The Use of Bayesian Networks for Web Effort Estimation: Further Investigation

Emilia Mendes
The University of Auckland
emilia@cs.auckland.ac.nz

Abstract

The objective of this paper is to further investigate the use of Bayesian Networks (BN) for Web effort estimation when using a cross-company dataset. Four BNs were built; two automatically using the Hugin tool with two training sets; two using a structure elicited by a domain expert, with parameters obtained from automatically fitting the network to the same training sets used in the automated elicitation (hybrid models). The accuracy of all four models was measured using two validation sets, and point estimates. As a benchmark, the BN-based predictions were also compared to predictions obtained using Manual StepWise Regression (MSWR), and Case-Based Reasoning (CBR). The BN model generated using Hugin presented similar accuracy to CBR and Mean effort-based predictions. Our results suggest that Hybrid BN models can provide significantly superior prediction accuracy. However, good results also seem to depend on characteristics of the training and validation sets used.

1. Introduction

Web development currently represents a market that increases at an average rate of 20% per year, with Web e-commerce sales alone surpassing 95 billion USD in 2004 (three times the revenue from the world's aerospace industry)[1][40]. However, in contrast, evidence shows that most Web development projects suffer from unrealistic project schedules, leading to applications that are rarely developed on time and within budget [40]. One of the foundations of a successful Web project management is sound effort estimation, the process by which effort is predicted and used to determine costs and allocate resources effectively, enabling projects to be delivered on time and within budget.

Effort estimation is a complex domain where decisions are non-deterministic with an inherently uncertain nature.

To understand Web effort estimation, previous studies have developed models that typically use size of a Web application, and cost drivers (e.g. tools, developer's quality, team size) as input factors, and provide effort estimates as output. The differences between these studies were the number and type of size measures used, choice of cost drivers and occasionally the techniques employed to build effort estimation models. Despite numerous previous studies, only recently did Mendes [25] investigate the inclusion of uncertainty inherent to effort estimation into a model for Web effort estimation. Results showed the effort estimates obtained using an uncertainty-based model were sound and significantly superior to predictions based on two benchmark models, using the mean and median effort respectively. However, despite encouraging results, there were other compelling issues that warranted further investigation, such as:

i) Would an uncertainty-based model consistently provide superior prediction if more than one validation set was used?

ii) Would an uncertainty-based model provide superior prediction when compared to other techniques such as regression analysis and case-based reasoning?

iii)Would an uncertainty-based hybrid model provide better predictions than a data-driven only uncertainty model?

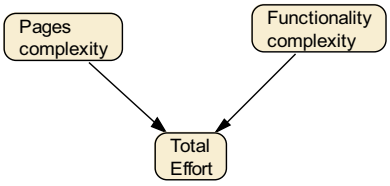
The motivation therefore and the contribution of this paper is to extend Mendes' work and investigate the use of cross-company data-driven and hybrid uncertainty-based models for early effort estimation of Web projects. As in [25], the uncertainty-based models were built using Bayesian Networks (BNs). A BN is a model that supports reasoning with uncertainty due to the way in which it incorporates existing knowledge of

[1]http://www.aia-erospace.org/stats/aero_stats/stat08.pdf
http://www.tchidagraphics.com/website_ecommerce.htm

978-0-7695-3261-5/08 $25.00 © 2008 IEEE
DOI 10.1109/ICWE.2008.16

a complex domain [15][38]. Existing knowledge is represented using two parts. The first, the qualitative part, represents the structure of a BN as depicted by a directed acyclic graph (digraph) (see Figure 1). The digraph's nodes represent the relevant variables (factors) in the domain being modelled, which can be of different types (e.g. observable or latent, categorical). The digraph's arcs represent the causal relationships between variables, where relationships are quantified probabilistically [15][37][46]. The second, the quantitative part, associates a node probability table (NPT) to each node, its probability distribution. A parent node's NPT describes the relative probability of each state (value) (Figure 1, nodes 'Pages complexity' and 'Functionality complexity'); a child node's NPT describes the relative probability of each state conditional on every combination of states of its parents (Figure 1, node 'Total Effort'). So, for example, the relative probability of 'Total Effort' being 'Low' conditional on 'Pages complexity' and 'Functionality complexity' being both 'Low' is 0.7.

Each row in a NPT represents a conditional probability distribution and therefore its values sum up to 1 [15].



Pages complexity		
Low	*Medium*	*High*
0.2	0.3	0.5

Functionality complexity	
Low	*High*
0.1	0.9

Total Effort (Low, Medium, High)				
Pages complexity	**Functionality complexity**	*Low*	*Medium*	*High*
Low	*Low*	0.7	0.2	0.1
Low	*High*	0.2	0.6	0.2
Medium	*Low*	0.1	0.7	0.2
Medium	*High*	0	0.5	0.5
High	*Low*	0.2	0.6	0.2
High	*High*	0	0.1	0.9

Figure 1 - A small BN model and three NPTs

Once a BN is specified, evidence (e.g. values) can be entered into any node, and probabilities for the remaining nodes automatically calculated using Bayes' rule [38][46]. Therefore BNs can be used for different types of reasoning, such as predictive, diagnostic, and "what-if" analyses to investigate the impact that changes on some nodes have on others [37][11][44].

The BNs detailed in this paper focus on early Web effort estimation. We had the opportunity to gather data on 195 industrial Web projects as part of the Tukutuku[2] Benchmarking project [31], and use this data to create the BNs presented herein. The project data characterises Web projects using size measures and cost drivers targeted at early effort estimation. Since we had a dataset of real industrial Web projects, we were also able to compare the accuracy of the Web effort BNs to that using Manual StepWise Regression (MSWR) [16] and Case-Based Reasoning (CBR), which are used here as a benchmark due to their frequent use in Web & software effort estimation studies. For this we computed point forecasts for the BNs using the method described in [39], and used in [25], to be detailed later.

Prediction accuracy was measured using the Mean Magnitude of Relative Error (MMRE) [7], the Median Magnitude of Relative Error (MdMRE) [7], Prediction at level l, where l = 25 (Pred(25)) [7], the Mean Magnitude of Relative Error relative to the Estimate (MEMRE) [17], the Median Magnitude of Relative Error relative to the Estimate (MdEMRE) [17], boxplots of absolute residuals (actual effort – estimated effort) and finally boxplots of z (estimated effort ÷ actual effort).

This paper extends the work presented in [25], where a hybrid Web effort BN model was built and validated using data on Web projects from the Tukutuku database and input from a Domain expert, and had its prediction accuracy compared with the mean- and median-based effort models. The main differences between this study (S2) and Mendes' [25] (S1) are as follows:

- S1 used data on 150 Web projects from the Tukutuku database; S2 used data on 195 Web projects as data on another 45 projects were volunteered since S1 was published.
- S1 used the entire Tukutuku database of 150 projects to elicit the initial BN structure, later validated by a Domain Expert (DE) and modified further using the technique proposed in [39]. After its validation, a subset of 120 randomly selected projects (training set) from the Tukutuku database was used for parameter learning. Therefore S1 in effect used a hybrid BN model, where the structure was expert-driven and its probabilities data-driven. Their BN model was validated using the remaining 30 projects (validation set). In contrast, S2 used four models: two models were automatically obtained from data (both structure elicitation and parameter learning) using one BN tool, and two training sets each containing 130 projects randomly selected from the Tukutuku database; another two models were hybrid, using structures

[2] Tukutuku means Web in Maori, the native language of New Zealand

elicited by a DE and probabilities obtained by automatically fitting the BN structure to the same training sets and tool mentioned above. Here probabilities were not validated by a DE due to the large volume of values that would need to be re-checked. Rather, each of the models was validated using a 65-project validation set.

- The DE who participated in S2 was not the same person who previously participated in S1. This happened because S1's DE was unable to participate in S2; however, S2's DE was also an experienced director of a successful Web company.
- As a benchmark, S1 used the mean- and median-based effort models. S2 employed MSWR and CBR. Two separate MSWR-based models and CBR case bases were used, each using one of the two training sets of 130 projects.

The remainder of the paper is organised as follows: Section 2 provides a literature review of Web effort estimation studies, followed by the description in Section 3 of the procedure used to build and validate the Early Web effort BN models. Sections 4 and 5 present the results using manual stepwise regression and case-based reasoning, respectively. The prediction accuracy of all techniques employed is compared in Section 6, and discussed in Section 7. Threats to the validity of the results are presented in Section 8, and finally conclusions and comments on future work are given in Section 9.

2. Literature Review

There have been numerous attempts to model effort estimation for Web projects. However, except for [25], none have used a probabilistic model beyond the use of a single probability distribution. Table 1 presents a summary of previous studies. Whenever two or more studies compare different effort estimation techniques using the same dataset, we only include the study that uses the greatest number of effort estimation techniques.

Mendes and Counsell [27] were the first to empirically investigate Web effort prediction. They estimated effort using machine-learning techniques with data from student-based Web projects, and size measures harvested late in the project's life cycle. Mendes and collaborators also carried out a series of consecutive studies [13],[26]-[36] building models using multivariate regression and machine-learning techniques using data on student-based and industrial Web projects. Recently Mendes [25] investigated the use of Bayesian Networks for Web effort estimation, using data on industrial Web projects from the Tukutuku database.

Other researchers have also investigated effort estimation for Web projects: Reifer [40],[41] proposed an extension of the COCOMO model, and a single size measure harvested late in the project's life cycle. None were validated empirically. This size measure was later used by Ruhe et al. [42], who further extended a software engineering hybrid estimation technique, named CoBRA© [5], to Web projects, using a small data set of industrial projects, mixing expert judgement and multivariate regression. Later, Baresi et al. [2],[3], and Mangia et al. [24] investigated effort estimation models and size measures for Web projects based on a specific Web development method, namely the W2000. Finally, Costagliola et al. [8] compared two sets of existing Web-based size measures for effort estimation.

Table 1 shows that most Web effort estimation studies to date used data on student-based projects; estimates obtained by applying Stepwise regression or Case-based reasoning techniques; accuracy measured using MMRE, followed by MdMRE and Pred(25).

Table 1 - Summary Literature Review

Study	Type	# datasets - (# datapoints)	Subjects	Size Measures	Prediction techniques	Best technique(s)	Measure Prediction Accuracy
1st [27]	Case study	2 - (29 and 41)	2nd year Computer Science students	Page Count, Reused Page Count, Connectivity, Compactness, Stratum, Structure	Case based reasoning, Linear regression, Stepwise regression	Case based reasoning for high experience group	MMRE
2nd [41]	Not detailed	1 - (46)	professionals	Web objects	WEBMO (parameters generated using linear regression)	-	Pred(n)
3rd [29]	Case study	1 - (37)	Honours and postgraduate Computer Science students	Length size, Reusability, Complexity, Size	Linear regression Stepwise regression	Linear Regression	MMRE
4th [13]	Case study	1 - (37)	Honours and postgraduate Computer Science	Structure metrics, Complexity metrics, Reuse metrics,	Generalised Linear Model	-	Goodness of fit

			students	Size metrics			
5th [30]	Case study	1 - (25)	Honours and postgraduate Computer Science students	Requirements and Design measures, Application measures	Case-based reasoning		MMRE, MdMRE, Pred(25), Boxplots of absolute residuals
6th [35]	Case study	1 - (37)	Honours and postgraduate Computer Science students	Page Count, Media Count, Program Count, Reused Media Count, Reused Program Count, Connectivity Density, Total Page Complexity	Case-based reasoning, Linear regression, Stepwise regression, Classification and Regression Trees	Linear/ stepwise regression or case-based reasoning (depends on the measure of accuracy employed)	MMRE, MdMRE, Pred(25), Boxplots of absolute residuals
7th [42]	Case study	1 - (12)	professionals	Web Objects	COBRA, Expert opinion, Linear regression	COBRA	MMRE, Pred(25), Boxplots of absolute residuals
8th [32]	Case study	2 - (37 and 25)	Honours and postgraduate CS students	Page Count, Media Count, Program Count, Reused Media Count (only one dataset), Reused Program Count (only one dataset), Connectivity Density, Total Page Complexity	Case-based reasoning	-	MMRE, Pred(25), Boxplots of absolute residuals
9th [3]	Formal experi ment	1 - (30)	Computer Science students	Information, Navigation and Presentation model measures	Ordinary least squares regression	-	-
10th [24]	Not detailed	unknown	unknown	Functional, Navigational Structures, Publishing and Multimedia sizing measures	An exponential model named Metrics Model for Web Applications (MMWA)	-	-
11th [8]	Case study	1 – (15)	professionals	Web pages, New Web pages, Multimedia elements, New multimedia elements, Client side Scripts and Applications, Server side Scripts and Applications, All the elements that are part of the Web Objects size measure	Linear regression, Stepwise regression, Case-based reasoning, Classification and Regression Trees	All techniques provided similar prediction accuracy	MMRE, MdMRE, Pred(25), Boxplots of residuals, boxplots of z
12th [25]	Case study	1 – (150)	professionals	Total Web pages, New Web pages, Total Images, New Images, Features off-the-shelf (Fots), High & Low effort Fots-Adapted, High & Low effort New Features, Total High & Low Effort Features	Bayesian Networks, Stepwise Regression	Bayesian Networks provided superior predictions	MMRE, MdMRE, MEMRE, MdEMRE, Pred(25), Boxplots of residuals, boxplots of z

3. Building the Web Effort BN Models

3.1. Introduction

The analysis presented in this paper was based on data from 195 Web projects in the Tukutuku database, part of the Tukutuku Benchmarking project [31], which aims to collect data from completed Web projects, to develop early Web effort estimation models and benchmark productivity across and within Web Companies. The Tukutuku database includes data on Web applications [6], which represent software applications that depend on the Web or use the Web's infrastructure for execution and are characterized by functionality affecting the state of the underlying business logic. Web applications usually include tools suited to handle persistent data, such as local file system, (remote) databases, or Web Services. Typical developers are Computer Science or Software Engineering professionals [40].

The Tukutuku database has data on 195 projects where:

- Projects come mostly from 10 different countries, mainly New Zealand (47%), Italy (17%), Spain (16%), Brazil (10%), United States (4%), England (2%), and Canada (2%).
- Project types are new developments (65.6%) or enhancement projects (34.4%).
- The languages used are mainly HTML (81%), Javascript (DHTML/DOM) (62.1%), PHP (42.6%), Various Graphics Tools (31.8%), ASP (VBScript, .Net) (13.8%), SQL (13.8%), Perl (11.8%), J2EE (9.2%), and Other (9.2%).

Each Web project in the database is characterized by 22 variables, related to a Web application and its development process (see Table 2). These size

measures and cost drivers were obtained from the results of a survey investigation [31], using data from 133 on-line Web forms that provided quotes on Web development projects. They were also confirmed by an established Web company and a second survey involving 33 Web companies in New Zealand. Consequently, it is our belief that the 22 variables identified are suitable for early Web effort estimation, and are meaningful to Web companies.

Within the context of the Tukutuku project, a new high-effort feature/function requires at least 15 hours to be developed by one experienced developer, and a high-effort adapted feature/function requires at least 4 hours to be adapted by one experienced developer. These values are based on collected data.

Table 2 - Variables for the Tukutuku database

Variable Name	Description
WEB PROJECT DATA	
TypeProj	Type of project (new or enhancement).
nLang	Number of different development languages used
DocProc	If project followed defined and documented process.
ProImpr	If project team involved in a process improvement programme.
Metrics	If project team part of a software metrics programme.
DevTeam	Size of a project's development team.
TeamExp	Average team experience with the development language(s) employed.
TotEff	Actual total effort in person hours used to develop a Web application.
EstEff	Estimated total effort in person hours to develop a Web application.
Accuracy	Procedure used to record effort data.
WEB APPLICATION	
TypeApp	Type of Web application developed.
TotWP	Total number of Web pages (new and reused).
NewWP	Total number of new Web pages.
TotImg	Total number of images (new and reused).
NewImg	Total number of new images created.
Fots	Number of features reused without any adaptation.
HFotsA	Number of reused high-effort features/functions adapted.
Hnew	Number of new high-effort features/functions.
TotHigh	Total number of high-effort features/functions
FotsA	Number of reused low-effort features adapted.
New	Number of new low-effort features/functions.
TotNHigh	Total number of low-effort features/functions

Summary statistics for the numerical variables are given in Table 3, and Table 4 summarises the number and percentages of projects for the categorical variables. As for data quality, in order to identify effort guesstimates from more accurate effort data, we asked companies how their effort data was collected (see Table 5). At least for 93.8% of Web projects in the Tukutuku database, effort values were based on more than just guesstimates.

Table 3 - Summary Statistics for numerical variables

Variable	Mean	Median	Std. Dev.	Min.	Max.
nlang	3.9	4	1.4	1.0	8
DevTeam	2.6	2	2.4	1.0	23
TeamExp	3.8	4	2.0	1.0	10
TotEff	468.1	88	938.5	1.1	5,000
TotWP	69.5	26	185.7	1.0	2,000
NewWP	49.5	10	179.1	0.0	1,980
TotImg	98.6	40	218.4	0.0	1,820
NewImg	38.3	1	125.5	0.0	1,000
Fots	3.2	1	6.2	0.0	63
HFotsA	12.0	0	59.9	0.0	611
Hnew	2.1	0	4.7	0.0	27
totHigh	14.0	1	59.6	0.0	611
FotsA	2.2	0	4.5	0.0	38
New	4.2	1	9.7	0.0	99
totNHigh	6.5	4	13.2	0.0	137

Table 4 - Summary for categorical variables

Variable	Level	# Projects	% Projects
TypeProj	New	128	65.6
	Enhancement	67	34.4
ProImpr	No	104	53.3
	Yes	91	46.7
DocProc	Yes	105	53.8
	No	90	46.2
Metrics	No	130	66.7
	Yes	65	33.3

Table 5 - How effort data was collected

Data Collection Method	# Projs	% Projs
Hours worked per project task per day	81	41.5
Hours worked per project per day/week	40	20.5
Total hours worked each day or week	62	31.8
No timesheets (guesstimates)	12	6.2

3.2. Procedure used to build the early Web effort BN models

The BNs presented in this paper were built and validated using an adapted Knowledge Engineering of Bayesian Networks (KEBN) process [9][23][46] (see Figure 2). In Figure 2, arrows represent flows through the different tasks, which are depicted by rectangles. Such tasks are executed either by people – the Knowledge Engineer (KE) and the Domain Experts (DEs) [46] (light colour rectangles), or automatic algorithms (dark grey rectangles). Dark grey cornered rectangles represent tasks that can be carried out either automatically, manually, or using a combination of both. Within the context of this research project, the author is the knowledge engineer, and an experienced director from a Web company in Auckland (New Zealand) is the DE.

The three main steps part of the KEBN process are the *Structural Development*, *Parameter Estimation*, and

Model Validation. The KEBN process iterates over these steps until a complete BN is built and validated. Below we provide a brief description of the process; readers interested in a detailed description please refer to [25].

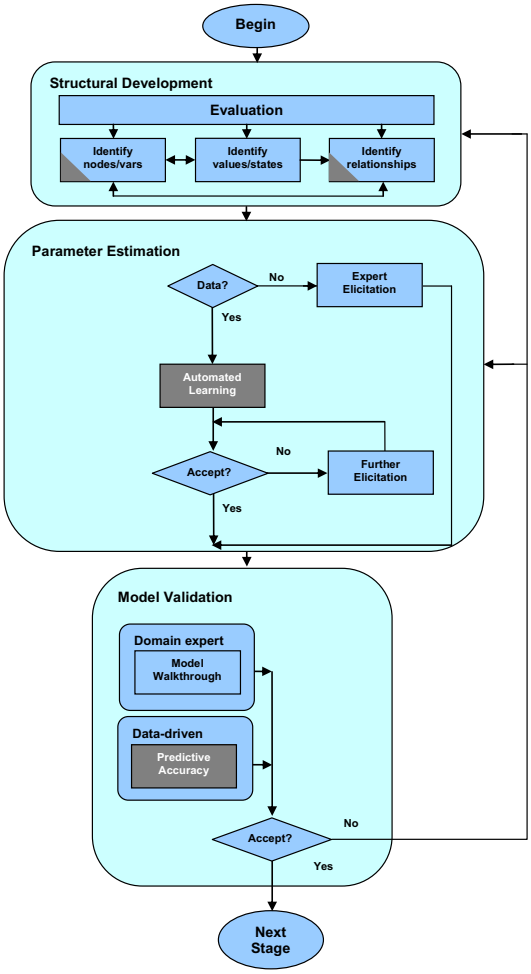


Figure 2 - KEBN, adapted from [46]

Structural Development: Entails the creation of the BN's graphical structure comprised of nodes (variables) and causal relationships. These can be identified by DEs, directly from data, or using a combination of both. Within the context of this work the BNs' structures were obtained using data from the Tukutuku database and current knowledge from a DE who is the director of a well-established Web company in Auckland (New Zealand). This DE has been a software developer and project manager for more than 25 years, and the director of a Web company for at least 7 years.

The identification of values for each of the nodes and corresponding causal relationships was initially obtained automatically using a BN tool, Hugin, and two training sets each containing 130 projects randomly chosen, leading to two of the BN models used herein. Later, another two BN models were created, all using a single model structure elicited by a DE, and probabilities obtained by automatically fitting this structure to the same two training sets and tool previously used. The variables used in all BN models were the ones available in the Tukutuku database. Hugin was chosen because it was also the tool used in [25]. All the Tukutuku database's continuous variables were discretised by converting them into multinomial variables [20], to be used with Hugin. There are no strict rules as to how many discrete approximations should be used. Some studies have employed three [39], others five [12], and others eight [44]. We chose five because the DE who participated in this study was happy with this choice, and also because anecdotal evidence from eliciting BNs with local Web companies in New Zealand has shown that companies find three to five categories sufficient. Hugin offers several discretisation algorithms. We employed the equal-frequency intervals algorithm, as suggested in [19] and used in [25], and five intervals, as also done in [25]. Therefore, each interval contained approximately 195/5 data points. Sometimes a variable presented repeated values making it impossible to have exactly the same number of data points per interval. This was the case for variables *Fots*, *HFotsA*, *Hnew*, *totHigh*, *FotsA* and *New*. None of the four BN structures were optimised [15],[10],[38] (a technique used to reduce the number of probabilities that need to be assessed for the BN model) to guarantee that every BN node would have its NPT generated solely using the Tukutuku data.

Parameter Estimation: Represents the quantitative component of a BN, i.e., conditional probabilities that quantify the causal relationships between variables [15][20]. Probabilities can be obtained via Expert Elicitation, automatically, or using a combination of both. For all the four BN models presented in this paper, probabilities were obtained by automatically fitting a BN structure to a training set of 130 Web projects (Automated learning) using a learning algorithm. Here this algorithm was the EM-Learning algorithm [21], provided in Hugin.

Model Validation: This step validates the BN constructed from the two previous steps, and determines the necessity to re-visit any of those steps. Two different validation methods are generally used - Model Walkthrough and Predictive Accuracy. Both verify if predictions provided by a BN are on average, better than those currently obtained by a DE. Model Walkthrough represents the use of real case scenarios by a DE to assess if the predictions provided by a BN correspond to the predictions (s)he would have chosen

based on his/her own expertise. Success is measured by the frequency with which the BN's predicted value with the highest probability for a target variable (e.g. total effort) corresponds to the DE's own assessment. Predictive Accuracy is normally carried out using quantitative data, and was the validation approach employed by this paper. Two validation sets, each containing 65 projects, were employed for the Model Validation step to assess the effort prediction accuracy of each BN model. Since there is no de facto standard of how many projects a validation set should contain, we chose to use a 66:33 split, as in [4][33]. The estimated effort for each of the 65 projects in each of the two validation sets was obtained using a point forecast, computed using the method described in [39]. This method computes estimated effort as the sum of the probability (ρ) of a given effort scale point multiplied by its related mean effort (μ), after normalising the probabilities such that their sum equals one. Therefore, assuming that Estimated Effort is measured using a 3-point scale (Low to High), we have:

$$Estim(Effort) = \rho_{Low}\mu_{Low} + \rho_{Medium}\mu_{Medium} + \rho_{High}\mu_{High} \quad (1)$$

This method was chosen because it had already been used within the context of software effort estimation [39] and also for early Web effort estimation [25].

3.3. The early Web effort BN structures

The structures of the three BN structures used in this paper are presented in Figure 3 (arrows point to TotalEffort, the variable to be estimated by each BN model). Note that two BN models used the same structure, elicited by a DE (see Figure 3(c)), so only three BN structures are shown in Figure 3. The BN structures (a) and (b) were automatically fit to each of the two training sets, using the Necessary Path Condition (NPC) algorithm [45], implemented in the Hugin tool. The BN structure (c) was completely elicited by the DE who participated in this study.

Table 6 shows that *TotWP*, *TotImg*, *NewWP*, and *Fots* were the only four variables chosen by at least two of the BN structures to have a direct causal effect upon *TotalEffort*.

These results corroborate previous work where number of Web pages and features/functions were found to be good predictors of total effort [18][26][33].

In this paper the predictions obtained using the four different Early Web Effort BN Models were benchmarked against those obtained using MSWR and CBR. We chose MSWR and CBR because these are the two techniques frequently used in the Web effort estimation literature. The next two Sections describe the use of MSWR and CBR, and Section 6 presents the comparison amongst the three effort estimation techniques used in this study.

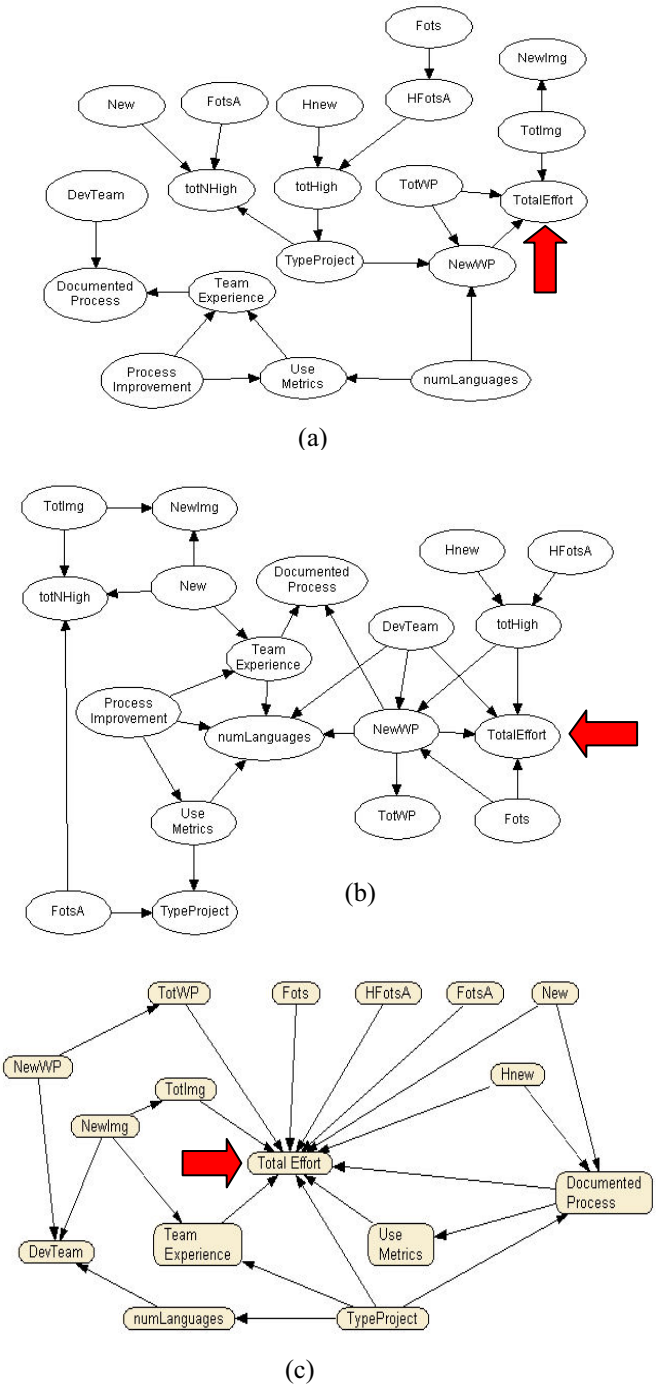


Figure 3 - Early Web Effort Estimation BN Structures

Table 6 - Variables pointing directly at *TotalEffort*

	BN structures		
Variables pointing to *TotalEffort*	**(a)**	**(b)**	**(e)**
DevTeam		✓	
Documented Process			✓
Fots		✓	✓
FotsA			✓
HFotsA			✓
Hnew			✓
New			✓
NewImg			
NewWP	✓	✓	
NumLanguages			
TeamExp			✓
TotImg	✓		✓
TotWP	✓		✓
TypeProj			✓
totHigh		✓	
Use Metrics			✓

4. Building the Regression-based Early Web Effort Model

We used the Manual StepWise Regression procedure (MSWR) proposed by Kitchenham [16] to build two regression-based models to be used as benchmark. This procedure uses residuals (*actual – estimated effort*) to select the categorical and numerical variables that jointly have a statistically significant effect on the dependent variable, *TotEffort*. Once the most important variables are selected, we then employ a multivariate regression procedure to build the final model (Equation) [16].

Each regression-based model was built using data on 130 projects from the Tukutuku dataset, using the same two training sets employed when building the BN models. Each regression model was then applied to a validation set containing data on 65 projects, and prediction accuracy measures were gathered. Before building each of the two regression-based models we ensured that variables that had more than 40% of their values missing, or zero, were excluded [14][22], such that the residuals would be homoscedastic (one of the assumptions required by any regression-based technique). After applying this exclusion criterion to both validation sets, the original set of 19 variables was reduced to 13, and the following variables were excluded from further analysis: *Fots*, *HFotsA*, *Hnew*, *totHigh*, *FotsA* and *New*. In addition, whenever numerical variables were highly skewed, they were transformed before being used in the manual stepwise procedure. This was done in order to comply with the assumptions underlying stepwise regression [16].

Boxplots, Histograms and the Shapiro-Wilk normality test confirmed that none of the numerical variables were normally distributed, and so they were transformed. The transformation employed was to take the natural log (ln), which makes larger values smaller and brings the data values closer to each other.

We created four dummy variables, one for each of the categorical variables *TypeProj*, *DocProc*, *ProImpr*, and *Metrics*.

To verify the stability of the effort model the following steps were used [18]:

i) Use of a residual plot showing residuals vs. fitted values to investigate if the residuals were random and normally distributed.

ii) Calculate Cook's distance values for all projects to identify influential data points. Those with distances greater than 4/130 were temporarily removed to test the model's stability. If the selected variables remained unchanged, the model coefficients remained stable and the goodness of fit improved, the influential projects were retained.

The first regression-based Web effort model (MSWR-1) selected four significant independent variables: *LTotWP*, *Lnlang*, *MetricsY*, and *LDevTeam*. Its adjusted R^2 was 0.711, so these four variables explained 71.1% of the variation in *TLotEff*. The residual plot showed that 13 projects seemed to have very large residuals, also confirmed using Cook's distance. To check the model's stability, a new model was generated without these 13 projects, giving an adjusted R^2 of 0.833. In the new model the independent variables remained significant but the coefficients presented different values to those in the original model. Therefore, these 13 high influence data points were permanently removed from further analysis. The MSWR-1 model is described in Table 7.

Table 7 - MSWR-1 Web effort Model

	Unstandardised Coefficients			
	B	Std. Error	t	Sig.
(Constant)	0.548	0.322	1.702	0.091
LTotWP	0.786	0.065	12.036	0.000
Lnlang	0.987	0.191	5.169	0.000
MetricsY	-1.458	0.179	-8.156	0.000
LDevTeam	0.940	0.134	7.008	0.000

The final equation, transformed back to the raw data scale, is the following:

$$TotEff = 1.729 TotWP^{0.786} nlang^{0.987} e^{-1.458 MetricsY} DevTeam^{0.940} \quad (2)$$

The residual P-P plots for the MSWR-1 Web effort both show that the residuals are normally distributed; however they were omitted due to lack of space.

The second regression-based Web effort model (MSWR-2) selected five significant independent variables: *LTotWP*, *Lnlang*, *LDevTeam*, *TypeNew*, and *ProImprY*. Its adjusted R^2 was 0.687, so these five

variables explained 68.7% of the variation in *TLotEff*. The residual plot showed that nine projects seemed to have very large residuals, also confirmed using Cook's distance. To check the model's stability, a new model was generated without these nine projects, giving an adjusted R^2 of 0.773. In the new model the independent variables remained significant but the coefficients presented different values to those in the original model. Therefore, these nine high influence data points were also permanently removed from further analysis. The MSWR-2 model is described in Table 8.

Table 8 - MSWR-2 Web effort Model

	Unstandardised Coefficients			
	B	Std. Error	t	Sig.
(Constant)	-0.090	0.420	-0.213	0.832
LTotWP	0.848	0.072	11.820	0.000
Lnlang	1.422	0.218	6.531	0.000
LDevTeam	0.840	0.146	5.753	0.000
TypeNew	-0.825	0.193	-4.280	0.000
ProImprY	-0.425	0.173	-2.460	0.015

The final Equation, transformed back to the raw data scale, is the following:

$$LTotEff = 0.4065 TotWP^{0.848} nlang^{1.422} DevTeam^{0.840} e^{-0.825 TypeNew} e^{-0.425 \mathrm{Pr}o\mathrm{Im}prY} \quad (3)$$

The residual and P-P plots for the MSWR-2 Web effort model both show that the residuals are normally distributed; however they were omitted due to lack of space.

5. Building the Case-Based Reasoning Predictions

Case-based Reasoning (CBR) is a branch of Artificial Intelligence where knowledge of similar past cases is used to solve new cases [43]. It provides effort estimates for new projects by comparing the characteristics of the current project to be estimated against a library of historical data from completed projects with a known effort (case base) [1].

It is important to note that when using CBR there are several parameters that need to be decided upon. However, existing literature on the use of CBR for Web or software effort estimation has not yet provided a consensus on what should be the best combination of parameters to provide the best effort predictions. Therefore the choice of parameters will depend on which combination works best based on the available data being used. In addition, some parameters may not be available in the CBR tool being used.

We used a commercial CBR tool - CBR-Works from tec:inno, to obtain effort estimates and the choice of parameters used in this study was motivated by previous studies that applied CBR for Web effort estimation [8],[27],[28],[29],[32]-[36], and to some extent, on the CBR tool employed:

- The similarity measure chosen was the Euclidean distance.
- The number of closest cases was of 1, 2 and 3. These correspond respectively to effort estimates obtained using the effort for the most similar project in the case base (CBR-1), the average effort of the two most similar projects in the case base (CBR-2) and the average effort of the three most similar cases in the case base (CBR-3).
- All the project attributes considered by the similarity function had equal influence on the selection of the most similar project(s).

Since CBR-Works does not provide a feature subset selection mechanism [43], we decided to use only those features significantly associated with *TotEff* [8],[28],[36]. Associations between numerical variables and *TotEff* were measured using a nonparametric test, the Spearman's rank correlation test; the associations between numerical and categorical variables were checked using the one-way ANOVA test. All tests were carried out using SPSS 12.0.1 and $\alpha = 0.05$. For both training sets, all attributes, except *TeamExp*, *HFotsA*, *FotsA* and *DocProc*, were significantly associated with *TotEff*.

CBR does not provide an explicit model as those obtained using techniques such as BN or MSWR. We simply loaded all 195 projects as the case base and marked the projects in the validation sets as 'unfinished', to guarantee that they would not be selected by the CBR tool when searching for the most similar projects in the case base.

6. Comparing Predictions

6.1. Introduction

To date the three measures commonly used in both Web and Software Engineering to compare different effort estimation techniques have been [7]:

- The Mean Magnitude of Relative Error (MMRE or Mean MRE).
- The Median Magnitude of Relative Error (MdMRE or Median MRE).
- The Prediction at level *l* (Pred(*l*)), which measures the percentage of estimates that are within *l*% of the actual values..

MRE is the basis for calculating MMRE and MdMRE, and defined as:

$$MRE = \frac{\left|e - \hat{e}\right|}{e} \quad (4)$$

where e represents actual effort and $\hat{e}$ estimated effort.

However, Kitchenham et al. [17] showed that MMRE and Pred(l) are respectively measures of the spread and kurtosis of z, where ($z = \frac{\hat{e}}{e}$). They suggest the use of boxplots of z and boxplots of the residuals ($e - \hat{e}$) as useful alternatives to simple summary measures since they can give a good indication of the distribution of residuals and z and can help explain summary statistics such as MMRE and Pred(25). In addition, they also suggest the use of the Magnitude of Relative Error relative to the Estimate (EMRE) as a comparative measure. The EMRE, unlike the MRE, uses the estimate as the divisor. As with the MRE, we can also calculate the mean EMRE (MEMRE) and Median EMRE (MdEMRE). Therefore, in this paper we use boxplots of residuals and of z, MMRE, MdMRE, Pred(25), MEMRE and MdEMRE to compare the three effort techniques used in this study.

6.2. Comparison of techniques

The techniques were compared using two validation sets each of 65 projects randomly chosen from the Tukutuku database. The values obtained for each validation set, and for each effort estimation technique, using six different prediction measures, are shown in Tables 9 and 10 respectively. Note that we also benchmarked the results against the Mean- and Median-based models, i.e., the mean and median effort for the training set were used as estimated effort. BNAuHu, BNHyHu, MSWR, CBR1, CBR2 and CBR3 stand for respectively BN automatically generated using Hugin, BN Hybrid model using Hugin, Manual Stepwise Regression, Case-based reasoning using one analogy, Case-based reasoning using two analogies, and Case-based reasoning using three analogies. The statistical significance of all results was checked using the nonparametric test Wilcoxon Signed Ranks test (α = 0.05).

The statistical significance tests based on absolute residuals show that predictions obtained using BNHyHu, MSWR, CBR1, CBR2 and CBR3 were significantly superior to those using the Mean effort, and only one technique – MSWR, presented accuracy significantly superior to Median-based effort predictions. CBR2, CBR3, BNAuHu and BNHyHu presented similar accuracy to Median-based predictions, and CBR1 showed significantly worse accuracy than Median-based predictions. MSWR was the only technique that outperformed all other techniques. Except for MSWR, the BNHyHu model presented either similar to or significantly better accuracy (Mean-based predictions, CBR1) than the remaining techniques. This model was obtained using the same Bayesian tool and a very similar process to that employed in [25]. The difference between the process used in this study and the one used in [25] is that Mendes optimised the BN's structure by applying automated learning to a structure that contained only variables that presented the highest correlation with total effort. We chose to keep the DE-based BN structure intact to fully reflect the DE's viewpoint, and also to reduce any likely bias caused by the further removal of variables.

Table 9 - Predictions Obtained using Validation Set 1

Accuracy	*MMRE*	*MdMRE*	*Pred(25) %*	*MEMRE*	*MdEMRE*
BNAuHu	7.65	1.67	7.69	1.07	0.76
BNHyHu	1.90	0.86	15.38	13.06	2.38
MSWR	1.50	0.64	23.08	1.36	0.64
CBR1	5.27	0.97	7.69	31.70	3.43
CBR2	5.06	0.87	10.77	3.59	0.81
CBR3	5.63	0.97	9.23	4.17	0.88
Mean Effort	30.35	3.99	15.38	1.07	0.91
Median Effort	5.02	0.93	9.23	4.43	0.94

These trends can be observed by looking at the boxplots of absolute residuals (see Figure 4).

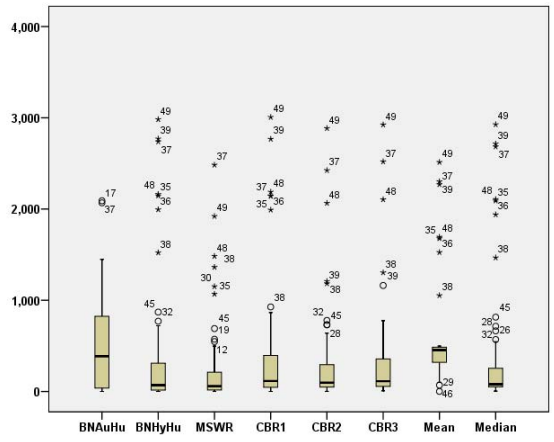


Figure 4 – Boxplots of Absolute residuals for Validation Set 1

However, the trends observed using absolute residuals differed when checking the statistical significance of results using z. Based on z (see Figure 5) the technique that significantly outperformed any other technique was BNHyHu, not MSWR. Here, MSWR did not significantly outperform CBR1 or Median-based predictions. CBR1 and BNHyHu were the only two techniques to outperform Median-based

predictions, and also are the two techniques that presented the highest MEMRE and MdEMRE.

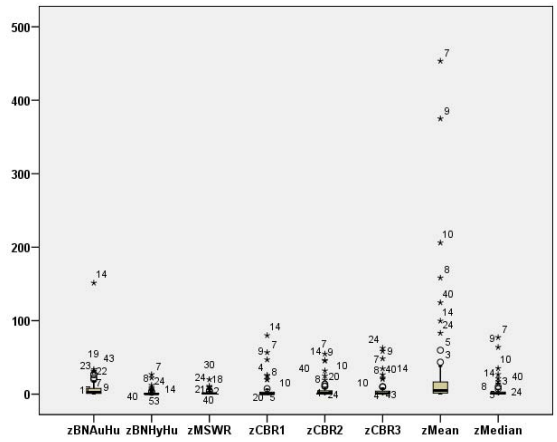


Figure 5 – Boxplots of *z* values for Validation Set 1

The statistical significance tests using Validation set 2 (see Table 10), based on absolute residuals, show that, similar to the results using Validation set 1, most techniques presented significantly superior predictions to predictions obtained using Mean effort. Here these techniques were BNAuHu, BNHyHu, MSWR, CBR1, CBR2 and CBR3. Also similar to the results obtained for Validation set 1, only MSWR presented accuracy significantly superior to Median-based predictions. Contrary to the results for Validation set 1, Median-based predictions were, except for MSWR, significantly superior to the predictions from all other techniques (including Mean-based predictions). In addition, also contrary to the results obtained using Validation set 1, and to our surprise, the best BN model was BNAuHu and not BNHyHu.

Table 10 - Predictions for Validation Set 2

Accuracy	*MMRE*	*MdMRE*	*Pred(25) %*	*MEMRE*	*MdEMRE*
BNAuHu	4.09	0.96	1.54	7.90	0.93
BNHyHu	27.95	5.31	3.08	1.34	0.90
MSWR	0.73	0.66	10.77	2.86	1.21
CBR1	4.46	0.92	7.69	21.81	0.95
CBR2	6.73	0.89	15.38	15.65	0.90
CBR3	6.09	0.84	9.23	13.26	0.89
Mean Effort	27.94	5.31	3.08	1.34	0.90
Median Effort	4.95	0.89	15.38	4.62	0.78

BNAuHu's predictions were significantly superior to those from any other BN model, and were similar to all CBR-based predictions. These trends are confirmed by the boxplots of absolute residuals (see Figure 6), which show that all distributions were highly skewed presenting a large number of outliers and extreme outliers. This is the same pattern observed when using Validation set 1. This time the results obtained using *z* were very similar to those abovementioned (see Figure 7). The differences were as follows: Not only MSWR but also CBR1 presented accuracy significantly superior to Median-based predictions. The Median-based predictions were only superior to predictions obtained using BNHyHu.

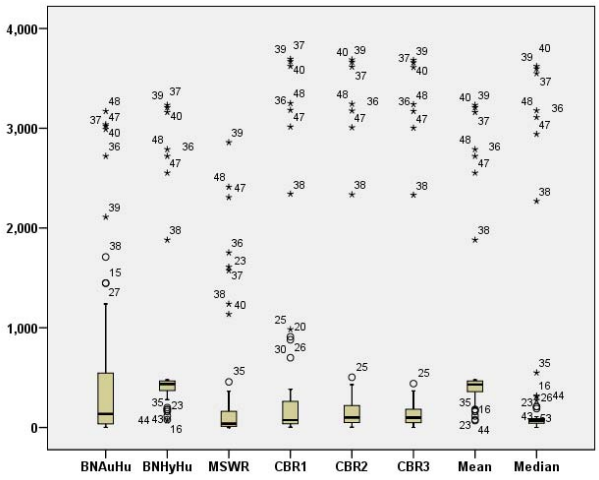


Figure 6 – Boxplots of Absolute residuals for Validation Set 2

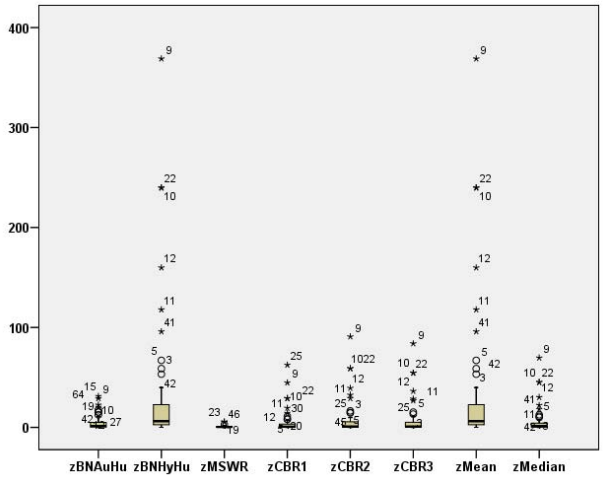


Figure 7 – Boxplots of *z* values for Validation Set 2

7. Discussion

In terms of the comparison between the tool-based BN (BNAuHu) and MSWR- & CBR-based results, trends remained fairly consistent across validation sets: MSWR presented significantly superior prediction to BNAuHu. The only difference was that CBR1 presented significantly better predictions than BNAuHu for Validation set 2 only. In relation to the comparison

between BNAuHu with Mean & Median effort models, its predictions were significantly worse than those using the Median-based model, based on Validation set 1; and its predictions were significantly better than those using the Mean-based model, based on Validation set2. So overall MSWR presented significantly superior accuracy to BNAuHu, and BNAuHu presented similar accuracy to CBR2 and CBR3, and similar to or significantly superior predictions than those obtained using the Mean-based effort model.

In terms of the comparison between the Hybrid BN (BNHyHu) and MSWR, CBR, Mean effort and Median effort, BNHyHu presented significantly superior accuracy to any other techniques when based on Validation set 1, and, except for Mean effort, presented significantly worse accuracy to any other techniques when based on Validation set 2. The results for BNHyHu using Validation set 1 were similar to those by Mendes [25] when also using a hybrid Bayesian model with a different DE-based structure and a smaller dataset from the Tukutuku database. Note that our study and that of Mendes are not independent because both shared a subset of 150 projects from the Tukutuku database.

In an attempt to understand why the results using BNHyHu differed so much across validation sets we compared the characteristics of both training and validation sets, detailed below:

- Training and Validation sets 1: Both presented the same medians for *nlang*, *DevTeam*, *TeamExp*, *NewImg*, *HFotsA*, *Hnew*, *FotsA* and *New*; validation set medians were higher than training set medians for *TotEff*, *TotWP*, *NewWP*, *Fots* and *totNHigh*; validation set medians were lower than training set medians for *TotImg* and *totHigh*.
- Training and Validation sets 2: Both presented the same medians for *nlang*, *DevTeam*, *TeamExp*, *Fots*, *HFotsA*, *Hnew* and *totHigh*; validation set medians were lower than training set medians for *TotEff*, *TotWP*, *NewWP*, *New* and *totNHigh*; validation set medians were higher than training set medians for *TotImg*, *NewImg* and *FotsA*.

Both sets presented very similar descriptive statistics; however there was one noticeable difference between them: Validation set 1 had median *TotEff* and *TotWP* higher than the median *TotEff* and *TotWP* in Training set 1, and median *TotImg* lower than the median *TotImg* in Training set 1, suggesting that projects in the validation set were slightly larger in total number of pages and effort, and smaller in total number of images than the projects in the training set. Conversely, Validation set 2 had median *TotEff* and *TotWP* lower than the median *TotEff* and *TotWP* in Training set 2, and median *TotImg* higher than the median *TotImg* in Training set 2, suggesting that projects in the validation set were slightly smaller in total number of pages and effort, and larger in total number of images than the projects in the training set. These, in addition to other variables that also differed across training/validation sets (e.g. *Fots*, *New*) may have influenced the probabilities, and therefore the results obtained. Another reason for the large differences between the two versions of BNHyHu could be have been related to the probabilities associated to *TotEff* that were elicited by the tool since Hugin did not use the same set of probabilities in both scenarios.

8. Threats to the Validity of Results

There are several factors that could have affected the validity of our results, to be detailed below:

- The dataset used in this study did not capture all relevant combinations amongst variables. However, this situation occurs whenever real industrial datasets of software or Web projects are used to build BN models.
- The choice of variable discretisation, structure learning algorithms, parameter estimation algorithms, and the number of categories used in the discretisation all affect the results and there are no clear-cut guidelines on what would be the best choice to employ. It may simply be dependent on the dataset being used and the amount of data available. Our future work includes the comparison of our results with those using variables that were discretised using a greater number of categories and a different choice of discretisation.
- Our study only used the Tukutuku variables to elicit BN's structure given that otherwise any extra nodes added to a structure would need to be elicited by a DE, and the use of more nodes would have made the comparison with MSWR and CBR impractical. Our future work includes the elicitation of other BN structures with DEs, which are not restricted to the Tukutuku variables. This will give us the opportunity to investigate the possibility of eliciting a large and unified Web effort model.
- As with any other real industrial datasets of software or Web projects, the Tukutuku dataset does not represent a random sample of projects, therefore the results presented herein are only applicable to the Web companies that volunteered data the Tukutuku project and companies that develop similar projects to those used in this study.
- This study investigated the use of data-driven BN models, which may have had a significant effect on the results. Future work includes the elicitation of BN models completely based on expert opinion, to be compared to the BN models described in this paper.

- The probabilities used by the Hybrid BN models were solely based on the automatic learning algorithm available in the BN tool used, which we believe may also have influenced the results presented herein. As part of our future work we plan to ask DEs to validate the probabilities to be used in Hybrid BN models, obtained via automatic learning.
- The Hybrid model was based on a structure elicited from only one DE, and this structure differed from the DE-based structure used in [25]. However, as part of our future work we plan to merge these two structures and use the resulting structure to obtain effort estimates, to be compared to each of the three separate structures.

9. Conclusions and Future Work

This paper presents the results of an investigation where four Bayesian Network models were built and used to estimate effort for Web projects. Two models were automatically generated using a BN tool – Hugin; another two were Hybrid BN models, built using a structure elicited by a Domain Expert, with probabilities automatically 'learnt' from the training sets. Two training and validation sets were used, each containing 130 and 65 projects respectively. The prediction accuracy of the BN models was benchmarked against predictions obtained using Manual stepwise regression and Case-based reasoning. The measures of accuracy employed were the MMRE, MdMRE, Pred(25), MEMRE, MdEMRE, absolute residuals, z, Mean and Median effort of projects in a training set. All techniques were compared using two validation sets each of 65 projects. Pairs of absolute residuals and *z* were compared using a non-parametric statistical significance test - the Wilcoxon Signed Paired Test, with $\alpha = 0.05$.

In terms of the prediction accuracy, the main results were as follows:

- MSWR presented significantly superior accuracy than the tool-based BN model;
- BNAuHu presented similar accuracy to CBR2 and CBR3, and similar to or significantly superior predictions than those using Mean effort.
- BNHyHu presented significantly superior accuracy to any other techniques when based on Validation set 1, and, except for Mean effort, presented significantly worse accuracy to any other techniques when based on Validation set 2.

The statistical significance results for BNHyHu varied when using absolute residuals or *z* values. However since the results based on *z* values also converged with those using MEMRE and MdEMRE, we chose to use *z*-based results when discussing our findings.

The BN models used in this paper were data-driven and the dataset used was small. However, even under these circumstances, the tool-based BN model (BNAuHu) presented similar accuracy to CBR and the Mean effort; one of the Hybrid BN models (BNHyHu) presented significantly superior accuracy to any other technique based on Validation set 1. We believe these are encouraging results, which corroborate the findings from [25].

As part of our future work we plan to:

- Compare the results presented in this paper with those using variables that were discretised using a greater number of categories and a different choice of discretisation.
- Elicitation of other BN structures with DEs, to investigate the possibility of eliciting a large and unified Web effort model.
- Elicitation of BN models completely based on expert opinion, to be compared to the BN models described in this paper.
- Ask DEs to validate the probabilities to be used in Hybrid BN models, obtained via automatic learning.

10. Acknowledgements

I would like to thank the domain expert and the Web companies who volunteered data to the Tukutuku project, in particular A/Prof. F. Ferrucci. I would also like to thanks Dr. Mosley[3] for his comments on a previous version of this paper. This work is sponsored by the Royal Society of New Zealand, under the Marsden Fund research grant UOA0611.

11. References

[1] L. Angelis and I Stamelos, A Simulation Tool for Efficient Analogy Based Cost Estimation, *Empirical Software Engineering*, 5, pp. 35-68, 2000.

[2] L. Baresi, S. Morasca, and P. Paolini, An empirical study on the design effort for Web applications, *Proceedings of WISE 2002*, pp. 345-354, 2002.

[3] L. Baresi, S. Morasca, and P. Paolini, Estimating the design effort for Web applications, *Proceedings of Metrics 2003*, pp. 62-72, 2003.

[4] L.C. Briand, T. Langley, and I. Wieczorek, A Replicated Assessment and Comparison of Common Software Cost Modeling Techniques, *Proceedings of ICSE 2000*, Limerick, Ireland, pp 377-386, 2000.

[5] L.C. Briand, K. El Emam, F. Bomarius, COBRA: A Hybrid Method for Software Cost Estimation, Benchmarking and Risk Assessment, *Proceedings of the 20th International Conference on Software Engineering*, 1998, pp. 390-399, 1998.

[3] http://www.metriq.biz

[6] S.P. Christodoulou, P.A. Zafiris, and T.S. Papatheodorou, WWW2000: The Developer's view and a practitioner's approach to Web Engineering, *Proc. Second ICSE Workshop on Web Engineering*, 4 and 5 June 2000, Limerick, pp. 75-92, 2000.

[7] S. Conte, H. Dunsmore, and V. Shen, *Software Engineering Metrics and Models*. Benjamin/Cummings, Menlo Park, California, 1986.

[8] G. Costagliola, S. Di Martino, F. Ferrucci, C. Gravino, G. Tortora, and G. Vitiello, Effort estimation modeling techniques: a case study for web applications, *Procs. Intl. Conference on Web Engineering* (ICWE'06), pp. 9-16, 2006.

[9] M.J. Druzdzel, A. Onisko, D. Schwartz, J.N. Dowling, and H. Wasyluk, Knowledge engineering for very large decision-analytic medical models, *Proceedings of the 1999 Annual Meeting of the American Medical Informatics Association*, pp. 1049-1054, 1999.

[10] M.J. Druzdzel, and L.C. van der Gaag, Building Probabilistic Networks: Where Do the Numbers Come From?, *IEEE Trans. on Knowledge and Data Engineering*, 12(4), 481-486, 2000.

[11] N. Fenton, P. Krause, and M. Neil, Software Measurement: Uncertainty and Causal Modeling, *IEEE Software*, 116-122, 2002.

[12] N. Fenton, W. Marsh, M. Neil, P. Cates, S. Forey, and M. Tailor, Making Resource Decisions for Software Projects, *Proc. ICSE'04*, pp. 397-406, 2004.

[13] R. Fewster, and E. Mendes, Measurement, Prediction and Risk Analysis for Web Applications, *Proceedings of IEEE Metrics Symposium*, pp. 338 – 348, 2001.

[14] R. Jeffery, M. Ruhe and I. Wieczorek, Using public domain metrics to estimate software development effort, *Proceedings Metrics'01*, London, pp. 16-27, 2001.

[15] F. V. Jensen, *An introduction to Bayesian networks*. UCL Press, London, 1996.

[16] B.A. Kitchenham, A procedure for analysing unbalanced data sets. *IEEE Trans. Software Engineering*. 24(4), pp 278-301, 1998.

[17] B.A. Kitchenham, L.M. Pickard, S.G. MacDonell, and M.J. Shepperd, What accuracy statistics really measure, *IEE Proc. - Software Engineering*, 148(3), June, 2001.

[18] B.A. Kitchenham, and E. Mendes, A Comparison of Cross-company and Single-company Effort Estimation Models for Web Applications, *Proceedings EASE 2004*, pp 47-55, 2004.

[19] A.J. Knobbe, and E.K.Y. Ho, Numbers in Multi-Relational Data Mining, *Proceedings of PKDD 2005*, Portugal, 2005.

[20] K.B. Korb, and A.E. Nicholson, *Bayesian Artificial Intelligence*, CRC Press, USA, 2004.

[21] S.L. Lauritzen, The EM algorithm for graphical association models with missing data, *Computational Statistics & Data Analysis*, 19:191-201, 1995.

[22] C. Lokan, and E. Mendes, Cross-company and Single-company Effort Models using the ISBSG Database: a Further Replicated Study, *Proceedings of ACM/IEEE ISESE*, pp. 75-84, 2006.

[23] S.M. Mahoney, and K.B. Laskey, Network Engineering for Complex Belief Networks, *Proc. Twelfth Annual Conference on Uncertainty in Artificial Intelligence*, pp. 389-396, 1996.

[24] L. Mangia, and R. Paiano, MMWA: A Software Sizing Model for Web Applications, *Proc. Fourth International Conference on Web Information Systems Engineering*, pp. 53-63, 2003.

[25] E. Mendes, The Use of Bayesian Network for Web Effort Estimation, Proceedings of ICWE'07, pp. 90-104, 2007.

[26] E. Mendes, and B.A. Kitchenham, Further Comparison of Cross-company and Within-company Effort Estimation Models for Web Applications, *Proc. IEEE Metrics*, pp. 348-357, 2004.

[27] E. Mendes, and S. Counsell, Web Development Effort Estimation using Analogy, *Proc. 2000 Australian Software Engineering Conference*, pp. 203-212, 2000.

[28] E. Mendes, and N. Mosley, Further Investigation into the Use of CBR and Stepwise Regression to Predict Development Effort for Web Hypermedia Applications, *Proc. ACM/IEEE ISESE*, Nara, Japan, pp. 79-90, 2002.

[29] E. Mendes, and N. Mosley, and S. Counsell, Web metrics - Metrics for estimating effort to design and author Web applications. *IEEE MultiMedia*, January-March, 50-57, 2001.

[30] E. Mendes, and N. Mosley, and S. Counsell, The Application of Case-based Reasoning to Early Web Project Cost Estimation, *Proceedings of IEEE COMPSAC*, pp. 393-398, 2002.

[31] E. Mendes, and N. Mosley, and S. Counsell, Comparison of Length, complexity and functionality as size measures for predicting Web design and authoring effort, *IEE Proc. Software*, 149(3), June, 86-92, 2002.

[32] E. Mendes, and N. Mosley, and S. Counsell, Do Adaptation Rules Improve Web Cost Estimation?. *Proceedings of the ACM Hypertext conference 2003*, Nottingham, UK, pp. 173-183, 2003.

[33] E. Mendes, and N. Mosley, and S. Counsell, Investigating Web Size Metrics for Early Web Cost Estimation, *Journal of Systems and Software*, 77(2), 157-172, 2005.

[34] E. Mendes, and N. Mosley, and S. Counsell, The Need for Web Engineering: an Introduction, *Web Engineering*, Springer-Verlag, Mendes, E. and Mosley, N. (Eds.) ISBN: 3-540-28196-7, pp. 1-26, 2005.

[35] E. Mendes, I. Watson, C. Triggs, N. Mosley, and S. Counsell, A Comparison of Development Effort Estimation Techniques for Web Hypermedia Applications, *Proceedings IEEE Metrics Symposium*, June, Ottawa, Canada, pp. 141-151, 2002.

[36] E. Mendes, I. Watson, C. Triggs, N. Mosley, and S. Counsell, A Comparative Study of Cost Estimation Models for Web Hypermedia Applications, *EMSE*, 8(2), 163-196, 2003.

[37] M. Neil, N. Fenton, and L. Nielsen, Building Large-scale Bayesian networks, *The knowledge Engineering Review*, KER, Vol. 15, No. 3, 257-284, 2000.

[38] J. Pearl, *Probabilistic Reasoning in Intelligent Systems*, Morgan Kaufmann, San Mateo, CA, 1988.

[39] P.C. Pendharkar, G.H. Subramanian, and J.A. Rodger, A Probabilistic Model for Predicting Software Development Effort, *IEEE Trans. Software Eng*. Vol. 31, No. 7, 615-624, 2005.

[40] D.J. Reifer, Web Development: Estimating Quick-to-Market Software, *IEEE Software*, Nov.-Dec., 57-64, 2000.

[41] D.J. Reifer, Ten deadly risks in Internet and intranet software development, *IEEE Software*, Mar-Apr, 12-14, 2002.

[42] M. Ruhe, R. Jeffery, and I. Wieczorek, Cost estimation for Web applications, *Proceedings ICSE 2003*, 285-294, 2003.

[43] M.J. Shepperd, and G. Kadoda, Using Simulation to Evaluate Prediction Techniques, *Proceedings IEEE Metrics'01*, London, UK, pp. 349-358, 2001.

[44] Stamelos, L. Angelis, P. Dimou, and E. Sakellaris, On the use of Bayesian belief networks for the prediction of software productivity, *Information and Software Technology*, Vol. 45, No. 1, 1 January 2003, 51-60(10), 2003.

[45] H. Steck, and V. Tresp, Bayesian Belief Networks for Data Mining, *Proceedings of The 2nd Workshop on Data Mining und Data Warehousing*, Sammelband, September 1999.

[46] O. Woodberry, A. Nicholson, K. Korb, and C. Pollino, Parameterising Bayesian Networks, *Proc. Australian Conference on Artificial Intelligence*, pp. 1101-1107, 2004.