 lists the attribute rankings of each filtering level for the datasets used. The attributes in bold include the leading (ranked first) attributes of each dataset and the attributes that yielded absolute sum of weights within the 80% of the weight value of the leading attribute. The results summarised in Table 6 are analysed for each case respectively:

***Desharnais dataset:*** the leading attributes were consistent among the various filtering levels. We observe that all three filtering levels suggested relatively similar inputs as leading determinants among the first four, namely the DU and SC attributes, which are thus included in the first validation set (Desharnais-Val-1). From the rest of the attributes promoted by the thresholds namely TR, TE, PA, PNA and EN, only TR appears in at least two other filtering levels (*Strict* and *Medium*). The selection of the TR attribute leads to the creation of a second validation set, namely Desharnais-Val-2, comprising of the DU, SC and TR attributes.

***ISBSG dataset:*** All filtering levels consistently proposed FS, AC and OC as the most significant attributes among the first seven. Thus, these attributes formed the first validation set called ISBSG-Val-1. Taking into consideration the rest of the suggested attributes from the filtering levels, i.e., INC, CC, MTS, AFP, DC, EC, IFC and RL, the above process distinguished CC, DC, EC and IFC, which, together with the attributes in ISBSG-Val-1, form ISBSG-Val-2.

The attributes promoted in the Desharnais case relate to the schedule and scope of the project and more specifically to the calendar months occupied by the project and the overall function points' complexity factor. In the ISBSG case all attributes promoted relate to project sizing: Apart from the functional size, the next most significant attributes relate to the number of changes, additions and deletions performed after requirements specification, as well as counts of external outputs, external enquiries and external interface files used. The appearance of these variables in the first order of significance indicates that the basic components of the FP metric and the number of changes (i.e., updates, deletions, additions) to the initial specifications have a decisive effect on the effort variable. Thus, making changes after specifying requirements seems to add a considerable burden to the overall effort accounted during the development process. This finding is reasonable for traditional software development processes where the cost of changing specifications at the design and implementation phases increases as development proceeds to later phases. It is important to note here that the attributes promoted in the first order of significance and later used in the validation experiments may be measured at the early project phases, in the sense that a rough estimation of their magnitude may become

available at the beginning of the development process. In the same context, one may argue that project duration is a dependent variable; in our case, though, we consider it as an independent variable as the time plan is already prepared (or a close approximation of the required time-span) prior to effort estimation.

| Dataset | Filtering Level | Leading Inputs (starting from left) |
|---|---|---|
| **Desharnais** | *Strict* | $\mathbf{W_{DU}} > W_{SC} > W_{TR} > W_{TE} > W_{PA} > W_{ME} > W_{PNA} > W_{EN}$ |
| | *Medium* | $\mathbf{W_{SC}} > \mathbf{W_{DU}} > \mathbf{W_{PA}} > \mathbf{W_{TR}} > W_{EN} > W_{TE} > W_{PNA} > W_{ME}$ |
| | *Relaxed* | $\mathbf{W_{SC}} > \mathbf{W_{PNA}} > \mathbf{W_{DU}} > \mathbf{W_{EN}} > W_{TR} > W_{PA} > W_{ME} > W_{TE}$ |
| **ISBSG** | *Strict* | $\mathbf{W_{FS}} > \mathbf{W_{INC}} > \mathbf{W_{CC}} > \mathbf{W_{AC}} > \mathbf{W_{OC}} > \mathbf{W_{MTS}} > \mathbf{W_{AFP}} > W_{PET} > W_{DC} > W_{EC} > W_{IFC} > W_{RL} > W_{PIT} > W_{FC}$ |
| | *Medium* | $\mathbf{W_{FS}} > \mathbf{W_{DC}} > \mathbf{W_{OC}} > \mathbf{W_{EC}} > \mathbf{W_{AC}} > W_{CC} > W_{IFC} > W_{INC} > W_{MTS} > W_{AFP} > W_{RL} > W_{PIT} > W_{PET} > W_{FC}$ |
| | *Relaxed* | $\mathbf{W_{FS}} > \mathbf{W_{EC}} > \mathbf{W_{IFC}} > \mathbf{W_{OC}} > \mathbf{W_{DC}} > \mathbf{W_{AC}} > \mathbf{W_{RL}} > \mathbf{W_{INC}} > \mathbf{W_{CC}} > \mathbf{W_{PET}} > \mathbf{W_{AFP}} > \mathbf{W_{PIT}} > W_{MTS} > W_{FC}$ |

*Table 6: Leading Determinants According to ISA Weight Values from the ANN*

The appearance of FS as the leading attribute in all three threshold levels of ISBSG indicates that Functional Size influences software effort in a high degree, in comparison to the rest of the attributes examined, which is again a reasonable conclusion. In fact, most models proposed in the relevant SCE literature identify software size as the most important factor and usually base effort estimation on size attributes (i.e., Lines of Code, Function Points, Object Points etc.). Moreover, estimating size attributes from the early phases of the development process is considered an important subject of research in the area of SCE. Considering the ranking position of AFP, which is the transformed form of FS - AFP lies among the last in significance attributes and never to a position close to FS - leads us to infer that possibly AFP suffers from subjectivity. With this we mean that since FS seems to be the dominant attribute describing effort then we would expect its transformed version, expressed by AFP, to be equally strong in determining the evolution of effort. This, though, was not the case, as illuminated by the ANN (it was given lower rank and was considered less important than FS). Therefore, we may assume that transformation of FS to AFP, at least the one performed and recorded in the samples of the ISBSG dataset, 'destroyed' the original pattern of FS which reflected development effort, a fact that may be attributed to a non-universal and biased way of performing this transformation.

### 4.3 Validation Experiments

The final step of the fourth stage involved the execution of the validation experiments for each dataset with the newly formed and reduced in number of attributes validation sets (Val-1 and Val-2). The stage included again training, testing and validation of ANN (repeating the CCE component). The percentage of the best ANN utilised was adjusted from 10% to 25% in the CCE stage due to the small number of experiments executed, mostly as a result of the smaller number of attributes in each validation set. Finally, the results of the best performing ANN in terms of effort estimation for each

dataset using the full attributes sets were compared to those obtained with the reduced evaluation sets (Val-1 and Val-2) using the Mann-Whitney test [Mann, 47].

Table 7 presents a selected sample of the validation results with the transformed values for both datasets and their respective reduced datasets. Figures 3 and 4 depict graphically a partial view of the actual and predicted effort values during testing for the Desharnais and ISBSG datasets.

| Validation Dataset | ANN Topology | MMRE | CC | NRMSE | Pred(l) |
|---|---|---|---|---|---|
| **Desharnais-Val-1** | $2\text{-}6\text{-}1_{58}$ | 0.133 | 0.964 | 0.285 | 0.933 |
| | $2\text{-}5\text{-}1_{146}$ | 0.154 | 0.974 | 0.229 | 1.000 |
| | $2\text{-}5\text{-}1_{23}$ | 0.185 | 0.974 | 0.221 | 1.000 |
| | **Median** | **0.396** | **0.806** | **0.609** | **0.933** |
| **Desharnais-Val-2** | $3\text{-}8\text{-}1_{81}$ | 0.130 | 0.971 | 0.231 | 1.000 |
| | $3\text{-}6\text{-}1_{3}$ | 0.230 | 0.895 | 0.455 | 1.000 |
| | $3\text{-}5\text{-}1_{237}$ | 0.245 | 0.957 | 0.289 | 1.000 |
| | **Median** | **0.366** | **0.832** | **0.566** | **0.933** |
| **ISBSG-Val-1** | $3\text{-}8\text{-}1_{226}$ | 0.131 | 0.973 | 0.237 | 1.000 |
| | $3\text{-}7\text{-}1_{16}$ | 0.132 | 0.953 | 0.362 | 0.955 |
| | $3\text{-}8\text{-}1_{27}$ | 0.138 | 0.965 | 0.347 | 0.955 |
| | **Median** | **0.344** | **0.657** | **0.773** | **0.955** |
| **ISBSG-Val-2** | $7\text{-}10\text{-}1_{109}$ | 0.100 | 0.961 | 0.283 | 1.000 |
| | $7\text{-}15\text{-}1_{150}$ | 0.105 | 0.967 | 0.274 | 1.000 |
| | $7\text{-}15\text{-}1_{57}$ | 0.116 | 0.969 | 0.304 | 0.955 |
| | **Median** | **0.281** | **0.776** | **0.657** | **0.955** |

*Table 7: Indicative Validation Experiments of the best performed ANN with reduced attributes (inputs) of the Desharnais and ISBSG datasets*

Based on the performance of the ANN shown in Table 7 assessed using the same metrics as before, it seems that adequate effort estimation accuracy was achieved. More specifically, in terms of accuracy, relatively low error values similar to those obtained using the whole attribute sets were achieved for both datasets. This becomes obvious by comparing the values yielded by the previous experiments with those of the final set of experiments. We observe that when using the Desharnais dataset the ANN with the selected inputs present a negligible increase in the *MMRE* and *NRMSE* figures compared to the original experiments during the testing phase, while both *CC* and *Pred(0.25)* remain at similar levels. The median values obtained from the total of 250 experiments executed with the reduced validation datasets indicate a reduction in the prediction accuracy in *MMRE*, *NRMSE*, *CC*, while *Pred(0.25)* remains at the same levels. With the immense decrease of the number of attributes used in the validation experiments in relation to the original experiments some decrease in the descriptive power of the ANN models was expected. Nevertheless, the reduction in performance accuracy may be considered as minor in relation to the significant reduction of the ANN input space dimension achieved for the specific datasets. Additionally, what is more important here is that these validation results suggest that if the so-called 'significant' or 'influencial' attributes are identified and isolated in a dataset, and then

used for predicting effort, the accuracy of the estimations obtained is comparable to that of the original ANN experiments.

The graphical representation of prediction accuracy of the values transformed in the original scale presented in Figures 3 and 4 provide verification of the successful performance exhibited by the ANN. The predicted samples are very close (with very few exceptions in the ISBSG case), to the actual values in the testing phase of the forecasting process. The results of the Mann-Whitney test listed in Table 8 show that statistical difference exists among the distribution of the *RE* error values obtained with the full attributes and the Val-1 and Val-2 subsets in the Desharnais case and between the full attributes and the Val-1 subset in the ISBSG case. The indication of mean rank 'Dataset A' > 'Dataset B' shows the outperforming results of Dataset A over Dataset B for all the set of predictions obtained.

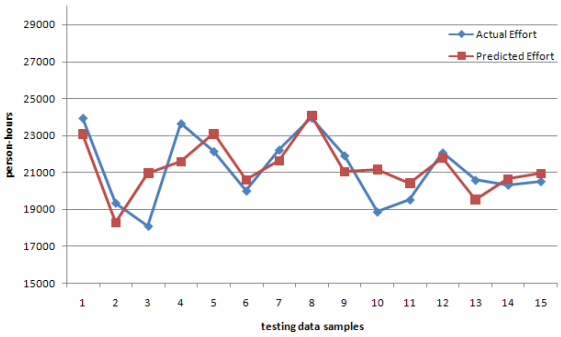


*Figure 3: The partial samples of Actual vs. Predicted Effort samples during Validation Experiments for Desharnais dataset using a 3-8-1 ANN topology*

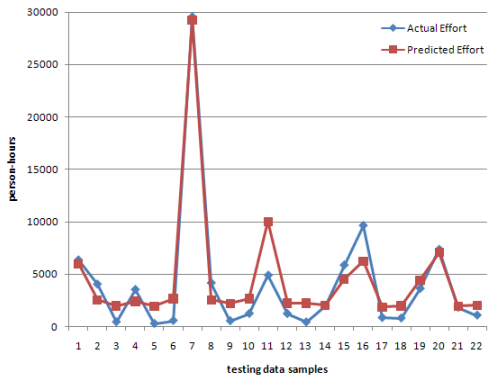


*Figure 4: The partial samples of Actual vs. Predicted Effort samples during Validation Experiments for ISBSG dataset using a 7-10-1 ANN topology*

| Dataset | ANN Topology | Mean Rank | U | z | p |
|---|---|---|---|---|---|
| **Desharnais** | Original: 8-15-$1_{87}$<br>Val-1: 2-5-$1_{23}$ | Original>Val-1 | 51 | -2.551 | 0.011 |
| | Original: 8-15-$1_{87}$<br>Val-2: 3-8-$1_{81}$ | Original>Val-2 | 56 | -2.344 | 0.019 |
| | Val-1: 2-5-123<br>Val-2: 3-8-$1_{81}$ | Val-1<Val-2 | 87 | -1.058 | 0.290 |
| **ISBSG** | Original: 14-26-$1_{145}$<br>Val-1: 3-8-$1_{226}$ | Original>Val-1 | 129 | -2.652 | 0.008 |
| | Original: 14-26-$1_{145}$<br>Val-2: 7-15-$1_{150}$ | Original>Val-2 | 196 | -1.080 | 0.280 |
| | Val-1: 3-8-$1_{226}$<br>Val-2: 7-15-$1_{150}$ | Val-1<Val-2 | 215 | -0.634 | 0.526 |

*Table 8: Mann-Whitney signed-rank test results for the ANN experiments*

The final set of experiments show that the reduction of attributes proposed for the Desharnais case is not significant; whereas what underlines the significance of the results obtained in the ISBSG case is that using the subsets of the selected attributes (Val-2 subset) the prediction accuracy is not compromised. Thus, for the ISBSG projects comparable accuracy to that of the original ANN models utilised for SCE is achieved using the proposed and reduced attributes of Val-2. In addition, some of these project attributes required for SCE in the ISBSG case may be considered easier and less expensive to collect than others, while in the meantime, having to collect fewer measurements contributes to bounding the complexity of the estimation process. Practically, the proposed methodology shows an automatic way to minimise the number of independent attributes of ANN, emphasising on scheduling and sizing attributes for predicting effort. Thus, we may also assume that we actually need to gather and validate a relatively smaller set of project data attributes to perform ANN estimations since these attributes make an appreciable contribution to the ANN result, which is an improvement to the classical form of ANN experiments conducted so far. Therefore, we have additionally suggested a way for avoiding the time and money consuming process of managing and maintaining large portions of data and eliminating the need for measuring too many metrics, while at the same time, managed to preserve highly accurate estimates consistently throughout the experiments performed.

### 4.4 Comparative Experiments with Regression Models

Assessing and comparing various SCE approaches in terms of statistical accuracy throughout a set of experiments is a popular research topic especially in data-driven methods [Finnie, 97; Briand, 00; Heiat, 02; Jørgensen, 07]. In this section, we report the best results of regression models, after trying out multiple data splits and experimenting with many parameters and calibrations of stepwise regression. The results are reported based on the full sets of attributes in the Desharnais and ISBSG datasets, as well as, using the reduced validation subsets (Val-1 and Val-2).

Main aim of this section is to compare the results obtained from ANN with those of a linear regression approach. We carried out the Wilcoxon ranked test [Wilcoxon, 45] to assess whether the results obtained with the best models of each technique are significantly different. Furthermore, we investigated whether adequate prediction accuracy may be achieved with the reduced sets of attributes (as proposed by ISA) in the case where they are utilised by another modelling techniques apart from ANN (using the Mann-Whitney test [Mann, 47]). Thus, the attribute selection process of ISA is assessed in terms of generalisation and universality.

In the experiments conducted we have used Multiple Linear Regression (MLR) which produces a simple cost model but it required some degree of calibration. The level of accuracy obtained using this model was assessed by using the same data and employing the same performance measures as with ANN. The set of cost attributes were investigated on the dependent attribute (i.e., effort) to see how well the regression hyperplane approximates the actual data points, assuming that there is a linear relationship between the dependant variable and a set of independent variables.

The comparison was carried out after data were separated into training and testing sets (as explained previously) and in the MLR case they were additionally transformed to their natural logarithms as their relation was found to be non-linear through residual analysis. After the two techniques were applied in parallel to the same sample data the reverse transformation was used to calculate the final error values. In the MLR case, the coefficients were used to calculate the dependant variable (effort) for the training set and calculate the error figures. The same process was repeated for the testing set of values (i.e. using the testing holdout samples). The latter may be considered the validation process of MLR that tested the appropriateness of the model by predicting the average value of the effort variable in terms of the known values of other variables.

The results of the validation experiments employing the coefficients obtained from the best MLR models in terms of accuracy are summarised in Table 9. The value of development effort is predicted using the complete set of attributes (original) and the same reduced set of attributes proposed by ISA in the ANN experiments (Val-1 and Val-2) for both the Desharnais and the ISBSG data samples.

| Dataset | | MMRE | CC | NRMSE | Pred(l) |
|---|---|---|---|---|---|
| **Desharnais** | Original | 0.496 | 0.889 | 0.765 | 0.294 |
| | Val-1 | 0.463 | 0.715 | 0.846 | 0.235 |
| | Val-2 | 0.463 | 0.823 | 0.689 | 0.412 |
| **ISBSG** | Original | 1.032 | 0.715 | 0.725 | 0.391 |
| | Val-1 | 0.668 | 0.539 | 1.025 | 0.130 |
| | Val-2 | 1.071 | 0.151 | 1.104 | 0.217 |

*Table 9: Performance Results of the best trained MLR Models for SCE*

The experimentation process with the MLR method was executed 250 times with different parts of the dataset used for training and testing (multiple sampling) to optimise the results of Table 9 and identify the subsets of attributes that offered better fitting. Both stepwise selection methods were used (forward and backward) which controlled the inclusion or exclusion of attributes starting with all candidate attributes and testing them one by one for statistical significance, removing those attributes that

were not significant. Several models were thus developed that utilised a variety of attributes, but due to space limitations on one hand, and the fact that the accuracy levels obtained were similar to the results presented in Table 9 on the other, the partial regression results will not be presented in full.

In addition, the backward or forward stepwise selection process of the MLR method on the original (filtered) datasets may be considered as a promising method to promote the most significant predictors (or variables). As specified earlier, though, this is not the target of the relevant experiments carried out in this paper and thus we will not investigate further the order of significance suggested by each regression model tested, leaving this investigation for the future.

The Wilcoxon ranked test (Table 10) performed to evaluate the difference between the medians of the performance of the two techniques (ANN and MLR) revealed that the two techniques did not differ significantly ($p$>0.05) except in the case of using the whole Desharnais dataset and the Val-2 ISBSG dataset where MLR outperformed ANN.

| Dataset | ANN Topology | ANN MMRE | Mean Rank | p |
|---|---|---|---|---|
| **Desharnais** | Original: 8-12-$1_{241}$ | 0.753 | ANN < MLR | 0.028 |
| | Val-1: 2-4-$1_{110}$ | 0.602 | ANN = MLR | 0.463 |
| | Val-2: 3-3-$1_{122}$ | 0.429 | ANN > MLR | 0.287 |
| **ISBSG** | Original: 14-18-$1_{80}$ | 1.032 | ANN < MLR | 0.465 |
| | Val-1: 3-5-$1_{54}$ | 1.200 | ANN < MLR | 0.465 |
| | Val-2: 7-14-$1_{179}$ | 2.265 | ANN < MLR | 0.003 |

*Table 10: Wilcoxon signed-rank test results comparing ANN and MLR experiments*

Overall, the results obtained show that MLR outperforms ANN, something which indicates that either the former technique is a superior approximator of the effort values of the samples, or perhaps the ANN requires more effort in tuning. Nevertheless, the observed performance of the two techniques is comparable and in most of the cases not significantly different.

In addition, the results of the Mann-Whitney test (listed in Table 11) show no statistical difference among the distribution of the *RE* error values obtained with the full attributes and the Val-1 and Val-2 subsets. What underlines the significance is that the proposed reduced subsets (Val-1 and Val-2) in the MLR case provide comparable in terms of accuracy results with respect to that of the original sample values in both datasets cases. In addition, the mean rank indicates superiority of the original sets of attributes in comparison to the reduced subsets in some cases, whereas the opposite is observed in some other cases.

Consequently, the attribute reduction proposed by ISA in the Desharnais and the ISBSG cases yields similar predictions (no statistical difference is observed) with those of the original data samples through the simple linear regression model showing that we cannot rule out the potential of ISA in reducing dimensionality in SCE and preserving accuracy levels.

| Dataset | Mean Rank | U | z | p |
|---|---|---|---|---|
| **Desharnais** | Original > Val-1 | 127 | -0.603 | 0.547 |
| | Original < Val-2 | 130 | -0.499 | 0.617 |
| | Val-1 > Val-2 | 129 | -0.534 | 0.593 |
| **ISBSG** | Original = Val-1 | 255 | -0.209 | 0.835 |
| | Original > Val-2 | 221 | -0.956 | 0.339 |
| | Val-1 > Val-2 | 231 | -0.736 | 0.462 |

*Table 11: Mann-Whitney signed-rank test results for the MLR experiments*

### 4.5 Threats to Validity

There are a few limitations of the approach presented in this paper that generate some threats to the validity of the results. We will now discuss briefly these threats:

i. The Desharnais dataset includes a small size of samples, something that on one hand gives birth to doubts as regards proper ANN training versus overfitting of the data and on the other limits the significance of the findings supporting that this Computational Intelligent approach is suitable for SCE. Nevertheless, the former limitation was faced successfully by using holdout samples and testing the generalisation ability of the networks, while the latter was altered through a similar study of the larger in size ISBSG dataset. In addition, overfitting in training is avoided by stopping training if a maximum amount of epochs or time is reached, if performance reached the goal set and if validation performance increased more than 5 times of maximum validation failures since the last time it decreased.

ii. The variables selected and included in the models of this work were those considered more appropriate to describe development effort. This selection was purely empirical and was not based on any scientific evidence apart from relevant studies describing attempts and experiences with other models utilising these variables. The variables selected involved some highly subjective measures, such as team experience, project size and complexity, whose effect on effort may not be thoroughly captured or explained by any model. Our target, though, was not to assess the subjectivity of the measurements but to produce successful effort estimations with the use of a limited set of variables from the specific datasets, something that was finally achieved.

iii. The ANN models developed and trained will not necessarily work sufficiently well when conditions (data samples) change. Having in mind that the proposed models are empirical investigations based on available effort datasets, it is clear that when new data emerge the trained models may fail to generalise. Especially in software development environments that are frequently characterised by rapid change in the technologies used, the people involved and the products built, it is hardly the case that within different conditions the same accuracy results in terms of performance errors will be obtained. In the investigations carried out on the specific datasets the trained ANN seem to have worked sufficiently well using the holdout samples during validation. The same models, though, in light of a large number of

completely different projects may or may not work that well. In such case, a possible solution will be to repeat the process of training until the network restores its ability to generalise with the new data (if possible). Conclusively, there are several arguments mentioned throughout this work regarding the generalisation ability of the ANN that support that the methodology may be used easily with projects of different contexts as the stages of experimentation and validation are very simple, fairly repeatable and analysable.

iv. The method used in Stage two to select the best in accuracy ANN over the testing phase affects the type of networks that are then utilised by ISA. Thus, this method consequently affects the attributes that are finally considered as most 'significant' and are included in the validation subsets. Another possible threat relevant to the attributes that are promoted by the proposed approach is the threshold values of the *Strict*, *Medium* and *Relaxed* filtering levels which are empirically defined. This is another reason why the validation experiments conducted included execution of the Core Cost Estimation (CCE) component with two subsets for each dataset, Val-1 and Val-2, the latter being supplementary to the former, which have resulted from combinations of the above thresholds.

v. The selected method of ISA used in Stage three to calculate the overall connection strength of each input to the output does not consider any impact of the weights connecting the hidden nodes to the output. There are other saliency measures of input variables that calculate the impact of the input vector on the output by using the whole set of connection weights between neurons (e.g., Garson's algorithm [Garson, 91]). Our validation experiments, though, were designed so as to provide an assessment whether in practical, real-world cases a simple and straightforward way of determining the significance of a certain input, like the ISA selection of saliency measure, may be proven robust in revealing the dependencies among the input parameters and effort in the ANN case. Our results suggested that this is highly likely, as the reduced set of attributes suggested by ISA yielded equally accurate cost predictions with the full set.

vi. A final possible threat lies with the fact that the results of data-driven methodologies usually depend on the existence and quality of the respective data. There are cases where the form of data, the presence of collinearity, or even outliers within the data jeopardise the outcome of estimation processes that rely on learning by examples (like the one used in this work) and require further analysis or filtering activities to enhance their effectiveness and assess their appropriateness.

## 5 Conclusions

This work investigates the combination of a Computational Intelligence method with a filtering technique, namely that of Artificial Neural Networks (ANN) and Input Sensitivity Analysis (ISA) respectively, with the overarching aim to model and estimate software development costs. The methodology outlined in this paper suggests a simple, easy and straightforward way to conduct experiments and reduce

the input space of ANN models maintaining at the same time prediction levels. More specifically, the proposed procedure selects a set of important software cost attributes (from a larger pool) and retains effort estimations accuracy at the same levels of using the whole pool of attributes. The selected attributes form a subset of cost drivers that seem to have the strongest influence in predicting the value of development effort in ANN. Thus, ANN may constitute a useful tool in the hands of project managers by estimating the total effort required for the completion of software projects and provide better resource planning, monitoring, coordinating and controlling in the development phases.

The experiments carried out employed different ANN architectures in terms of number of neurons in the input and hidden layers, which were trained with empirical cost data recorded in two publicly available datasets, the Desharnais and the ISBSG. The approach was based on randomly selecting holdout samples for iteratively training, validating and testing the performance of specific ANN topologies. The randomisation removed from the cost models developed chronological dependence of how project data were fed thus removing possible bias. In addition, to obtain conclusive remarks on the best performed ANN in terms of accuracy, a set of objective error measures were employed, potential outliers were removed (to ensure that the ANN were not affected by bias) and results were filtered with respect to three threshold levels. These filtering levels were defined empirically according to a combination of error measures so that the prediction error, that is, the difference between actual and predicted effort values, was taken into account, as well as the positive or negative percentage of evolution follow-up between actual and predicted values, along with the difference from a simple mean-value predictor. Repeating the experiments a sufficient number of times and attaining consistently similar estimation errors strongly suggested that the ANN were stable and able to generalise the knowledge acquired during training.

Subsequently, the cost drivers that were considered more important in describing effort were extracted through a process called Input Sensitivity Analysis (ISA), which suggested an order of significance for the inputs of the ANN developed. When ISA was combined with the filtering thresholds mentioned above, it became evident that some of the attributes were globally accepted, meaning that they were always highly ranked among the leading attributes in each filtering level and could be assumed as more informative than others. In addition, considering the rank of the rest of the attributes we were able to include or exclude some of them in subsequent validation experiments. Subsets containing a reduced number of attributes were formed and a series of validation experiments were executed. The outcome of these experiments revealed that accuracy performance was retained to the same levels as with the original (unreduced) sample data, while the validation subsets worked reasonably well with a simple Multiple Linear Regression (MLR) model.

The results of the methodology were relatively consistent for both datasets as regards the type of attributes that were promoted by ISA. Moreover, most of the significant (promoted) attributes related with scheduling, complexity and functional size attributes, which are attributes known from the early stages of the project (i.e. at the end of the specification phase), accompanied by attributes describing final software adaptations such as Added Count, Changed Count and Deleted Count. The latter proposes that apart from parameters like duration and functional size that are

available from the early phases of the software life-cycle, the overall probability of late changes in the requirements should be taken into account in order to obtain accurate enough approximations of the overall effort value. This means that the cost of change in requirements is considerably high for adding, changing or deleting software functionality and estimations should be repeated in the later stages of the development.

The main contribution of the paper lies with the identification and isolation of the most promising attributes to predict software development effort based on historical samples. This leads to introducing parsimony and greater flexibility in ANN cost models with respect to project parameters as it frees them from too many independent variables. Thus, the reduction of the network's input dimension means that a smaller number of influential attributes to effort estimation is required and therefore fewer measurements need to be collected and monitored along the software process. This, in turn, requires less time, effort and cost for the data gathering activities. Project cost data can be collected much faster thus facilitating the construction of more refined cost models and industry benchmarks.

The methodology described enhances the understanding of dataset features and increases the efficiency and effectiveness of adaptation ANN in the SCE process. Moreover, ANN achieve accurate estimations of software development costs which is a critical prerequisite for any approach. The attributes promoted in each dataset indicate that the duration and scope of a project highly affect effort values and that functional size attributes of some major components (such as outputs, enquiries, interface files of the program under development) have a significant and influential relationship with effort. Moreover, the distinction from the numerical analysis of attributes related to counts of changes and deletions performed during project delivery indicate immediate impact, or correlation of these attributes with effort. The ISA approach also identified different (distant) ranking positions of attributes that are defined as transformations of one to another, i.e., from FS (unadjusted variable) to AFP (adjusted). This provides an indication that possibly the transformation suffers from bias and subjectivity which should be further examined in future work.

In addition, further analyses of the attributes within SCE datasets need to be carried out. Future work may include considering attributes that were excluded from the datasets of this work, such as nominal (categorical) attributes of the Desharnais and the ISBSG, and examination of different characteristics from other available cost datasets, or from the newer version of the ISBSG repository. Also, other methods of ANN sensitivity analysis may be investigated (such as fuzzy curves, change of mean square error, and other modifications and extensions from the ISA measures used,) to rank input feature importance and compare the results with those obtained in this work. One of the main problems faced in this work was the difficulty to handle missing values which resulted in major data reduction. This restriction could be tackled with imputation techniques in future endeavours. Additionally, the datasets could be divided into portions referring to measurements of small, medium and large software systems and then separate ANN models may be developed to perform the analysis presented in this paper for each portion.

Finally, our future research might include further experimentation utilising other datasets and Computational Intelligent methods, such as Genetic Algorithms (GA) for evolving the ANN settings (input and hidden neurons, learning functions and internal

parameters like the momentum and learning rates). Our ultimate goal will be to be able to incorporate enterprise- or organisation- dependent factors in ANN models that will be automatically tuned via the GA and to assess the degree to which a set of inputs measured under the same software development conditions (i.e., team and project attributes) can be used for SCE.

## References

[Albrecht, 79] Albrecht, A.J.: Measuring Application Development Productivity, Proceedings of the Joint SHARE/GUIDE/IBM Application Development Symposium, 1979, 83-92.

[Albrecht, 83] Albrecht, A.J., Gaffney, J.R.: Software Function Source Lines of Code, and Development Effort Prediction: A Software Science Validation, IEEE Transactions on Software Engineering 9 (6), 639-648 1983.

[Angelis, 01] Angelis, L., Stamelos, I., Morisio, M.: Building A Software Cost Estimation Model Based On Categorical Data, Proceedings of the 7th International Symposium on Software Metrics, IEEE Computer Society, 2001, 4-15.

[Azoff, 94] Azoff, E.M.: Neural Network Time Series Forecasting of Financial Markets, John Wiley & Sons, New York, 1994.

[Azzeh, 10] Azzeh, M., Neagu, D., Cowling, P.I.: Fuzzy grey relational analysis for software effort estimation, Empirical Software Engineering 15 (1), 60-90, 2010.

[Belue, 95] Belue, L.M., Bauer, K.W.: Determining input features for multilayer perceptrons, Neurocomputing 7, 111-121, 1995.

[Boehm, 00] Boehm, B.W., Abts, C., Chulani, S.: Software development cost estimation approaches—A survey, Annals of Software Engineering 10 (1), 177-205, 2000.

[Briand, 00] Briand, L.C., Wieczorek, I.: Resource Estimation in Software Engineering, International Software Engineering Research Network, Technical Report, Fraunhofer Institute for Experimental Software Engineering, Germany, ISERN-00-05, 2000.

[Chiu, 07] Chiu, N., Huang, S.: The adjusted analogy-based software effort estimation based on similarity distances, Journal of Systems and Software 80, 628-640, 2007.

[Chulani, 99] Chulani, S., Boehm, B.W., Steece, B.: Bayesian Analysis of Empirical Software Engineering Cost Models, IEEE Transactions on Software Engineering 25 (4), 573-583, 1999.

[Conte, 86] Conte, S.D., Dunsmore, H.E., Shen, V.Y.: Software Engineering Metrics and Models, Benjamin-Cummings Publishing Inc., Menlo Park, California, 1986.

[Desharnais, 89] Desharnais, J.M.: Analyse Statistique de la Productivite des Projects de Development en Informatique a Partir de la Technique de Points de Fonction, MSc. Thesis, Université du Québec, Montréal, 1989.

[Finnie, 97] Finnie, G.R., Wittig, G.E., Desharnais, J.M.: A Comparison of Software Effort Estimation Techniques using Function Points with Neural Networks, Case Based Reasoning and Regression Models, Journal of Systems Software 39, 281-289, 1997.

[Garson, 91] Garson, G.D.: Interpreting neural-network connection weights, AI Expert 6,46-51, 1991.

[Glorfeld, 96] Glorfeld, L.W.: A Methodology for Simplification and Interpretation of Backpropagation-Based Neural Network Models, Expert Systems with Applications 10, 37-54, 1996.

[Haykin, 99] Haykin, S.: Neural Networks: A Comprehensive Foundation, second ed., Prentice Hall, Upper Saddle River, New Jersey, 1999.

[Heiat, 02] Heiat, A.: Comparison of artificial neural network and regression models for estimating software development effort, Information and Software Technology 44 (15), 911-922, 2002.

[Hughes, 97] Hughes, R.T.: An Evaluation of Machine Learning Techniques for Software Effort Estimation, Department of Computing, The University of Brighton, UK, 1997.

[Idri, 02] Idri, A., Khoshgoftaar, T.M., Abran, A.: Can Neural Networks be Easily Interpreted in Software Cost Estimation?, Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ-IEEE) 2002, 1162-1167.

[Idri, 04] Idri, A., Mbarki, S., Abran, A.: Validating and Understanding Software Cost Estimation Models based on Neural Networks, Proceedings of the International Conference on Information and Communication Technologies: From Theory to Applications, 2004, 433-434.

[ISBSG, 05] The International Software Benchmarking Standards Group - Repository Data Release 9, 2005, http://www.isbsg.org/.

[Jeffery, 96] Jeffery, R., Stathis, J.: Function Point Sizing: Structure, Validity and Applicability, Empirical Software Engineering 1 (1), 11-30, 1996.

[Jørgensen, 04] Jørgensen, M.: A Review of Studies on Expert Estimation of Software Development Effort, Journal of Systems and Software 70, 37-60, 2004.

[Jørgensen, 07] Jørgensen, M., Shepperd, M.: A Systematic Review of Software Development Cost Estimation Studies, IEEE Transactions on Software Engineering 33 (1), 33-53, 2007.

[Jørgensen, 95] Jørgensen, M.: Experience with the Accuracy of Software Maintenance Task Effort Prediction Models, IEEE Transactions on Software Engineering 21 (8), 674-681, 1995.

[Karray, 04] Karray, F.O., Silva, C.W.D.: Soft Computing and Intelligent Systems Design: Theory, Tools and Applications, Addison-Wesley, 2004.

[Kitchenham, 01] Kitchenham, B., MacDonell, S., Pickard, L., Shepperd, M.: What Accuracy Statistics Really Measure, IEE Proceedings of Software Engineering 148 (3), 81-85, 2001.

[Kumar, 08] Kumar, K.V., Ravi, V., Carr, M., Kiran, N.R.: Software Development Cost Estimation using Wavelet Neural Networks, Journal of Systems and Software 81, 1853-1867, 2008.

[Leung, 02] Leung, H., Fan, Z.: Software Cost Estimation, Handbook of Software Engineering and Knowledge Engineering, Vol. 2, S.K. Chang (Ed.), World Scientific, 2002.

[Lokan, 99] Lokan, C.J.: An Empirical Study of the Correlations Between Function Point Elements [software Metrics], Proceedings of the Sixth International Software Metrics Symposium, Boca Raton, FL, USA, 1999, 200-206.

[MacDonell, 97] MacDonell, S.G., Gray, A.R.: A Comparison of Modeling Techniques for Software Development Effort Prediction, Proceedings of International Conference on Neural Information Processing and Intelligent Information Systems, Dunedin, New Zealand, Springer-Verlag, 1997, 869-872.

[Mair, 00] Mair, C., Kadoda, G., Lefley, M., Phalp, K., Schofield, C., Shepperd, M., Webster S.: An Investigation of Machine Learning based Prediction Systems, Journal of Systems and Software 53 (1), 23-29, 2000.

[Mann, 47] Mann, H.B., Whitney, D.R.: On a test of whether one of two random variables is stochastically larger than the other, Annals of Mathematical Statistics 18, 50-60, 1947.

[McCulloch, 43] McCulloch, W.S., Pitts, W.: A logical calculus of the ideas immanent in nervous activity, Bulletin of Mathematical Biology 5 (4), 115-133, 1943.

[Moløkken, 03] Moløkken, K., Jørgensen, M.: A review of software surveys on software effort estimation, Proceedings of the International Symposium on Empirical Software Engineering (ISESE), 2003, 223-230.

[Olden, 02] Olden, J.D., Jackson, D.A.: Illuminating the "black box": a randomization approach for understanding variable contributions in artificial neural networks, Ecological Modelling 154, 135-150, 2002.

[Park, 08] Park, H., Baek, S.: An empirical validation of a neural network model for software effort estimation, Expert Systems with Applications 35, 929-937, 2008.

[Port, 08] Port, D., Korte, M.: Comparative Studies of the Model Evaluation Criterions *MMRE* and *pred* in Software Cost Estimation Research, Proceedings of the 2nd ACM-IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM), Kaiserslautern, Germany 2008, 51-60.

[Refenes, 95] Refenes, A.N., Kollias, C., Zarpanis, A.: External Security Determinants of Greek Military Expenditure: An Empirical Investigation Using Neural Networks, Defence and Peace Economics 6, 27-41, 1995.

[Samson, 97] Samson, B., Ellison, D., Dugard, P.: Software Cost Estimation using an Albus Perceptron (CMAC), Information and Software Technology 39, 55-60, 1997.

[Satizábal, 07] Satizábal, H.M., Pérez-Uribe, A.: Relevance Metrics to Reduce Input Dimensions in Artificial Neural Networks, Artificial Neural Networks-ICANN, Springer Berlin / Heidelberg, 2007, 39-48.

[Serluca, 95] Serluca, C.: An Investigation into Software Effort Estimation using a Back-Propogation Neural Network, MSc. Thesis, Bournemouth University, 1995.

[Sommerville, 07] Sommerville, I.: Software Engineering, eighth ed., Addison-Wesley Longman Publishing Co., Inc., 2007.

[Srinivasan, 95] Srinivasan, K., Fisher, D.: Machine Learning Approaches to Estimating Software Development Effort, IEEE Transactions on Software Engineering 21 (2), 126-137, 1995.

[Tronto, 08] Tronto, I.F.D.B., Silva, J.D.S.D., Sant'Anna, N.: An Investigation of Artificial Neural Networks based Prediction Systems in Software Project Management, Journal of Systems and Software 81, 356-367, 2008.

[Wilcoxon, 45] Wilcoxon, F.: Individual comparisons by ranking methods, Biometrics 1, 80-83, 1945.

[Wittig, 97] Wittig, G.E., Finnie, G.R.: Estimating Software Development Effort with Connectionist Models, Information and Software Technology 39 (7), 469-476, 1997.