Estimators Characteristics and Effort Estimation of Software Projects

Hrvoje Karna and Sven Gotovac
Department of Electronics, Faculty of Electrical Engineering, Mechanical Engineering and Naval Arhitecture, University of Split, Ruđera Boškovića 32, Split, Croatia

Keywords: Software Effort Estimation, Human-Judgment, Data Mining, Neural Networks.

Abstract: Effort estimation is an important part of software project management. Accurate estimates ensure planned project execution and compliance with the set time and budget constraints. Despite attempts to produce accurate estimates by using formal models there is no substantial evidence that these methods guarantee better estimates than those experts make. In order to improve the effort estimation process it is crucial to enhance understanding of the human estimator. When producing estimates each expert exhibits mental effort. In such situation estimator relies on his personal characteristics, some of which are, in context of effort estimation, more important than others. This research tries to identify these characteristics and their relative influences. Data for the research have been collected from projects executed in large company specialized for development of IT solutions in telecom domain. For identification of expert characteristics data mining approach is used (the multilayer perceptron neural network). We considered the use of this method as it is similar to the way human brain operates. Data sets used in modelling contain more than 2000 samples collected from analysed projects. The obtained results are highly intuitive and later could be used in the assessment of reliability of each estimator and estimates he produces.

1 INTRODUCTION

The software development life cycle consists of phases and related activities designed to ensure the building of the final product. Each activity within these phases is represented as work item or a task that has to be completed so that the work can progress. As different items interact with each other, delay or defect in one affects the completion of another. This often results in software deliveries that are behind schedule, exceeding planned budget and possibly poor quality. Studies report that 60-80% projects encounter effort and/or schedule overruns and that the average cost overruns are 30-40% (Moløkken, 2004).

In this paper we focus on expert estimation as today it is a dominant estimation strategy. Expert estimation is performed by a human-expert, where the estimate generation is hidden from us as it is based on estimators mental processes i.e. a major part of estimation is based on intuition (Jørgensen, 2000). During this process, based on the given input information, the estimator uses his judgment capability that largely depends on his personal characteristics, background (Boetticher, 2006) but also environment in which estimation is generated (project, organization, etc.) (Humphrey, 2007; Wang, 2007).

The reasons for use of this effort estimation strategy in software development process are obvious. Firstly it is the ease of its implementation. Secondly, the evidence suggests that the use of formal estimation methods does not lead to more accurate estimates (Moløkken, 2004; Jørgensen, 2004 and Jørgensen, 2009). Furthermore, estimation by analogy is hard to implement in environments where there are no similar previous projects (Shepperd, 1996; Shepperd, 1997 and Keung, 2009). In short, other methods don't have comparative benefits and they don't guarantee better estimates in comparison to those experts produce. All these reasons explain the fact why the expert or human-based effort estimation remains the dominant technique (Jørgensen, 2004; Hill, 2000).

Despite its comparative advantages some challenges still remain and they are mainly linked to the very nature of how the estimates are produced, their accuracy and causes of effort estimation errors

Karna H. and Gotovac S..
Estimators Characteristics and Effort Estimation of Software Projects.
DOI: 10.5220/0005002600260035
In *Proceedings of the 9th International Conference on Software Engineering and Applications* (ICSOFT-EA-2014), pages 26-35
ISBN: 978-989-758-036-9

(Jørgensen, 2000; Lin, 2008 and Jørgensen, 2014). This becomes even more important especially in today's environment when software becomes more complex and dynamics of development process increase. Among other, these reasons also lead to the wide acceptance of the expert estimation methods in various agile development approaches (Cheng, 2012 Coelho, 2012; Zulkefli, 2011 and Ziauddin, 2012). To increase understanding of how estimators cope with these issues we have to detect estimator inherent characteristics that he relies on when producing effort estimates.

To improve software engineering practice engineers are increasingly applying various advanced techniques in everyday work. Data mining algorithms are such an example and as recent studies report software engineering can benefit from use of this approach (Xie, 2009; Layman, 2008). These algorithms can help engineers to figure out facts and relations previously not identified or obvious. Data mining in terms of software engineering consists of collecting software engineering data, extracting knowledge and if possible using this knowledge to improve the software engineering process. In this study we use neural networks to build predictive models. The relationships between target and predictors are determined during the learning process. Although it is sometimes hard to interpret the results from such models in this study we find them to be intuitive and in line with common sense.

The remaining part of this paper is organized as follows: section 2 quotes related research in this area. Section 3 describes the design of study that was conducted (the data mining approach, methodological framework for the study and setup of experiment). In section 4 survey results and their implications are discussed. Section 5 gives the conclusion and directions for future research.

2 RELATED RESEARCH

The study of software engineering economics and effort estimation is a long running topic lasting from early work of Boehm (Boehm, 1981), Albrecht et al. (Albrecht, 1983), Shepperd (Shepperd, 2007) and others. These attempts resulted in number of different effort estimation models over last decades. In general models for estimating software development effort can be classified into three categories: formal, analogy and expert based estimation models.

Valuable information concerning different aspects of effort estimation in general and particularly expert effort estimation have been published in works of Jørgensen, Moløkken, Grimstad, and others (Moløkken, 2004; Grimstad, 2007). Studies give evidence that models fail to systematically perform better than the experts when estimating the effort required to complete software development tasks. This can be attributed to the natural advantages that experts typically possess and flexibility in how they process the information. Two conditions probably lead to this: the models are not calibrated to the organizations using them, and that the experts process important contextual information that is not included in formal models and apply it efficiently (Jørgensen, 2005; Jørgensen, 2007). All these facts led to the situation that estimating effort on basis of expert judgment is the most common approach today. Identifying estimators characteristics that matter the most in case of the expert estimation still remains as a challenge. Conducted research suggests that improving software effort estimation doesn't necessarily require introduction of sophisticated formal estimation models or expensive project experience databases (Jørgensen, 2005). Rather than building complex predictor models, empirical software engineer researchers should focus on the humans making those estimates (Boetticher, 2007; Faria 2012).

Also, there are publications on application of data mining techniques in software engineering in general (Xie, 2009). Also, a lot of research is conducted in the area of application of artificial intelligence and neural networks in particular to the field of software effort estimation (Tadayon, 2005; Satyananda, 2009; Singh, 2011 and Abbas, 2012). The review of such articles concerning use of neural network based models for software effort prediction is available in (Dave, 2012). Yet, when it comes to the combination of these two, expert or human based effort estimation and data mining techniques, there are relatively few studies. Some valuable work can be found in studies by Boetticher (Boetticher, 2001; Boetticher, 2006 and Boetticher, 2007). More research is required in pursuit to identify experts characteristics that matter the most to the success of effort estimation process.

As quantities of software engineering data become greater many opportunities emerge. This suggests that there is more research needed in this area as results seem to be promising not only from the theoretical perspective but also because of its implications in every day practice of software engineers (Xie, 2013). Integrating effective data mining techniques into every day practice and the use of interdisciplinary approach in research could produce valuable insights into this currently insufficiently explored filed.

3 STUDY DESIGN

The data used in this study have been collected within a large international company specialized for IT solutions development in telecom domain. The Croatian branch of that company counts more than 350 employees located on several locations. Most of the employees are project managers and software engineers responsible for handling development and maintenance tasks on different local and international projects.

In well organized, structured development process the execution of work over all project phases is tracked. Today this is often done by using some form of tool that supports such activity. These tools allow not only creation and handling of work items but sometimes also serve as source code repositories, wiki, collaboration, and reporting environments that support development process. Usually these tools allow some form of export capability which can be useful for different forms of data analysis.

This research was conducted so that participants were not aware of the study. For the purpose of the study two main data sources were used:

- Tracking system i.e. application lifecycle management tool implemented on projects that support development process. In this case it primarily served as a central place for collection of work item data. For this purpose on all considered projects Microsoft Team Foundation Server was used. Advantage that this and similar tools offer is the capability of various forms of data presentation, manipulation and export. The last capability, export of data in various forms, supports data mining process.
- The estimators data, gathered from the various internal and external sources, allows the creation of experts or estimators profile. This data was later structured in format that enabled manipulation and linking of profile data with data exported from tracking system.

What is also important is that data for all seven projects being analysed have been gathered in relatively short time interval, in this case a few days, so that it did not allow significant changes in either project tracking system entries or employee profile data. This way so called snapshots of both project items and profiles were created.

For all employees involved on projects, collected profile data were structured in appropriate form, this made the total of 36 estimator profiles that entered the analysis. Input variables that are used to represent estimators profile characteristics are logically organized into several groups or segments, these are: general, education, experience, position (role and responsibilities) and competences.

General variables identify estimators gender and chronological age.

The education group of variables contains data regarding estimators degree (achieved education level) and the field of education.

The experience group contains information about the total estimators experience, the experience working for the current company in which this research is conducted (both expressed in years), the length of experience on current the project (for which modelling is done, expressed in months) and the number of projects the estimator has worked on.

The experts position in the company and on the project is expressed by the group of variables that define employees role (project manager, software engineer, etc.) and the set of responsibilities that are assigned to him (software development, test and verification, etc.). The reason for using such classification to define employees position are based on two facts: first, it is the company's internal classification of functions and the second is internal organization of projects being executed in which the number of team members (average of 7±2 per team) encourages the cross functional setup in which (principally excluding project manager) team members are responsible for all types of work (writing specifications, design and test documents, implementation, testing and verification and even customer support activities). This setup is the characteristic of all seven projects being analysed in this study. Finally, position is defined by the position level (being either junior, advanced or senior).

Last group of variables covers expert competences (skills and knowhow) in area of development specific know how (tools and programming languages), solution specific know how (systems, equipment and technology used on a project), product know how (components for current system being built as well as features of previous versions and integrated modules), professional know how (areas of professional occupation, current and previous) and other know how (existence of certifications, (non)formal skills, etc.), each ranked as either basic, advanced or expert. It is important to state that this evaluation is based on self-assessment i.e. during organized data collecting session each employee fills a predefined form ranking his competences in each segment.

Data exported from tracking system contain both reference to an item owner and assigned efforts. This allowed two things: first, linking of an item to estimators profile and second, calculation of estimation error.

As a measure of estimation accuracy the magnitude of relative error is used (MRE) (Conte, 1986), defined as

$$MRE = \frac{abs(actual\ effort - estimated\ effort)}{actual\ effort}$$

The MRE is by far the most widely used measure of effort estimation accuracy (Stensrud, 2003; Ferrucci, 2010 and Basha, 2010) it is basically a degree of estimation error in an individual estimate. Based on values of actual and estimated effort MRE was calculated for each item i.e. estimation entity in data set. During study we used other criteria (e.g. MER, BRE) to assess the performance of effort estimation models but MRE produced best results.

For all work items extracted from each project both operations were done. As a result of previous operations seven data sets, one for each project, were created.

3.1 Data Mining Approach

Data mining analysis of in principle large data sets is conducted with goal of discovering relations and patterns in data and their representation in ways that provide new and understandable information to the user. These insights can then be used in decision making process to enhance its quality and effectiveness. The use of machine learning methods as a form of artificial intelligence in this type of research seemed obvious.

Artificial neural network, or simply neural network, can be defined as a biologically inspired computation model which consists of a network architecture composed of artificial neurons. This structure contains a set of parameters, which can be adjusted to perform certain task. Due to their similarity to the human brain, neural networks are useful models for problem-solving and knowledge-engineering in a way very similar to that of a human (Gonz´alez, 2008). They tend to express a nonlinear function by assigning weights to input variables, accumulate their effects and produce an output following some sort of decision function.

As it is mentioned, these systems therefore function similar to the way human brain works - by passing impulses from neuron to neuron across synapses creating a complex network that has the ability of learning (Nisbet, 2009). This ability to train and learn from experience to form decision or judgment has made neural networks the first method of choice for our study.

There are many different types of neural networks, in our research we used the multilayer perceptron with single hidden layer and back-propagation learning. The multilayer perceptron is characterized by a neuron model, network architecture, associated functional elements and training algorithm. Figure 1. represents applied architecture of neural network in our model.

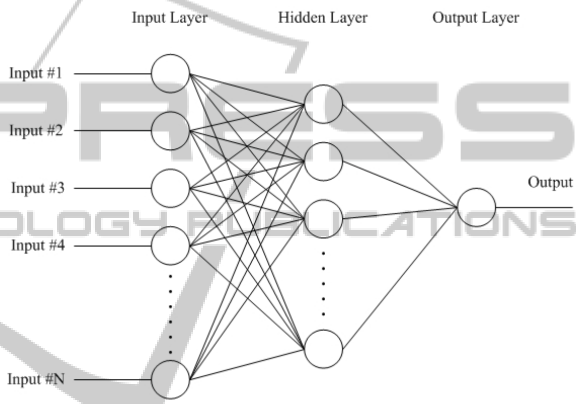

Figure 1: Architecture of neural network applied in model.

Each node in a model is a so called perceptron. The perceptron receives information in some form of input signal, integrates it with a set of parameters and produces a single output signal. Similar to biological neural system an artificial neural network is built of neurons in network architecture. The architecture of this network defines number of neurons their arrangement and connectivity. The multilayer perceptron uses so called feed-forward network architecture. Neurons from input layer are connected to nodes of a hidden layer and every node from hidden layer is connected to a node in output layer. Input layer represents raw information that is fed into the network, in our case represented by set of predictors. Every input is sent to the nodes in hidden layer. Hidden layer accepts data from the input layer. It uses input values and modifies them using some weight value. The activation function defines the output signal from the neuron. There are many activation functions, in our case the most common one, the sigmoid function is used. This new value is than sent to the output layer but it will be modified by some weight from connection between hidden and output layer. Output layer process information received from the hidden layer and produces an output. The back-propagation looks for the minimum of the error function. As a stopping rule the error cannot be further decreased criteria is used. The combination of weights which minimize the error function is considered to be a solution of the learning problem. Although a single neuron can solve some simple learning tasks, the power of neural network comes from connecting many neurons in network architecture. A learning algorithm is an adaptive method by which a network of computing units self-organizes to implement a desired behaviour. This is a closed loop of presentations of examples and of corrections to the network parameters. Learning of neural network is done in training phase during which a learning algorithm adapts the network parameters according to previous experience until a solution, if it exists, is found (Rojas, 1996).

Another method of machine learning that could be considered in future research is the use of decision trees. Application of these methods could be interesting because of their ability to find rules for separation i.e. classification of input set of variables and the fact that it could provide us with more readable models that are, in comparison to neural networks, relatively easy to interpret. This remains to be more closely studied.

3.2 Methodological Framework

Building of the data mining model considered in this research required the definition of analysis objectives. In this case it is the identification of the expert estimators profile characteristics and their relative importance in producing reliable effort estimates. This business objective was mapped to data mining objective with intention to create such a model that could later be implemented in every day practice.

Methodological framework consists of following phases:

1) Data Collection: during which both work item and employee profile data were collected. This stage therefore included export of project tasks, identification of involved team members and structuring of their profile data.

2) Data Preparation: at this stage data was processed according to specific needs of model building process. The end products are data sets that contain data of each item and related employee (item was assigned to) for each project. This way seven data sets were generated. At this stage the outliers, extremes and missing data are handled.

3) Data Partitioning: input data is randomly divided into two sets: training and test data sets. On each project the ratio of 2/3 of the data is used for the training and 1/3 for the testing phase, following standard data mining practice. The training data sets are used to build models. Models are then tested using test data to assess their performance.

4) Model Building: during this phase the predictive models are built using neural network algorithms and are evaluated for their accuracy and predictive performance.

3.3 Experiment Setup

Observed projects for which data sets were collected were executed in same department of the aforementioned company. This department is specialized in development of solutions for telecom operators. Technologies used on projects are similar and based on Microsoft stack (Team Foundation Server, Visual Studio, C#, .NET, MS SQL, etc.). All projects followed sequential i.e. waterfall development methodology.

As mentioned earlier data was gathered from different sources so for each work item, profile data had to be joined to form valid data sets entry. After that data sets were cleaned and aggregated to produce input data files of total 2102 records corresponding to projects being analysed. Projects data sets displayed variability in terms of number of initial items extracted from tracking system and the amount of invalid data. As a result, input data sets entering modelling phase differ in size i.e. number of items. So early in a phase of data collection and structuring initial data cleaning was performed. This way the quantity of data was decreased by something more than 30% but the quality of data was significantly improved. The input sets per each project differ in size ranging from few dozen to few hundred items. This amount of invalid data raised an interest to conduct the study that would investigate the amounts of invalid data in other sets. Variables considered in the input data sets are listed in Table 1. From the input set of variables 18 are used as predictors and single variable (MRE) as a target.

Experiment was conducted using IBM SPSS Modeler 14.2. For each project being analysed a stream or flow of execution was developed to perform the experiment. The experiments followed the sequence in which data is initially fed into the stream after which it passed steps of preparation, transformation and partitioning before it entered the modelling element.

Table 1: Predictors and target in input dataset.

Profile Segment	Variables	
	Name	*Type*
General	Gender	Predictor
	Age	
Education	Degree Title	
	Education Field	
Experience	Total Experience	
	Company Experience	
	Project Experience	
	Project Count	
Position	Role	
	1st Responsibilty	
	2nd Responsibility	
	3rd Responsibility	
	Position Level	
Competences	Development Specific Know How	
	Solution Specific Know How	
	Product Specific Know How	
	Professional Specific Know How	
	Other Know How	
	MRE	Target

Quality of input data was evaluated early in stage just after data collection was performed. This means that entries with invalid effort values (e.g. missing actual and/or estimated effort values) were removed from further processing as for those entries it was not possible to calculate MRE. Similar, tasks that were not linked to owner i.e. estimator were also eliminated from data set as connection between estimator and estimate was missing. These issues can be attributed to the bad task handling discipline a thing that can be more influenced in every day work practice. But again these are genuine data so we could expect such behaviour. During later stages of modelling outliers, extremes and missing values variable values were handled as they can negatively affect the model precision. All these actions are supported by specialized Modeler elements.

The modelling element implements the multilayer perceptron neural network that forms the relationships between the target and predictors during the learning process. During formation of the neural network the model determines how the network connects the predictors to the target. This is done by so called hidden layer and although each hidden unit is some function of predictors basically its internal configuration is unobservable.

4 SURVEY RESULTS

The outputs resulting from the modelling are overall model accuracy and relative importance of top predictors. As it is said the importance of each predictor is relative to the model and it identifies the input variables that matter the most during prediction process. The overall model accuracy is an indicator of the accuracy of predictions that states whether or not the whole model is accurate and it is expressed in percentages. Table 2. summarizes the results of predictive performance of all 18 predictors for each project. The average accuracy of built models for all seven analysed projects is 63.80%.

It is interesting to see that some models have low accuracy (for example Projects 1 and 2) although they significantly differ in number of input data set entries. On the other hand Projects 2 and 3 have comparable number of initial and input items and relatively similar and low model accuracy. In general we can say that models with 60% and greater accuracy can give us valuable insight into predictive power of input variables (Projects 4, 5, 6 and 7). Data sets of these projects typically have higher proportion of hits i.e. correct estimates.

Table 2: Output model accuracy.

Project	Accuracy
Project 1	45.00%
Project 2	40.80%
Project 3	51.90%
Project 4	78.20%
Project 5	76.90%
Project 6	62.50%
Project 7	91.30%
Average:	**63.80%**

Table 3. displays the assessment of predictor performance of classifiers over all seven projects. For each project, based on input vector of 18 predictors used in a model and a given data set, on the output neural network returns the vector of 10 predictors with greatest predictive power.

Table 3: Predictors rank (*R*) and relative importance (*I*) in observed projects.

Project	1		2		3		4		5		6		7	
Predictor	*R*	*I*	*R*	*I*	*R*	*I*	*R*	*I*	*R*	*I*	*R*	*I*	*R*	*I*
Gender	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Age	4	0.08	-	-	**3**	0.08	7	0.05	-	-	6	0.07	6	0.06
Degree Title	-	-	4	0.08	6	0.05	**3**	0.07	-	-	5	0.08	-	-
Education Field	-	-	5	0.07	-	-	-	-	10	0.05	-	-	9	0.05
Total Experience	**2**	0.10	**2**	0.15	4	0.08	**2**	0.17	4	0.10	4	0.08	8	0.06
Company Experience	**1**	0.12	**1**	0.21	**1**	0.29	6	0.05	**3**	0.10	**3**	0.09	**2**	0.14
Project Experience	**3**	0.09	**3**	0.11	7	0.05	**1**	0.18	-	-	8	0.06	**3**	0.13
Project Count	8	0.06	6	0.06	**2**	0.09	4	0.07	7	0.07	**1**	0.17	-	-
Role	-	-	8	0.04	9	0.04	8	0.04	5	0.07	9	0.05	5	0.07
Position Level	-	-	-	-	-	-	10	0.04	-	-	-	-	-	-
1st Responsibility	-	-	9	0.04	5	0.06	-	-	8	0.06	10	0.05	-	-
2nd Responsibility	9	0.05	-	-	-	-	-	-	**1**	0.12	7	0.07	**1**	0.14
3rd Responsibility	10	0.05	10	0.03	10	0.04	-	-	6	0.07	**2**	0.13	10	0.05
Development Specific Know How	-	-	-	-	8	0.04	-	-	9	0.06	-	-	4	0.09
Solution Specific Know How	6	0.06	-	-	-	-	9	0.04	-	-	-	-	7	0.06
Product Specific Know How	5	0.08	-	-	-	-	5	0.06	-	-	-	-	-	-
Professional Specific Know How	7	0.06	7	0.04	-	-	-	-	**2**	0.11	-	-	-	-
Other Know How	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Predictive power of each predictor is relative to the model i.e. project for which data mining was performed. Based on their occurrences i.e. incidence and relative importance we can make assessment of general predictive power of the predictors under consideration. This way we can identify those predictors that matter the most and can be considered important and ignore those with the low predictive power. Figure 2. displays predictor occurrences and their relative importance in resulting models for all seven projects.

Based on modelling results and analysis we can conclude that the typical predictors of estimation performance in our study are the group of predictors that represent experts experience (total experience, company experience, project experience and number of projects expert has participated).

The use of experts position (role and responsibilities) seems to be the second tier of predictors. This is followed by employees age and education level. Other predictors show low predictive power. It is somewhat surprising that group of predictors that represent experts competences did not show greater predictive power as we initially expected. This could be, at last partially, a result of the self-assessment process i.e. subjectivity of each estimator when assessing his competences. It suggests that more structured form of employee competence evaluation is needed that should not only be concerned with employee current project assignment but has to cover a much broader perspective. For this to be done involvement of other segments of organization is necessarily required. These insights gave us directions for some future work and investigation, possibly on greater data sets.

Findings from this study, based on results of modelling and later analysis, greatly confirmed our initial premises that experience and position significantly determine experts effort estimation reliability. This was not the case in respect to competences importance but it helped us determine valuable directions for future studies.

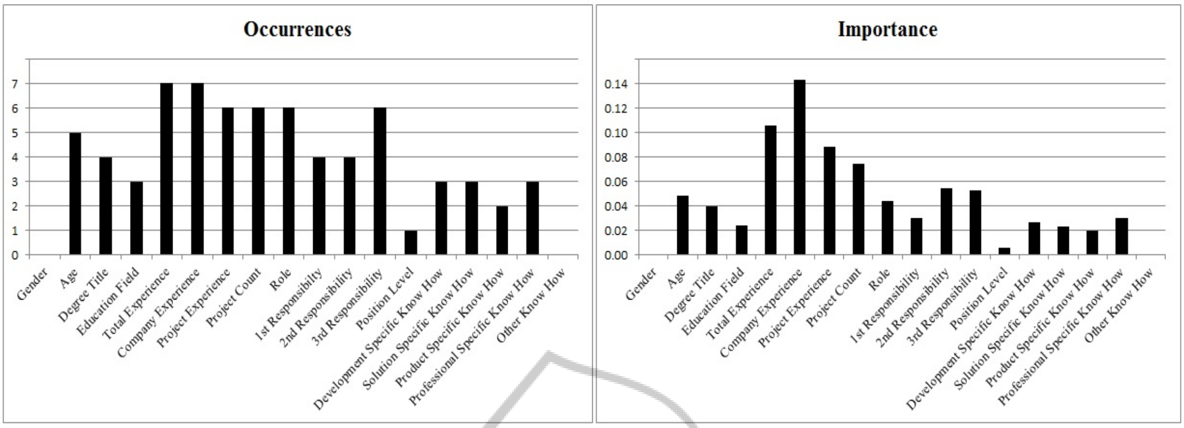

Figure 2: Predictor occurrences and relative importance in resulting models.

5 CONCLUSIONS

This research was conducted with intention to identify predictors (defined as experts profile characteristics) that could be used to assess experts reliability and assure future accurate effort estimates.

The paper reports a detailed description of the methodology used to develop predictive models in software engineering filed of effort estimation. Motivation comes from the need of introducing modelled approach of assessing expert performance in effort estimation and later estimation capability of each estimator based on his characteristics. The methodology was applied on real data extracted from the tracking system used on projects and collected employee profile data.

Results of this and future studies intend to support development of a model for enhanced expert effort estimation. Such a model is intended to enhance reliability of estimates and could be applied to everyday practice of software engineers. Based on better understanding of effects that estimators characteristics have on reliability of effort estimates it would allow the application of corrective measures at early stage of estimation process. As effort estimates constitute an important part of software project management, enhances in this area can bring substantial improvements to organizations implementing it. This is obvious when we know that reliability of conducted estimates affects both time and budget constraints but also development process efficiency. In short, such a model could significantly improve the efficiency of various aspects of software project management.

Possible limitation of the study is the fact that project data was collected form projects executed in department of single company, therefore the results are best describing this particular environment. On the other hand used research methodology is general and could be applied elsewhere.

As the research progresses this model will be upgraded and possibly deployed by using data from projects executed in different organizations. The goal is to advance understanding of expert effort estimation by investigating the impact of profile and project characteristics on experts effort estimation.

6 FUTURE DIRECTIONS

In time to come additional set of experiments will be conducted that will include data related to project context. This is planned with aim to better understand how project characteristics affect expert estimation.

Another challenge will be evaluation of other data mining techniques that could be used to produce consistent and robust model for evaluation of expert estimation reliability.

REFERENCES

Albrecht, A. and Gaffney, J. (1983) ‘Software function, source lines of code, and development effort prediction: a software science validation’, *IEEE Transactions on Software Engineering*, vol. 9, no. 6, November, pp. 639-648.

Abbas, A. S. et al, (2012) ‘Neural Net Back Propagation and Software Effort Estimation’, *ARPN Journal of Systems and Software*, vol. 2, no. 6, June.

Basha, S. and Ponnurangam, D. (2010) ‘Analysis of Empirical Software Effort Estimation Models’, International Journal of Computer Science and Information Security, vol. 7, no. 3, March.

Boehm, B., (1981) 'Software Engineering Economics', Englewood Cliffs, Prentice Hall, NJ, USA.

Boetticher, G., Lokhandwala, N., and Helm, J. (2006) 'Understanding the Human Estimator', *Second International Predictive Models in Software Engineering (PROMISE) Workshop co-located at the 22nd IEEE International Conference on Software Maintenance*, Philadelphia, PA.

Boetticher, G. and Lokhandwala, N. (2007) 'Assessing the Reliability of a Human Estimator', *Third International Predictive Models in Software Engineering (PROMISE) Workshop as part of the International Conference on Software Engineering*, Minneapolis, MN.

Boetticher, G. (2001) 'Using Machine Learning to Predict Project Effort: Empirical Case Studies in Data-Starved Domains', *Model Based Requirements Workshop*, San Diego, pp. 17 – 24.

Cheng, B. and Xuejun, Y. (2012) 'The Selection of Agile Development's Effort Estimation Factors based on Principal Component Analysis', Proceedings of International Conference on Information and Computer Applications, vol. 24, pp. 112.

Coelho, E. and Basu, A. (2012) 'Effort Estimation in Agile Software Development using Story Points', International Journal of Applied Information Systems (IJAIS), August, vol. 3, no. 7.

Conte, S. D., Dunsmore, H. E., and Shen, V. Y. (1986) 'Software Engineering Metrics and Models', Menlo Park, CA, Benjamin-Cummings.

Dave, V. S. and Dutta K. (2012) 'Neural network based models for software effort estimation: a review', *Artificial Intelligence Review*.

Faria, P. and Miranda E. (2012) 'Expert Judgment in Software Estimation during the Bid Phase of a Project – An Exploratory Survey', Software Measurement and the 2012 Seventh International Conference on Software Process and Product Measurement (IWSM-MENSURA), October 2012 Joint Conference of the 22nd International Workshop, pp. 126-131.

Ferrucci, F. et al (2010) 'Genetic Programming for Effort Estimation an Analysis of the Impact of Different Fitness Functions', *2nd International Symposium on Search Based Software Engineering*, Benevento, Italy, September, pp. 89-98.

Gonz´alez, L. R. (2008) 'Neural Networks for Variational Problems in Engineering', PhD thesis. Technical University Catalonia.

Grimstad, S. and Jørgensen, M. (2007) 'Inconsistency of Expert Judgment-based Estimates of Software Development Effort', Journal of Systems and Software archive, vol. 80, no. 11, November, pp. 1770-1777.

Hill, J., Thomas, L.C., and Allen, D.E. (2000) 'Experts estimates of task durations in software development projects', *International Journal of Project Management*, vol. 18, no. 1, February, pp. 13-21.

Humphrey, W.S. et al (2007) 'Future Directions in Process Improvement'. *CrossTalk The Journal of Defense Software Engineering*, vol. 20, no. 2, February, pp.17-22.

Jørgensen, M. et al (2000) 'Human judgement in effort estimation of software projects', *Presented at Beg, Borrow, or Steal Workshop, International Conference on Software Engineering*, June, Limerick, Ireland.

Jørgensen, M. (2004) 'Top-down and Bottom-Up Expert Estimation on Software Development Effort', *Journal of Information and Software Technology*, vol.46, no. 1, January, pp. 3-16.

Jørgensen, M., Boehm, B. and Rifkin, S. (2009) 'Software Development Effort Estimation: Formal Models or Expert Judgment?', *IEEE Software*, vol. 26, no. 2, March-April, pp. 14-19.

Jørgensen, M. (2007) 'Estimation of Software Development Work Effort: Evidence on Expert Judgment and Formal Models', *International Journal of Forecasting*, vol. 23, no. 3, pp. 449-462.

Jørgensen, M. (2005) 'Practical guidelines for expert-judgment-based software effort estimation', *IEEE Software*, vol. 22, no. 3, May-June, pp. 57-63.

Jørgensen, M. (2014) 'What We Do and Don't Know about Software Development Effort Estimation', IEEE Software, vol. 31 no. 2, pp. 37-40.

Keung, J. (2009) 'Software Development Cost Estimation using Analogy: A Review', *In proceeding of 20th Australian Software Engineering Conference,* Gold Cost, Australia, April, pp.327-336.

Layman, L. et al (2008) 'Mining Software Effort Data: Preliminary Analysis of Visual Studio Team System Data', *Proceedings of the 2008 International Working Conference on Mining Software Repositories*, May, pp.43-46.

Lin S. W and Bier V. M. (2008) 'A study of expert overconfidence', Reliability Engineering & System Safety, vol. 93, no. 5, pp. 711-721.

Moløkken, M. and Jørgensen, M. (2004) 'A review of surveys on Software Effort Estimation', *International Symposium on Empirical Software Engineering*, September-October, pp. 223-230.

Nisbet, R., Elder, J. and Miner, G. (2009) 'Handbook of Statistical Analysis and Data Mining Applications', Elsevier Inc.

Rojas, R. (1996) 'Neural Networks - A Systematic Introduction', Springer-Verlag, Berlin, New-York.

Satyananda, C. R. and Raju, K. (2009) 'A Concise Neural Network Model for Estimating Software Effort', *International Journal of Recent Trends in Engineering*, vol. 1, no. 1, May.

Singh, J. and Sahoo, B. (2011) 'Software Effort Estimation with Different Artificial Neural Network', *International Journal of Computer Applications (IJCA) - Special Issue on 2nd National Conference - Computing, Communication and Sensor Network (CCSN)*, vol. 4, pp. 13-17.

Shepperd, M., Schofield, C., and Kitchenham, B. (1996) 'Effort estimation using analogy', *In International Conference on Software Engineerin*, March, pp. 170–178.

Shepperd, M. and Schofield, C. (1997) 'Estimating software project effort using analogies', *IEEE Transactions on Software Engineering*, vol. 23, no. 11, November, pp. 736–743.

Shepperd, M. (2007) 'Software project economics - a roadmap', *Future of Software Engineering - 29th International Conference on Software Engineering*, Minneapolis, MN, USA, May, pp. 304-315.

Stensrud, E. et al (2003) 'A Further Empirical Investigation of the Relationship Between MRE and Project Size', *Empirical Software Engineering*, vol. 8, no. 2, pp. 139-161.

Tadayon, N. (2005) 'Neural Network Approach for Software Cost Estimation', *Proceedings of the International Conference on Information Technology: Coding and Computing*, vol. 2, April, pp. 815-818.

Wang, Y. (2007) 'On Laws of Work Organization in Human Cooperation', *International Journal of Cognitive Informatics and Natural Intelligence*, vol. 1, no. 2, pp. 1-15.

Xie, T. (2013) 'Synergy of Human and Artificial Intelligence in Software Engineering', *In Proceedings of the 2nd International NSF sponsored Workshop on Realizing Artificial Intelligence Synergies in Software Engineering*, San Francisco, CA.

Xie, T. et al (2009) 'Data Mining for Software Engineering', *IEEE Computer*, vol. 42, no. 8, August, pp. 35-42.

Rojas, R., (1996) 'Neural Networks - A Systematic Introduction', Springer-Verlag, Berlin, NY, USA.

Ziauddin et al. (2012) 'An Effort Estimation Model for Agile Software Development', Advances in Computer Science and its Applications (ACSA), vol. 2, no. 1.

Zulkefli M. et al. (2011) 'Review on Traditional and Agile Cost Estimation Success Factor in Software Development Project', International Journal on New Computer Architectures and Their Applications (IJNCAA), vol. 1, no. 3, pp. 942-952.