Bayesian Network Models for Web Effort Prediction: A Comparative Study

Emilia Mendes and Nile Mosley

Abstract—*Objective.* The objective of this paper is to compare, using a cross-company data set, several Bayesian Network (BN) models for Web effort estimation. *Method.* Eight BNs were built; four automatically using Hugin and PowerSoft tools with two training sets, each containing data on 130 Web projects from the Tukutuku database; four using a causal graph elicited by a domain expert, with parameters obtained by automatically fitting the graph to the same training sets used in the automated elicitation (hybrid models). The accuracy of all eight models was measured using two validation sets, each containing data on 65 projects, and point estimates. As a benchmark, the BN-based estimates were also compared to estimates obtained using Manual Stepwise Regression (MSWR), Case-Based Reasoning (CBR), and mean and median-based effort models. *Results.* MSWR presented significantly better predictions than any of the BN models built herein and, in addition, was the only technique to provide significantly superior predictions to a median-based effort model. Two BN models, BNAuHu and BNHyHu, presented similar to, or significantly better accuracy than, the mean-based effort model and similar accuracy to the median-based effort model; however, both showed significantly worse accuracy than MSWR. The other two BN models showed worse accuracy than at least mean-based predictions. *Conclusions.* This paper investigated data-driven and hybrid BN models using project data from the Tukutuku database. Our results suggest that the use of simpler models, such as the median effort, can outperform more complex models, such as BNs. In addition, MSWR seemed to be the only effective technique for Web effort estimation.

Index Terms—Web cost estimation, project management, software engineering, Web engineering.

1 Introduction

A cornerstone of 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 the causal relationship among factors is nondeterministic and with an inherently uncertain nature. For example, assuming there is a relationship between development effort and developers' experience using the development environment, it is not necessarily true that higher experience will lead to a reduction in effort. However, as experience increases so does the *probability* of decreased effort.

Several studies in software engineering have proposed the use of causal models and probabilistic reasoning for software effort and resource estimation (e.g., [13], [14], [42]). However, their results cannot be readily reused for Web effort estimation [37] given that Web development differs from traditional software development [31].

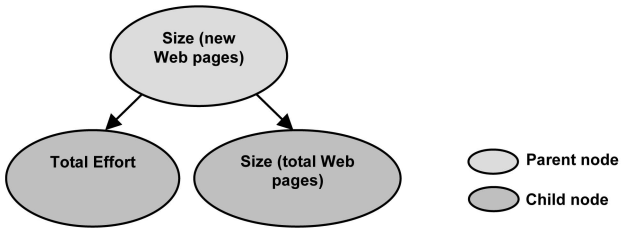
Within the context of Web effort estimation, numerous studies investigated the use of effort prediction techniques. However, to date, only Mendes [26], [27], [28] has investigated the inclusion of uncertainty, inherent to effort estimation, into a model for Web effort estimation. This model, a Hybrid Bayesian Network (BN) model, presented significantly superior predictions than the mean and median-based effort [27], multivariate regression [26], [28], Case-Based Reasoning (CBR), and Classification and Regression Tree (CART) [28].

Therefore, the goal and contribution of this paper is to compare and assess the prediction accuracy of several cross-company data-driven and hybrid BN models for Web effort estimation. In addition, to our knowledge, this is also the first time an effort estimation study has compared different data-driven and hybrid BN models and used more than one training/validation set to do so, extending the contribution of this paper beyond the scope of Web effort estimation. A detailed literature review on the use of BNs for software effort estimation is provided in [42].

A BN is a model which supports reasoning with uncertainty due to the way in which it incorporates existing complex domain knowledge [17], [41]. Herein, knowledge is represented using two parts. The first, which is the qualitative part, represents the structure of a BN as depicted by a directed acyclic graph (digraph; see Fig. 1). The digraph's nodes represent the relevant variables (factors) in the domain being modeled, 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 [17], [40], [49]. The second, which is 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); a child node's NPT describes the relative probability of each state conditional on every combination of states of its

- *E. Mendes is with the Computer Science Department, The University of Auckland, Private Bag 92019, Auckland, New Zealand. E-mail: emilia@cs.auckland.ac.nz.*
- *N. Mosley is with MetriQ Limited, PO Box 837, Waiheke 1840, Auckland, New Zealand. E-mail: nile@metriq.biz.*

Manuscript received 25 Apr. 2007; revised 16 June 2008; accepted 22 July 2008; published online 1 Aug. 2008.
Recommended for acceptance by A. Mockus.
For information on obtaining reprints of this article, please send e-mail to: tse@computer.org, and reference IEEECS Log Number TSE-2007-04-0142.
Digital Object Identifier no. 10.1109/TSE.2008.64.



NPT for node Size (new Web pages)	
Low	0.2
Medium	0.3
High	0.5

NPT for node Total Effort (TE)			
Size (new Web pages)	*Low*	*Medium*	*High*
Low	0.8	0.2	0.1
Medium	0.1	0.6	0.2
High	0.1	0.2	0.7

Fig. 1. A small BN model and two NPTs [26].

parents (e.g., the relative probability of total effort (TE) being "Low" conditional on Size (new Web pages; SNWP) being "Low" is 0.8). Each column in an NPT represents a conditional probability distribution and, therefore, its values sum up to 1 [17].

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 [41], [49]. Therefore, BNs can be used for different types of reasoning, such as predictive and "what-if" analyses, to investigate the impact that changes on some nodes have on others [40], [13], [47].

The BNs detailed in this paper focus on Web effort estimation. We had the opportunity to gather data on 195 industrial Web projects as part of the Tukutuku[1] Benchmarking project [34] and use these data to create the BNs presented herein. The project data characterize Web projects using size measures and cost drivers targeted at early effort estimation. Since we had a data set 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) [18] and CBR, which are used herein as a benchmark because of their frequent use in Web and software effort estimation studies. For this, we computed point forecasts for the BNs using the method described in [42] and used in [26], [27], [28], to be detailed later.

Prediction accuracy was measured using numerous measures described in Section 6.

This paper extends the work presented in [26], [27], [28], where a single hybrid Web effort BN model was built and validated using data on Web projects from the Tukutuku database and input from a Domain Expert (DE), and had its prediction accuracy compared with an SWR-based model [26], [28], mean and median-based effort models [27], CBR, and CART [28]. The main differences between this study (S2) and Mendes' [26], [27], [28] (S1) are given 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 a single BN tool, Hugin, for structure and parameter learning; S2 used two tools, Hugin and PowerSoft.
- S1 used the entire Tukutuku database of 150 projects to elicit the initial BN causal graph (structure), later validated by a DE, and modified further using the technique proposed in [42]. 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 causal graph was expert driven and its probabilities data driven. Their BN model was validated using the remaining 30 projects (validation set). In contrast, S2 used eight models: Four models were automatically obtained from data (both structure elicitation and parameter learning) using two BN tools and two training sets, each containing 130 projects randomly selected from the Tukutuku database; another four models were hybrid, using graphs elicited by a DE and probabilities obtained by automatically fitting the BN graph to the same training sets and tools mentioned above. Here, probabilities were not validated by a DE due to the large volume of values that would need to be rechecked. Each of the models was validated using two 65-project validation sets. S2 built eight different models and used more than one validation set in order to also investigate to what extent the type of BN model built, the BN tool used, and the number of validation sets can affect the predictions.
- The DEs who participated in S1 and S2 were not the same; however, both were experienced directors of successful Web companies in Brazil and New Zealand, respectively.
- As a benchmark, S1 built single SWR-based and CART-based models using the training set of 120 projects and also used CBR. S2 employed MSWR and CBR. Two separate MSWR-based models and CBR case bases were used, each containing one of the two training sets of 130 projects. Both used the mean and median-based effort predictions.

The remainder of this paper is organized as follows: Section 2 provides a literature review of Web effort

1. Tukutuku means Web in Maori, the native language of New Zealand.

estimation studies, followed by the description of the procedure used to build and validate the Web effort BN models in Section 3. Sections 4 and 5 present the results using MSWR and CBR, respectively. The prediction accuracy of all techniques employed is compared in Section 6 and threats to the validity of the results are discussed in Section 7. Finally, conclusions and comments on future work are given in Section 8.

2 Literature Review of Web Effort Estimation Studies

There have been numerous attempts to model effort estimation for Web projects, but, except for S1, none have used a probabilistic model beyond the use of a single probability distribution. Table 1 in the Appendix, which can be found at http://doi.ieeecomputersociety.org/10.1109/TSE.2008.64, presents a summary of previous studies. Whenever two or more studies compared different effort estimation techniques using the same data set, we only included the study that used the greatest number of effort estimation techniques.

Mendes and Counsell [30] 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 et al. also carried out a series of consecutive studies [15], [29], [30], [31], [32], [33], [34], [35], [36], [37], [38], [39], where models were built using multivariate regression and machine-learning techniques using data on student-based and industrial Web projects. Recently, Mendes [26], [27], [28] investigated the use of BNs 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 [43], [44] proposed an extension of the COCOMO model and a single size measure harvested late in the project's life cycle. This size measure was later used by Ruhe et al. [45], who further extended a software engineering hybrid estimation technique, named CoBRA [6], to Web projects, using a small data set of industrial projects, mixing expert judgment and multivariate regression. Later, Baresi et al. [2], [3] and Mangia and Paiano [25] investigated effort estimation models and size measures for Web projects based on a specific Web development method, namely, the W2000. Finally, Costagliola et al. [10] compared two sets of existing Web-based size measures for effort estimation.

In summary, most Web effort estimation studies to date used data on student-based projects; estimates obtained by applying stepwise regression or CBR techniques; accuracy measured using MMRE, followed by MdMRE and Pred(25).

3 Building the Web Effort BN Models

3.1 Introduction

The analysis presented in this paper was based on 195 Web projects data from the Tukutuku database [34], where

- projects come mostly from 10 different countries, mainly New Zealand (47 percent), Italy (17 percent), Spain (16 percent), Brazil (10 percent), United States (4 percent), England (2 percent), and Canada (2 percent),
- project types are new developments (65.6 percent) or enhancement projects (34.4 percent), and
- the languages used are mainly HTML (81 percent), Javascript (DHTML/DOM) (62.1 percent), PHP (42.6 percent), various graphics tools (31.8 percent), ASP (VBScript, .Net; 13.8 percent), SQL (13.8 percent), Perl (11.8 percent), J2EE (9.2 percent), and others (9.2 percent).

Each Web project in the database is characterized by 25 variables, related to a Web application and its development process (see Table 1). These size measures and cost drivers were obtained from the results of a survey investigation [34], using data from 133 online 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 25 variables identified are suitable for 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.

Summary statistics for the numerical variables are given in Table 2 and Table 3 summarizes the number and percentages of projects for each of 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 2 in the Appendix, which can be found at http://doi.ieeecomputersociety.org/10.1109/TSE.2008.64). At least for 93.8 percent of Web projects in the Tukutuku database, effort values were based on more than just guesstimates.

3.2 Procedure Used to Build the Early Web Effort BN Models

The BNs were built and validated using an adapted Knowledge Engineering of BN (KEBN) process [11], [24], [49] (see Fig. 2).

In Fig. 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 DEs [49] (light-colored rectangles)—or automatic algorithms (dark gray rectangles). Dark gray cornered rectangles represent tasks that can be done either automatically, manually, or using a combination of both. Within the context of this work, the first author is the KE and an experienced director from a Web company in Auckland, New Zealand, is the DE.

The three main steps that are 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. Each of these steps is briefly described.

Structural Development entails the creation of the BN's graphical structure (causal graph) containing nodes (variables) and causal relationships. These can be identified by

TABLE 1
Variables for the Tukutuku Database

	Variable Name	Description
Company Data	*Country*	Country company belongs to.
	Established	Year when company was established.
	nPeopleWD	Number of people who work on Web design and development.
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

TABLE 2
Summary Statistics for Numerical Variables

Variable	Mean	Median	Std. Dev.	Min.	Max.	Variable	Mean	Median	Std. Dev.	Min.	Max.
nlang	3.9	4	1.4	1.0	8	*Fots*	3.2	1	6.2	0.0	63
DevTeam	2.6	2	2.4	1.0	23	*HFotsA*	12.0	0	59.9	0.0	611
TeamExp	3.8	4	2.0	1.0	10	*Hnew*	2.1	0	4.7	0.0	27
TotEff	468.1	88	938.5	1.1	5000	*totHigh*	14.0	1	59.6	0.0	611
TotWP	69.5	26	185.7	1.0	2000	*FotsA*	2.2	0	4.5	0.0	38
NewWP	49.5	10	179.1	0.0	1980	*New*	4.2	1	9.7	0.0	99
TotImg	98.6	40	218.4	0.0	1820	*totNHigh*	6.5	4	13.2	0.0	137
NewImg	38.3	1	125.5	0.0	1000						

TABLE 3
Summary of Number of Projects and Percentages for Categorical Variables

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

DEs, directly from data, or using a combination of both. Within the context of this work, the BNs' graphs were obtained using data from the Tukutuku database and current knowledge from a DE, a 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 and relationships was initially obtained automatically using two BN tools, Hugin

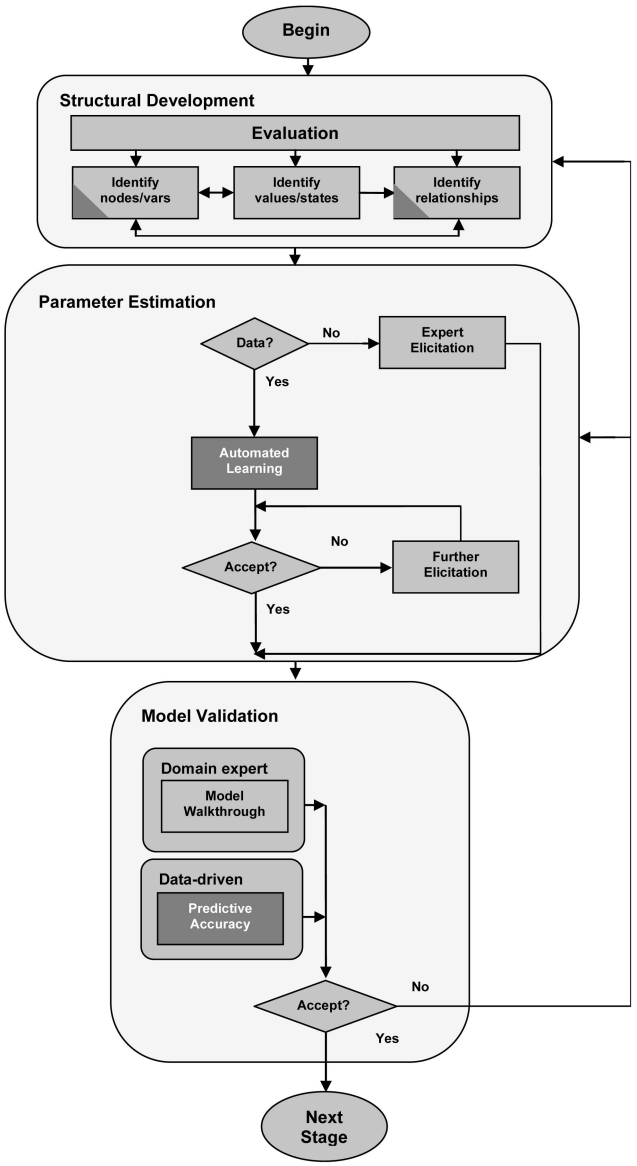


Fig. 2. KEBN, adapted from [49].

and PowerSoft, and two training sets each containing 130 projects randomly chosen, leading to four of the BN models used in this study. Later, another four BN models were created, all using a single model structure elicited by the DE and probabilities obtained by automatically fitting this structure to the same two training sets and tools previously used. The variables used in all BN models were the ones from the Tukutuku database. Hugin was chosen because it was also the tool used in [26] and PowerSoft was chosen because it implemented award-winning algorithms.[2] All continuous variables were discretized by converting them into multinomial variables [42], to be used with Hugin Expert and PowerSoft. There are no strict rules as to how many discrete approximations should be used. Some studies have employed three [42], others five [14], seven [4], and eight [47]. We chose five because the DE participating in this study was happy with this choice and also because anecdotal evidence from eliciting BNs with local Web companies has shown that companies find three to five categories sufficient. Both Hugin and PowerSoft offer several discretization algorithms. We used the equal-frequency intervals algorithm, as suggested in [21] and used in [26], [27], [28], and five intervals, as also done in [26], [27], [28]. Therefore, each interval contained approximately 130/5 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 eight BN structures was optimized [17], [12], [41] (a technique used to reduce the number of probabilities that need to be assessed for the network) to guarantee that every BN node would have its NPT generated solely using the Tukutuku data. The five effort categories used with both Hugin and PowerSoft were given as follows: [1, 1,000.88), [1,000.88, 2,000.66), [2,000.66, 3,000.44), [3,000.44, 4,000.22), [4,000.22, 5,000.11).

Parameter Estimation represents the quantitative component of a BN, which results in conditional probabilities that quantify the relationships between variables [17]. Probabilities can be obtained via Expert Elicitation, automatically, or using a combination of both. For all eight BN causal graphs in this paper, parameters were obtained by automatically fitting a BN graph to two training sets each of 130 Web projects (automated learning). Hugin used the EM-Learning algorithm [22] and PowerSoft used a proprietary algorithm [7]. Two validation sets, each containing 65 projects, were then 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 [5], [36].

Model Validation. This step validates the BN constructed from the two previous steps and determines the necessity to revisit any of those steps. Two different validation methods are generally used—Model Walkthrough and Predictive Accuracy [49]. Both verify if predictions provided by a BN are, on average, better than those currently obtained by a DE. Predictive Accuracy is normally carried out using quantitative data and was the validation approach employed by this paper. Estimated effort for each of the projects in a validation set was obtained using a point forecast, computed using the method described in [42]. 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 normalizing the probabilities such that their sum equals one. Therefore, assuming that Estimated Effort is measured using a 5-point scale (Very Low to Very High), we have

$$Estimated(Effort) = \rho_{VeryLow}\mu_{VeryLow} + \rho_{Low}\mu_{Low} + \rho_{Medium}\mu_{Medium} + \rho_{High}\mu_{High} + \rho_{VeryHigh}\mu_{VeryHigh}. \quad (1)$$

This method was chosen because it had already been used within the context of software [42] and early Web effort estimation [26], [27], [28].

Model walkthrough represents the use of real case scenarios that are prepared and used 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

2. http://www.cs.ualberta.ca/~jcheng/bnsoft.htm.

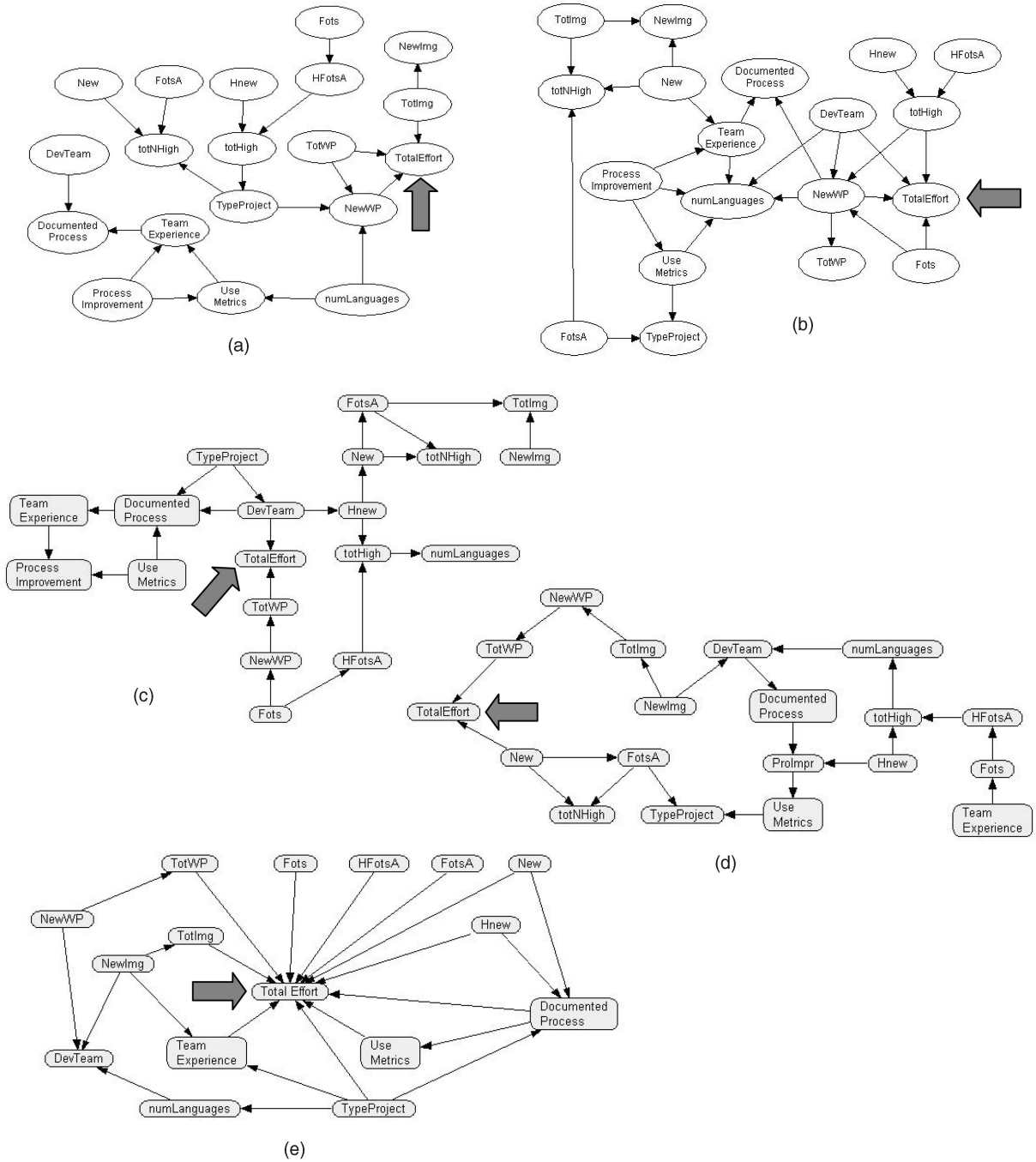


Fig. 3. Web effort estimation causal graphs.

which the BN's predicted value with the highest probability for a target variable (e.g., TE) corresponds to the DE's own assessment. We did not employ a model walkthrough to validate any of the Web effort BNs because we had already carried out a Predictive accuracy procedure using two validation sets of real data volunteered by numerous Web companies worldwide.

3.3 The Early Web Effort BNs

The causal graphs of the eight BN models are presented in Fig. 3 (arrows point to TotalEffort, the variable to be estimated by each BN model).

Note that four BN models used the same graph, elicited by a DE, so only five BN causal graphs are shown. Graphs (a)-(d) were automatically fit to each of the two training sets; (a) and (b) were fit using the Necessary Path Condition (NPC) algorithm [48] implemented in the Hugin tool; (c) and (d) were automatically fit using the algorithm B, implemented in the PowerSoft tool and detailed in [7]. Structure (e) was elicited by a DE. Table 4 shows that *TotWP* was the only variable chosen by nearly all BN structures, followed by *DevTeam*, *Fots*, *New*, *NewWP*, and *TotImg* (two models). Except for *TotImg*, the results corroborate previous studies where number of Web pages, features/functions,

TABLE 4
Variables Pointing Directly at *TotalEffort*

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

TABLE 5
Summary of Causal Relationships

			BN causal graphs								BN causal graphs				
Origin		Target	(a)	(b)	(c)	(d)	(e)	Origin		Target	(a)	(b)	(c)	(d)	(e)
Fots	→	*HFotsA*	✓	✓				*TeamExp*	→	*ProImpr*		✓			
HFotsA	→	*Fots*			✓			*ProImpr*	→	*TeamExp*	✓		✓		
NewImg	→	***TotImg***		✓		✓	✓	***ProImpr***	→	***Metrics***	✓		✓	✓	
TotImg	→	*NewImg*	✓		✓			*Metrics*	→	*ProImpr*		✓			
NewWP	→	***TotWP***		✓	✓	✓	✓	*DevTeam*	→	*nLang*			✓		
TotWP	→	*NewWP*	✓					*nLang*	→	*DevTeam*				✓	✓
TeamExp	→	*DocProc*	✓		✓			*Metrics*	→	*DocProc*		✓			
DocProc	→	*TeamExp*		✓				*DocProc*	→	*Metrics*		✓			✓
HFotsA	→	***totHigh***	✓	✓	✓	✓		*NewWP*	→	*TotEff*	✓		✓		
Hnew	→	***totHigh***	✓	✓	✓	✓		***TotWP***	→	***TotEff***	✓	✓		✓	✓
FotsA	→	***totNHigh***	✓	✓	✓	✓		*TotImg*	→	*TotEff*	✓				✓
New	→	***totNHigh***	✓	✓	✓	✓		*Metrics*	→	*TypeProj*			✓	✓	
DevTeam	→	***DocProc***	✓	✓		✓		*FotsA*	→	*TypeProj*			✓	✓	

and development team were found to be good predictors of TE [20], [29], [36].

The similarities among the five BN graphs are summarized in Table 5, where we present pairs of cause and effect relationships and where they were observed. We only show relationships identified by at least two graphs. Shaded areas show pairs of variables where the direction of the causal relationship differed from graph to graph. This situation can occur since, with the exception of graph (e), the direction of a relationship depends on the structure learning algorithm being used and two different algorithms were employed in this study. Some pairs are also in bold whenever they were common to at least three graphs.

Six of the nine causal relationships identified in at least three graphs are clear-cut because the "Target" variable includes the "Origin" variable (e.g., *TotImg* includes *NewImg*). The remaining three suggest the following: The size of the development team has a direct causal effect on the use of a documented process; being involved in a process improvement program has a direct causal effect on the use of metrics throughout a project; total number of Web pages has a direct causal effect on the total amount of effort required to develop a Web application. The shaded areas also suggest possible relationships between *Fots* and *HFotsA*, *TeamExp* and *DocProc*, *TeamExp* and *ProImpr*, *DevTeam* and *nLang*, and *Metrics* and *DocProc*; however, the direction of the causal relationship differed between causal graphs.

In this paper, the predictions obtained using the eight different 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 for Web effort estimation. Sections 4 and 5 describe the use of MSWR and CBR, and Section 6 presents the comparison among the three effort estimation techniques used herein.

4 Building the Regression-Based Web Effort Model

We used the MSWR procedure proposed by Kitchenham [18] to build two regression-based models to be used as benchmark. This procedure uses residuals $(\text{actual} - \text{estimated effort})$ to select the categorical and numerical variables that jointly have a statistically significant effect on the dependent variable, *TotEffort*. Once the most important

TABLE 6
MSWR-1 Web Effort Model

	Unstandardised Coefficients				95% Confidence Interval for B	
	B	Std. Error	t	Sig.	Lower Bound	Upper Bound
(Constant)	0.548	0.322	1.702	0.091	-0.090	1.186
LTotWP	0.786	0.065	12.036	0.000	0.657	0.915
Lnlang	0.987	0.191	5.169	0.000	0.608	1.365
MetricsY	-1.458	0.179	-8.156	0.000	-1.812	-1.104
LDevTeam	0.940	0.134	7.008	0.000	0.674	1.206

variables are selected, we then employ a multivariate regression procedure to build the final model (Equation) [18].

Each regression-based model was built 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 gathered. Before building each of the two regression-based models, we ensured that variables that had more than 40 percent of their values missing, or zero, were excluded [16], [23] such that the residuals would be homoscedastic (one of the assumptions required by any regression-based technique). After applying this exclusion criterion to both training 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 in order to comply with the assumptions underlying stepwise regression [18]. Boxplots, Histograms, and the Shapiro-Wilk normality test confirmed that none of the numerical variables was normally distributed and, so, they were transformed using the natural log transformation (ln), which makes larger values smaller and brings the data values closer to each other.

Four dummy variables were created, 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 [20]: 1) use of a residual plot showing residuals versus fitted values to investigate if the residuals were random and normally distributed; 2) use of 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, and 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 percent of the variation in *LTotEff*. 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 removed from further analysis. The final MSWR-1 model is described in Table 6.

The equation as read from the final model's output is

$$\begin{aligned} LTotEff = {} & 0.548 + 0.786 LTotWP + 0.987 Lnlang \\ & - 1.458 MetricsY + 0.940 LDevTeam, \end{aligned} \tag{2}$$

which, when transformed back to the raw data scale, gives the following equation:

$$\begin{aligned} & TotEff = \\ & 1.729 TotWP^{0.786} nlang^{0.987} e^{-1.458 MetricsY} DevTeam^{0.940}. \end{aligned} \tag{3}$$

The residual plot and the P-P plot for the MSWR-1 Web effort model are presented in Fig. 1 in the Appendix, which can be found at http://doi.ieeecomputersociety.org/10.1109/TSE.2008.64, and both suggest that the residuals are normally distributed.

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. 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 removed from further analysis. The MSWR-2 model is described in Table 7.

The equation as read from the final model's output is

$$\begin{aligned} & LTotEff = -0.090 + 0.848 LTotWP + 1.422 Lnlang \\ & + 0.840 LDevTeam - 0.825 TypeNew - 0.425 \,\mathrm{Pr}\, o\mathrm{Im}prY, \end{aligned} \tag{4}$$

which, when transformed back to the raw data scale, gives the following equation:

$$\begin{aligned} LTotEff = {} & 0.4065 TotWP^{0.848} nlang^{1.422} DevTeam^{0.840} \\ & \times e^{-0.825 TypeNew} e^{-0.425 \,\mathrm{Pr}\, o\mathrm{Im}prY}. \end{aligned} \tag{5}$$

The residual plot and the P-P plot for the MSWR-2 Web effort model are presented in Fig. 2 in the Appendix, which can be found at http://doi.ieeecomputersociety.org/10.1109/TSE.2008.64, and both suggest that the residuals are normally distributed.

TABLE 7
MSWR-2 Web Effort Model

	Unstandardised Coefficients				95% Confidence Interval for B	
	B	Std. Error	t	Sig.	Lower Bound	Upper Bound
(Constant)	-0.090	0.420	-0.213	0.832	-0.921	0.742
LTotWP	0.848	0.072	11.820	0.000	0.706	0.990
Lnlang	1.422	0.218	6.531	0.000	0.990	1.853
LDevTeam	0.840	0.146	5.753	0.000	0.551	1.129
TypeNew	-0.825	0.193	-4.280	0.000	-1.206	-0.443
ProImprY	-0.425	0.173	-2.460	0.015	-0.767	-0.083

5 Building the Case-Based Reasoning Predictions

Case-base Reasoning (CBR) is a branch of Artificial Intelligence where knowledge of similar past cases is used to solve new cases [46]. 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). It is important to note that ,when using CBR, there are several parameters that need to be decided upon. There is no 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. 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. Our choice of parameters was motivated by previous studies [10], [30], [31], [32], [35], [36], [37], [38], [39] and, to some extent, by the CBR tool employed: 1) The similarity measure chosen was the euclidean distance; 2) 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); 3) all of the project attributes considered by the similarity function had equal influence on the selection of the most similar project(s).

To simulate the feature subset selection mechanism [46] in CBR-Works, we used only features significantly associated with *TotEff* [10], [31], [39]. Associations between numerical variables and *TotEff* were measured using a nonparametric test, the Spearman's rank correlation test; associations between numerical and categorical variables were checked using the one-way ANOVA test. All tests were carried out using SPSS 15.0.1 and $\alpha = 0.05$ and, for both training sets, all attributes, except *TeamExp*, *HFotsA*, *FotsA*, and *DocProc*, were significantly associated with *TotEff*.

CBR does not provide an explicit model. 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 between Techniques

6.1 Introduction

To date, the four measures commonly used in Web/software engineering to compare different effort estimation techniques have been [9]:

- the Magnitude of Relative Error (MRE);
- the Mean MRE (MMRE);
- the Median MRE (MdMRE);
- the Prediction at level l ($\mathrm{Pred}(l)$), which measures the percentage of estimates that are within l percent of the actual values.

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

$$MRE = \frac{|\mathrm{e} - \hat{\mathrm{e}}|}{\mathrm{e}}, \tag{6}$$

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

Suggestions have been made [9] that l should be set at 25 percent and that a good prediction system should offer this accuracy level 75 percent of the time. However, in practice, it is important to also take into account the accuracy obtained using the mean and median-based effort models.

Despite MMRE, MdMRE, and Pred(25) being commonly used, Kitchenham et al. [19] showed that MMRE and $\mathrm{Pred}(l)$ are, respectively, measures of the spread and kurtosis of z ($z = \frac{\hat{e}}{e}$) and, therefore, summary statistics and boxplots of the z variable should instead be used to compare different prediction systems. However, they also added that, since the z variable presents some undesirable properties, which include asymmetry, boxplots of the residuals $(e - \hat{e})$ were a good alternative to z. Kitchenham et al. [19] also suggest the use of the MMRE and MdMRE relative to the Estimate (MEMRE and MdEMRE) as comparative measures. The EMRE, unlike the MRE, uses the estimate as the divisor and is defined as

$$EMRE = \frac{|\mathrm{e} - \hat{\mathrm{e}}|}{\hat{\mathrm{e}}}. \tag{7}$$

Therefore, in this paper, we used boxplots of residuals and of z, MMRE, MdMRE, Pred(25), MEMRE, and MdEMRE to compare the three effort techniques used in this study. The statistical significance of all results was

TABLE 8
Predictions Obtained Using Validation Set 1

Accuracy	MMRE	MdMRE	Pred(25) %	MEMRE	MdEMRE	Sum of Abs. Residuals
BNAuPo	13.97	2.57	4.62	**(1)** 0.78	**(3)** 0.81	25,378.14
BNAuHu	7.65	1.67	7.69	**(3)** 1.07	**(2)** 0.76	32,042.18
BNHyPo	36.00	4.90	7.69	**(2)** 1	0.93	38,263.84
BNHyHu	**(2)** 1.90	**(2)** 0.86	**(2)** 15.38	13.06	2.38	25,093.09
MSWR	**(1)** 1.50	**(1)** 0.64	**(1)** 23.08	1.36	**(1)** 0.64	**(1)** 16,255.82
CBR1	5.27	0.97	7.69	31.70	3.43	26,057.18
CBR2	5.06	**(3)** 0.87	**(3)** 10.77	3.59	**(3)** 0.81	**(2)** 20,233.17
CBR3	5.63	0.97	9.23	4.17	0.88	**(3)** 20,949.30
Mean Effort	30.35	3.99	**(2)** 15.38	**(3)** 1.07	0.91	34,713.00
Median Effort	**(3)** 5.02	0.93	9.23	4.43	0.94	24,457.08

TABLE 9
Predictions Obtained Using Validation Set 2

Accuracy	MMRE	MdMRE	Pred(25) %	MEMRE	MdEMRE	Sum of Abs. Residuals
BNAuPo	14.93	6.46	0	**(1)** 0.94	**(3)** 0.90	38,055.38
BNAuHu	**(2)** 4.09	0.96	1.54	7.90	0.93	32,961.62
BNHyPo	37.31	8.05	1.54	**(2)** 1.14	0.93	52,057.84
BNHyHu	27.95	5.31	3.08	**(3)** 1.34	**(3)** 0.90	41,746.92
MSWR	**(1)** 0.73	**(1)** 0.66	**(2)** 10.77	2.86	1.21	**(1)** 18,981.73
CBR1	**(3)** 4.46	0.92	7.69	21.81	0.95	31,593.05
CBR2	6.73	**(3)** 0.89	**(1)** 15.38	15.65	**(3)** 0.90	29,604.43
CBR3	6.09	**(2)** 0.84	**(3)** 9.23	13.26	**(2)** 0.89	**(3)** 28,976.31
Mean Effort	27.94	5.31	3.08	**(3)** 1.34	0.90	41,627.68
Median Effort	4.95	**(3)** 0.89	**(1)** 15.38	4.62	**(1)** 0.78	**(2)** 27,151.17

checked using the nonparametric Wilcoxon Signed Ranks test ($\alpha = 0.05$) with both absolute residuals and z. However, whenever the results using absolute residuals and z differed, we based our discussion on the former because, unlike z, absolute residuals present symmetry. Finally, detailed diagrams of all statistically significant relationships, for both absolute residuals and z, are available in the Appendix, which can be found at http://doi.ieeecomputersociety.org/10.1109/TSE.2008.64.

6.2 Comparison of Techniques

The techniques were compared using two validation sets each of 65 projects. The values obtained for each validation set, for each effort estimation technique, using six different prediction measures, are shown in Tables 8 and 9, respectively. Note that we also benchmarked the results against the Mean- and Median-based effort models, i.e., the mean and median effort for the training sets were used as estimated effort. BNAuPo, BNAuHu, BNHyPo, BNHyHu, MSWR, CBR1, CBR2, and CBR3 stand for, respectively, BN automatically generated using PowerSoft, BN automatically generated using Hugin, BN Hybrid model using PowerSoft, BN Hybrid model using Hugin, MSWR, CBR using one analogy, CBR using two analogies, and CBR using three analogies.

Table 8 presents the results using **validation set 1**, suggesting that all techniques presented poor predictions; however, it is important to interpret these results taking into account the accuracy obtained also using the mean and median efforts. We have also identified (in brackets) the three best results for each measure used, which suggest that the three best predictions were obtained, in descending order, using MSWR, BNHyHu, and CBR2. These results are partially supported by boxplots of absolute residuals (see Fig. 3 in the Appendix, which can be found at http://doi.ieeecomputersociety.org/10.1109/TSE.2008.64), which suggest that MSWR, BNHyHu, and Median provide the best predictions (in descending order). Boxplots of z values (Fig. 4 in the Appendix) show a different trend, suggesting that BNHyHu, CBR1, and MSWR/Median provide the best predictions (in descending order).

The statistical significance tests based on **absolute residuals** show the following trends:

1. MSWR was the only technique that outperformed all other techniques and also the only technique that presented accuracy significantly superior to Median-based predictions.
2. Except for MSWR, the BNHyHu model presented either similar to or significantly better accuracy (Mean effort, CBR1, BNHyPo) than the remaining techniques. This model was obtained using the same Bayesian tool and a very similar process to that employed in [26], [27], [28]. The difference between the process used in this paper and the one used in

[26], [27], [28] is that Mendes optimized the BN's causal graph by applying automated learning to a structure that contained only variables that presented the highest correlation with TE. 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.

3. Predictions obtained using BNAuPo, BNHyHu, MSWR, CBR1, CBR2, and CBR3 were significantly superior to those using the Mean effort.
4. CBR2, CBR3, BNAuHu, and BNHyHu presented similar accuracy to Median-based predictions; CBR1, BNAuPo, and BNHyPo showed significantly worse accuracy than Median-based predictions.
5. The worst model was BNHyPo, showing significantly worse predictions than any other techniques, including Mean-based predictions. This strongly suggests that the choice of algorithm used by a given BN tool to automatically learn probabilities from data is extremely important as it may strongly influence the predictions obtained. Both tool-generated BN models presented similar accuracy to CBR1.

However, the trends detailed above differed when checking the statistical significance of results using z values: Here, the technique that significantly outperformed any other technique was BNHyHu, not MSWR. MSWR did not significantly outperform CBR1 or Median-based predictions and CBR1 and BNHyHu were the only two techniques to outperform Median-based predictions. However, despite the large differences, z-based results confirmed that the BNHyPo model provided significantly worse predictions than any other technique, including Mean-based predictions.

Table 9 presents the results using **validation set 2**, which also suggest that all techniques presented poor predictions. Herein, we have also identified (in brackets) the three best results for each measure used, which, when aggregated, suggest that the three best predictions were obtained, in descending order, using MSWR, CBR3, and CBR2/CBR3/BNHyHu. These results were partially supported by boxplots of absolute residuals (see Fig. 5 in the Appendix, which can be found at http://doi.ieeecomputersociety.org/10.1109/TSE.2008.64) and boxplots of z values, which suggested that MSWR, Median, CBR1 and MSWR, CBR2, CBR3 provide the best predictions (in descending order), respectively.

The statistical significance tests based on **absolute residuals** showed similar trends to those observed for validation set 1: 1) most techniques (BNAuPo, BNAuHu, BNHyHu, MSWR, CBR1, CBR2, and CBR3) presented significantly superior predictions to predictions obtained using Mean effort; 2) only MSWR presented accuracy significantly superior to Median-based predictions. However, 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) and the best BN model was BNAuHu and not BNHyHu. BNAuHu's predictions were significantly superior to those from any other BN model and were similar to all CBR-based predictions. The worst model was again BNHyPo.

This time the statistical significance tests based on **z values** showed very similar results to those obtained using **absolute residuals**. The only differences were given as follows: Both MSWR and CBR1 presented accuracy significantly superior to Median-based predictions. The Median-based predictions were only superior to predictions obtained using BNAuPo, BNHyHu, and BNHyPo. BNAuHu presented predictions significantly worse than those using BNAuPo.

6.3 Discussion

In terms of the tool-based BNs (BNAuPo and BNAuHu), most trends were common across validation sets and statistical significance tests, for both absolute residuals and z values, as follows: 1) BNAuPo showed significantly better predictions than the Mean effort model and BNHyPo. Conversely, BNAuPo showed significantly worse predictions than the Median effort model, MSWR, CBR2, and CBR3; 2) BNAuHu showed significantly better predictions than BNHyPo and significantly worse predictions than MSWR. Overall, MSWR presented significantly superior accuracy to both tool-based BN models. A few other trends not common across validation sets, absolute residuals, and z values are detailed in Figs. 7-8 and 13-14 in the Appendix, which can be found at http://doi.ieeecomputersociety.org/10.1109/TSE.2008.64.

In terms of the hybrid BNs (BNHyPo and BNHyHu), the results that were common throughout validation sets and statistical significance tests, for both absolute residuals and z values, were given as follows: 1) BNHyPo was statistically significantly worse than any other techniques/models, including the mean effort model; 2) BNHyHu showed significantly better predictions than mean-based estimations and significantly worse predictions than those using MSWR. Overall, MSWR also presented significantly superior accuracy to both hybrid BN models. In addition to the abovementioned results for BNHyHu, there were other trends that differed drastically between validation sets and statistical significance tests: 1) When based on validation set 1, absolute residuals showed BNHyHu to provide similar predictions to both tool-based BNs, CBR2, CBR3, Median effort, and superior predictions to CBR1 and BNHyPo. However, z values showed BNHyHu to be significantly superior to any other technique/model. 2) Conversely, when based on validation set 2, both absolute residuals and z values showed that BNHyHu presented predictions, except for the Mean effort and BNHyPo, significantly worse than any other technique/model. The results for BNHyHu using validation set 1 were similar to those by Mendes [26], [27], [28] when also using a hybrid Bayesian model with a different DE-based causal graph and a smaller data set from the Tukutuku database. Note that our study and those of Mendes [26], [27], [28] 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 and statistical significance tests, we compared the characteristics of both training and validation sets, detailed as follows:

- 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 validation set 1 were slightly larger in total number of pages and effort, and smaller in total number of images than the projects in training set 1. 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 validation set 2 were slightly smaller in total number of pages and effort and larger in total number of images than the projects in training set 2. 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 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.

In terms of the remaining techniques and models, most results were common throughout validation sets and statistical significance tests, for both absolute residuals and z values, as follows: 1) MSWR provided significantly better predictions than CBR1, CBR2, CBR3, Median effort, and Mean effort; 2) Median effort was significantly better than the Mean effort; 3) CBR1, CBR2, and CBR3 provided significantly better predictions than the Mean effort. A few other trends not common across validation sets, absolute residuals, and z values are detailed in Figs. 9-10 and 15-16 in the Appendix, which can be found at http://doi.ieeecomputersociety.org/10.1109/TSE.2008.64.

What are the practical implications of these results?

MSWR was the only technique to show significantly superior predictions to those obtained using a Median effort-based model. This means that Web companies[3] looking to use a small data set of cross-company data to help estimate effort for their new projects should build their effort models using MSWR or use the Median effort as their second best alternative.

3. Applies to Web companies that either volunteered data to the Tukutuku database or that develop similar projects to those in the Tukutuku database.

Our results suggest that building data-driven and hybrid BNs that include complex causal graphs may only be a suitable approach if using a data set much larger than the one used in this study because a larger data set will provide project data that represents a wider range of possible combinations of parents' states and, hence, more realistic probabilities. The BN models used in [26], [27], [28], [42] were much simpler and smaller than the ones presented in this study and this may explain why both provided better predictions than the other techniques they were being compared against. Bibi et al. [4] built a large data-driven BN to predict productivity. The original BN model provided poor prediction accuracy; however, after modifying the BN's graph based on the authors' experiences, the predictions improved, but no statistical significance tests were conducted and the changes made to the BN's graph were not suggested or confirmed by an independent DE. Other options we are currently investigating include the building of BN models completely elicited from DEs and the tuning of data-driven probabilities by DEs.

Other practical implications of this work are that we would like to urge researchers in Web/software engineering to always: 1) include simpler models (e.g., median and mean-based effort models) and use at least two training/validation set combinations in any accuracy comparisons they carry out; 2) document the BN tool(s) employed and the algorithms used for structure and probability learning whenever investigating data-driven and/or hybrid BNs.

7 Threats to the Validity of Results

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

- Choice of BN tool: Hugin and PowerSoft were chosen because Hugin was a BN tool available to the authors and PowerSoft is freeware. Given that the two BNs used in this study showed different results, one can argue that there is a probability that other BN tools would also provide different results from those we presented herein. A more detailed comparison among different BN tools is one of the topics of our future work.
- The data set used in this study did not provide enough data capturing all relevant combinations of states among variables. This is a drawback to anyone willing to build BN models from data. There are other cross-company data sets of software projects much larger than the Tukutuku database; however, they only store data on conventional software projects and use size measures not adequate to size Web applications [32], [43].
- Some of the effort values provided were guesstimates; however, they corresponded to roughly 6 percent of the data; thus, we believe these projects do not present an important threat to the study.
- The Tukutuku database does not record projects' age (maturity) and, as such, we were unable to allocate projects to training and validation sets using age as a criterion and, instead, they were allocated randomly to training and validation sets.

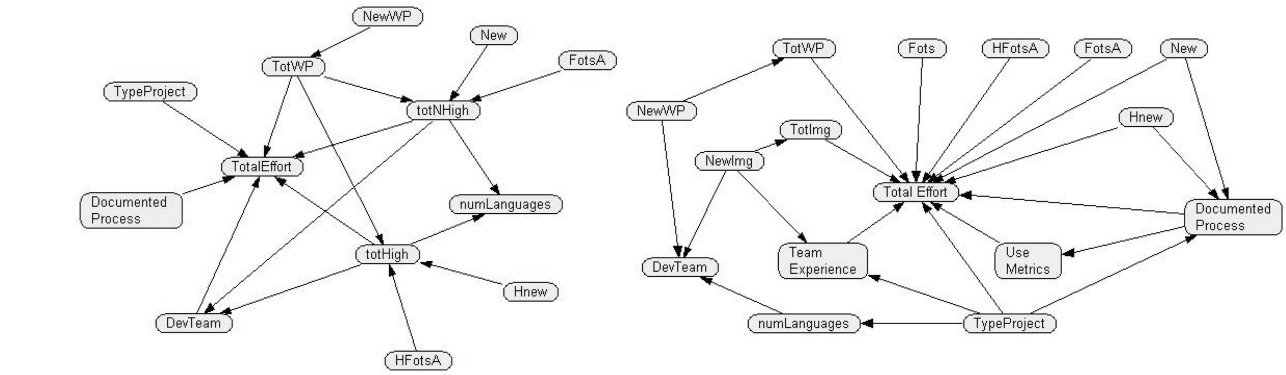


Fig. 4. BN causal graphs elicited by two DEs.

- The choice of variable discretization, structure learning algorithms, parameter estimation algorithms, and the number of categories used in the discretization 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 data set being used and the amount of data available. Our future work includes the comparison of our results with those using variables that were discretized using a greater number of categories and a different choice of discretization.
- Our study only used the Tukutuku variables to elicit BN's graphs given that any extra nodes added to a causal graph 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 graphs with DEs, which are not restricted to the Tukutuku variables. This will give us the opportunity to revisit the Tukutuku variables and also to investigate the possibility of eliciting a large and unified Web effort model.
- The Tukutuku data set does not represent a random sample of projects; therefore, the results presented herein may only be applicable to the Web companies that volunteered data to the Tukutuku project and companies that develop similar projects to those used in this study.
- This study only investigated the use of data-driven and hybrid BN models and we were unable to use a large data set of Web projects as the Tukutuku database only contains data on 195 projects. This may have had a significantly detrimental 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 algorithms available in the BN tools used, which we believe may have affected 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 models were based on a causal graph elicited from only one DE and this graph differed from the DE-based graph used in [26], [27], [28]. Recently, we asked another two DEs to elicit BN graphs using only the Tukutuku variables and yet again their causal graphs differed from the two structures previously elicited by DEs (see Fig. 4). All four DEs are directors of successful Web companies with large experience in developing and managing Web projects. Therefore, it does not seem applicable to choose one graph as best and any conclusions based on models derived from a single DE or BN tool should be interpreted with care. However, as part of our future work, we plan to merge these four graphs and have its predictions compared to those provided by each of the four separate causal graphs.

8 Conclusions and Future Work

This paper has presented the results of an investigation where eight BN models were compared on their accuracy to estimate effort for Web projects. Four models were automatically generated using two different BN tools—Hugin and PowerSoft; another four were Hybrid BN models, built using a causal graph elicited by a DE (a director of an established Web company in Auckland, New Zealand), with probabilities automatically "learned" 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 MSWR and CBR. The measures of accuracy employed were the MMRE, MdMRE, Pred(25), MEMRE, MdEMRE, absolute residuals, z, and Mean and Median efforts 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 nonparametric statistical significance test—the Wilcoxon Signed Paired Test, with $\alpha = 0.05$.

The five different BN causal graphs built suggested the following trends: The size of the development team has a direct causal effect on the use of a documented process; being involved in a process improvement program has a direct causal effect on the use of metrics throughout a project; total number of Web pages has a direct causal effect on the total amount of effort required to develop a Web application. Other suggested possible relationships were given as follows: *Fots* and *HFotsA*, *TeamExp* and *DocProc*,

TeamExp and *ProImpr*, *DevTeam* and *nLang*, and *Metrics* and *DocProc*; however, the direction of the causal relationship differed between BN graphs.

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

- BNAuPo showed significantly better predictions than the Mean effort model and BNHyPo and showed significantly worse predictions than the Median effort model, MSWR, CBR2, and CBR3.
- BNAuHu showed significantly better predictions than BNHyPo and significantly worse predictions than MSWR.
- BNHyPo was statistically significantly worse than any other techniques/models, including the mean effort model.
- BNHyHu showed significantly better predictions than mean-based estimations and significantly worse predictions than those using MSWR.
- MSWR presented significantly superior accuracy to both tool-based and hybrid BN models.
- MSWR provided significantly better predictions than CBR1, CBR2, CBR3, Median effort, and Mean effort.
- Median effort was significantly better than the Mean effort.
- CBR1, CBR2, and CBR3 provided significantly better predictions than the Mean effort.

Overall, MSWR was the only technique to show significantly superior predictions to those obtained using a Median effort-based model, thus suggesting that MSWR may be the only suitable effort estimation technique to use by Web companies who wish to estimate effort for new projects based on a small cross-company data set of Web projects. However, given that the Tukutuku database does not represent a random sample of projects, these results are only applicable to Web companies that volunteered data to the Tukutuku database and Web companies that develop projects similar to those in the Tukutuku database.

We urge researchers in Web/software engineering to always include simpler models (e.g., median and mean-based effort models) in any accuracy comparisons they perform and to use at least two training/validation set combinations. In addition, we would also urge researchers investigating data-driven and/or hybrid BNs to document the BN tool(s) employed and the algorithms used for structure and probability learning.

There were several threats to the validity of this study, as discussed in Section 7.

As part of our future work we plan to:

- Conduct a more detailed comparison using different BN tools.
- Compare the results presented in this paper with those using variables that were discretized using a greater number of categories and a different choice of discretization.
- Elicit other BN structures with DEs which are not restricted to the Tukutuku variables. This will give us the opportunity to revisit the Tukutuku variables and also to investigate the possibility of eliciting a large and unified Web effort model.
- Elicit 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.

ACKNOWLEDGMENTS

This work is supported by the Royal Society of New Zealand, under the Marsden Fund Research Grant UOA0611. The authors would like to thank the DE and the Web companies who volunteered data to the Tukutuku project, in particular A/Prof. F. Ferrucci. The authors would also like to thank A. Nicholson, K. Korb, their PhD students, and C. Pollino for early discussions in the use of BNs. Finally, the authors would like to thank the reviewers for their insightful comments and suggestions.

REFERENCES

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

[2] L. Baresi, S. Morasca, and P. Paolini, "An Empirical Study on the Design Effort for Web Applications," *Proc. Third Int'l Conf. Web Information Systems Eng.*, pp. 345-354, 2002.

[3] L. Baresi, S. Morasca, and P. Paolini, "Estimating the Design Effort for Web Applications," *Proc. Ninth Int'l Software Metrics Symp.*, pp. 62-72, 2003.

[4] S. Bibi, I. Stamelos, and L. Angelis, "Bayesian Belief Networks as a Software Productivity Estimation Tool," *Proc. First Balkan Conf. Informatics*, 2003.

[5] L.C. Briand, T. Langley, and I. Wieczorek, "A Replicated Assessment and Comparison of Common Software Cost Modeling Techniques," *Proc. 22nd Int'l Conf. Software Eng. (ICSE '00)*, pp. 377-386, 2000.

[6] L.C. Briand, K. El Emam, and F. Bomarius, "COBRA: A Hybrid Method for Software Cost Estimation, Benchmarking and Risk Assessment," *Proc. 20th Int'l Conf. Software Eng.*, pp. 390-399, 1998.

[7] J. Cheng, D.A. Bell, and W. Liu, "Learning Belief Networks from Data: An Information Theory Based Approach," *Proc. Sixth ACM Int'l Conf. Information and Knowledge Management*, 1997.

[8] 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 Web Eng.*, pp. 75-92, June 2000.

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

[10] G. Costagliola, S. Di Martino, F. Ferrucci, C. Gravino, G. Tortora, and G. Vitiello, "Effort Estimation Modeling Techniques: A Case Study for Web Applications," *Proc. Sixth Int'l Conf. Web Eng.*, pp. 9-16, 2006.

[11] M.J. Druzdzel, A. Onisko, D. Schwartz, J.N. Dowling, and H. Wasyluk, "Knowledge Engineering for Very Large Decision-Analytic Medical Models," *Proc. Ann. Meeting Am. Medical Informatics Assoc.*, pp. 1049-1054, 1999.

[12] M.J. Druzdzel and L.C. van der Gaag, "Building Probabilistic Networks: Where Do the Numbers Come from?" *IEEE Trans. Knowledge and Data Eng.*, vol. 12, no. 4, pp. 481-486, July/Aug. 2000.

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

[14] N. Fenton, W. Marsh, M. Neil, P. Cates, S. Forey, and M. Tailor, "Making Resource Decisions for Software Projects," *Proc. 26th Int'l Conf. Software Eng.*, pp. 397-406, 2004.

[15] R. Fewster and E. Mendes, "Measurement, Prediction and Risk Analysis for Web Applications," *Proc. Seventh IEEE Int'l Software Metrics Symp.*, pp. 338-348, 2001.

[16] R. Jeffery, M. Ruhe, and I. Wieczorek, "Using Public Domain Metrics to Estimate Software Development Effort," *Proc. Seventh IEEE Int'l Software Metrics Symp.,* pp. 16-27, 2001.
[17] F.V. Jensen, *An Introduction to Bayesian Networks.* UCL Press, 1996.
[18] B.A. Kitchenham, "A Procedure for Analysing Unbalanced Data Sets," *IEEE Trans. Software Eng.,* vol. 24, no. 4, pp. 278-301, Apr. 1998.
[19] B.A. Kitchenham, L.M. Pickard, S.G. MacDonell, and M.J. Shepperd, "What Accuracy Statistics Really Measure," *IEE Proc. Software Eng.,* vol. 148, no. 3, June 2001.
[20] B.A. Kitchenham and E. Mendes, "A Comparison of Cross-Company and Single-Company Effort Estimation Models for Web Applications," *Proc. Int'l Conf. Empirical Assessment in Software Eng.,* pp. 47-55, 2004.
[21] A.J. Knobbe and E.K.Y. Ho, "Numbers in Multi-Relational Data Mining," *Proc. Ninth European Conf. Principles and Practice of Knowledge Discovery in Databases),* 2005.
[22] S.L. Lauritzen, "The EM Algorithm for Graphical Association Models with Missing Data," *Computational Statistics and Data Analysis,* vol. 19, pp. 191-201, 1995.
[23] C. Lokan and E. Mendes, "Cross-Company and Single-Company Effort Models Using the ISBSG Database: A Further Replicated Study," *Proc. ACM/IEEE Int'l Symp. Empirical Software Eng.,* pp. 75-84, 2006.
[24] S.M. Mahoney and K.B. Laskey, "Network Engineering for Complex Belief Networks," *Proc. 12th Ann. Conf. Uncertainty in Artificial Intelligence,* pp. 389-396, 1996.
[25] L. Mangia and R. Paiano, "MMWA: A Software Sizing Model for Web Applications," *Proc. Fourth Int'l Conf. Web Information Systems Eng.,* pp. 53-63, 2003.
[26] E. Mendes, "Predicting Web Development Effort Using a Bayesian Network," *Proc. Int'l Conf. Evaluation and Assessment in Software Eng.,* pp. 83-93, 2007.
[27] E. Mendes, "The Use of a Bayesian Network for Web Effort Estimation," *Proc. Seventh Int'l Conf. Web Eng.,* pp. 90-104, 2007.
[28] E. Mendes, "A Comparison of Techniques for Web Effort Estimation," *Proc. ACM/IEEE Int'l Symp. Empirical Software Eng.,* pp. 334-343, 2007.
[29] E. Mendes and B.A. Kitchenham, "Further Comparison of Cross-Company and Within-Company Effort Estimation Models for Web Applications," *Proc. 10th IEEE Int'l Software Metrics Symp.,* pp. 348-357, 2004.
[30] E. Mendes and S. Counsell, "Web Development Effort Estimation Using Analogy," *Proc. 12th Australian Software Eng. Conf.,* pp. 203-212, 2000.
[31] 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 Int'l Symp. Empirical Software Eng.,* pp. 79-90, 2002.
[32] E. Mendes, N. Mosley, and S. Counsell, "Web Metrics—Metrics for Estimating Effort to Design and Author Web Applications," *IEEE Multimedia,* pp. 50-57, Jan.-Mar. 2001.
[33] E. Mendes, N. Mosley, and S. Counsell, "The Application of Case-Based Reasoning to Early Web Project Cost Estimation," *Proc. 26th IEEE Int'l Computer Software and Applications Conf.,* pp. 393-398, 2002.
[34] E. Mendes, N. Mosley, and S. Counsell, "Comparison of Length, Complexity and Functionality as Size Measures for Predicting Web Design and Authoring Effort," *IEE Proc. Software,* vol. 149, no. 3, pp. 86-92, June 2002.
[35] E. Mendes, N. Mosley, and S. Counsell, "Do Adaptation Rules Improve Web Cost Estimation?" *Proc. 14th ACM Conf. Hypertext and Hypermedia,* pp. 173-183, 2003.
[36] E. Mendes, N. Mosley, and S. Counsell, "Investigating Web Size Metrics for Early Web Cost Estimation," *J. Systems and Software,* vol. 77, no. 2, pp. 157-172, 2005.
[37] E. Mendes, N. Mosley, and S. Counsell, "The Need for Web Engineering: An Introduction," *Web Eng.,* E. Mendes and N. Mosley, eds., pp. 1-26, Springer-Verlag, 2005.
[38] E. Mendes, I. Watson, C. Triggs, N. Mosley, and S. Counsell, "A Comparison of Development Effort Estimation Techniques for Web Hypermedia Applications," *Proc. Eighth IEEE Int'l Software Metrics Symp.,* pp. 141-151, June 2002.
[39] E. Mendes, I. Watson, C. Triggs, N. Mosley, and S. Counsell, "A Comparative Study of Cost Estimation Models for Web Hypermedia Applications," *Empirical Software Eng.,* vol. 8, no. 2, pp. 163-196, 2003.
[40] M. Neil, N. Fenton, and L. Nielsen, "Building Large-Scale Bayesian Networks," *The Knowledge Eng. Rev. (KER),* vol. 15, no. 3, pp. 257-284, 2000.
[41] J. Pearl, *Probabilistic Reasoning in Intelligent Systems.* Morgan Kaufmann, 1988.
[42] 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, pp. 615-624, July 2005.
[43] D.J. Reifer, "Web Development: Estimating Quick-to-Market Software," *IEEE Software,* pp. 57-64, Nov.-Dec. 2000.
[44] D.J. Reifer, "Ten Deadly Risks in Internet and Intranet Software Development," *IEEE Software,* pp. 12-14, Mar.-Apr. 2002.
[45] M. Ruhe, R. Jeffery, and I. Wieczorek, "Cost Estimation for Web Applications," *Proc. 25th Int'l Conf. Software Eng.,* pp. 285-294, 2003.
[46] M.J. Shepperd and G. Kadoda, "Using Simulation to Evaluate Prediction Techniques," *Proc. Seventh IEEE Int'l Software Metrics Symp.,* pp. 349-358, 2001.
[47] I. 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, pp. 51-60(10), Jan. 2003.
[48] H. Steck and V. Tresp, "Bayesian Belief Networks for Data Mining," *Proc. Second Workshop Data Mining and Data Warehousing,* Sept. 1999.
[49] O. Woodberry, A. Nicholson, K. Korb, and C. Pollino, "Parameterising Bayesian Networks," *Proc. Australian Conf. Artificial Intelligence,* pp. 1101-1107, 2004.

Emilia Mendes received the PhD degree in computer science from the University of Southampton, United Kingdom, in 1999 after working in the software industry for 10 years. She is an associate professor in computer science at the University of Auckland, New Zealand. She has active research interests in the areas of empirical Web and software engineering, evidence-based research, hypermedia, computer science, and software engineering education, in which areas she has published widely, with over 100 refereed publications, including two books, one edited (2005) and one authored (2007). She is a member of the editorial boards of the *International Journal of Web Engineering and Technology*, the *Journal of Web Engineering*, the *Journal of Software Measurement*, the *International Journal of Software Engineering and Its Applications*, the *Empirical Software Engineering Journal*, and the *Advances in Software Engineering Journal*.

Nile Mosley received the PhD degree in 1997 in differential geometry/computational fluid dynamics from Northampton University, United Kingdom. He has worked in a range of industries and is now the CEO of MetriQ Limited (http://www.metriq.biz), the world's first and only provider of truly hands-free time sheet solution software, established in 2004. His areas of interest are effort estimation and management, and process improvement.