A comparison of software effort prediction models using small datasets

Chikako van Koten

This work has been submitted to the IEEE for possible publication. Copyright may be transferred without notice, after which this version may no longer be accessible.
Abstract

Constructing an accurate effort prediction model is a challenge in Software Engineering. One difficulty practitioners often experience is that they have only a very small amount of local data to construct a model. The small dataset limits predictive accuracy of the model, since the accuracy deteriorates as the size of the dataset decreases. This paper compares three different software development effort prediction models that are applicable to these small datasets. They are: (1) Bayesian statistical models, (2) multiple linear regression models and (3) case-based reasoning/analogy-based models. The predictive accuracy of these models is evaluated using two different software datasets. The results have shown that the accuracy of the Bayesian statistical models is higher than or competitive with that of the others, when calibrated using data collected from fewer than 10 systems. These suggest that the Bayesian statistical model would be a better choice in effort prediction when the practitioners have only a very small dataset, consisting of fewer than 10 systems similar to their system of interest.

Multivariate statistics, Modeling methodologies, Management techniques, Statistical methods, Cost estimation, Time estimation

I. INTRODUCTION

Accurate effort prediction at an early stage is often an important factor for successful software development. Hence, software development effort prediction models are a useful tool for software practitioners and project managers, providing an estimate based on data collected from systems previously developed. However, for many software systems, constructing an accurate model for predicting the effort is still a challenge.

Software development effort can vary depending on a large number of factors. In addition, the relationships of these factors may change from one system to another. Considering this, one way to achieve good accuracy is to build a model specific to the system of interest, and calibrate the model using locally collected data, consisting of only systems similar to the system of interest. Using local data is usually preferred to using multi-organizational data, since a number of studies reported that the use of multi-organizational datasets did not lead to as good accuracy as that obtained from local data [14], [24], suggesting that the relationships of the influential factors may also change, depending on factors specific to the organization where the development is undertaken.

Developing a software system is a complex endeavour that requires a large amount of time. Consequently, for a small or medium-sized developing organization, unfortunately it is often the
case that the amount of local data is very small. Even for a larger organization, it could also
be the case if the system of interest is different from those previously developed. Predictive
accuracy of software development effort prediction models generally deteriorates as the size of
the dataset decreases [34]. Hence, these small local datasets significantly limit the usefulness of
the models.

This paper compares three models, which can predict software development effort using only
a small amount of data. The models are: (1) Bayesian statistical models, (2) multiple linear
regression models and (3) case-based reasoning/analogy-based models built using AngelPlus [29].
The Bayesian statistical models studied are modified versions of the models previously developed
by the author for predicting software effort [34]. The models are based on a Bayesian statistical
modelling approach [2].

The three models are compared in two separate case studies, using a different software dataset.
One of the dataset was collected by the author, while the other is a dataset in the public domain.
Each dataset was split into two subsets, one for calibrating the models and the other for evaluating
the predictive accuracy of the models. The accuracy is measured in terms of the absolute residuals
and magnitude of relative error, which are commonly used in the software effort prediction
literature.

Although there have been a number of comparative studies of software development effort
prediction models to date [1], [7]–[9], [11]–[13], [15], [17], [19]–[23], [25], [26], [28], [30],
[31], these studies in general used different models and datasets, and reported largely conflicting
results regarding which model is most accurate. This study is distinguishable from these previous
studies because of its focus on a small dataset, particularly dataset consisting of fewer than 10
software systems.

The structure of the reminder of this paper is as follows. Section II describes each of the
three models studied, with an emphasis on the Bayesian statistical models since they are new in
comparison to the other two models. Section III provides the definition of predictive accuracy
measures used. Sections IV and V present the results of two separate case studies, in which the
predictive accuracy of the models is evaluated and compared. Section VI presents conclusions
and a direction of future studies.
II. SOFTWARE EFFORT PREDICTION MODELS

A. Bayesian statistical models

1) Bayesian statistical approach: Bayesian statistics uses Bayes’ theorem:

\[
P(Y \mid X) = \frac{P(X \mid Y)P(Y)}{P(X)}
\]

(1)

to make statistical inference, where \(P(Y)\), an initial probability distribution (prior probability distribution) of \(Y\), is updated to a new probability distribution (posterior probability distribution) \(P(Y \mid X)\), using given information about \(P(X)\) and \(P(X \mid Y)\).

Let us assume that software development effort \(Y\) can be predicted by a set of known predictor variables \(X\) and a set of unknown parameters \(\theta\). Then, according to probability theory, the joint probability distribution of \(Y, X\) and \(\theta\) can be defined as:

\[
P(Y, X, \theta) = P(Y, X \mid \theta)P(\theta)
\]

(2)

The unknown set of parameters \(\theta\) is added here in order to define the joint probability distribution using known probability density functions and mathematical equations. Equation 2 means that the joint probability distribution is composed by two probability distributions: \(P(\theta)\), the prior probability distribution of \(\theta\), and \(P(Y, X \mid \theta)\), the likelihood of data.

Alternatively, from the Bayes’ Theorem 1 it is also known that:

\[
P(\theta \mid Y, X) = \frac{P(Y, X \mid \theta)P(\theta)}{P(Y, X)}
\]

(3)

where \(P(\theta \mid Y, X)\) is the posterior probability distribution of \(\theta\).

Considering that \(P(Y, X)\) is a constant for a given set of observed values of \(Y\) and \(X\) in data, the Equation 3 means that:

\[
P(\theta \mid Y, X) \propto P(Y, X \mid \theta)P(\theta)
\]

(4)

That is, the posterior probability distribution of \(\theta\) is proportional to the likelihood of data multiplied by the prior distribution, which is the joint probability distribution of \(Y, X\) and \(\theta\). However, since the proportional constant in the relationship 4 disappears when the posterior probability distribution and the joint probability distribution are normalized, the relationship 4 actually states that the posterior probability distribution equals the updated joint probability distribution.
Hence, if the joint probability distribution is predefined as a combination of the likelihood of data and the prior probability distribution, and the likelihood is updated using data, the posterior probability distribution of $\theta$ can be known from the updated joint probability distribution. Once the posterior probability distribution is known, the values of unknown $\theta$ can be estimated from the distribution, then the probability distribution of software development effort $P(Y)$ can be obtained from the updated joint probability distribution $P(Y, X, \theta)$, by integrating the joint probability distribution over the known predictor variables $X$ and the estimated parameters $\theta$.

This integration is often performed numerically, using a software tool. This study uses a tool called WinBUGS (Windows version of Bayesian analysis using Gibbs sampling). A version of WinBUGS can be obtained from Imperial College and Medical Research Council in the U.K. under the license agreement. This paper has used the version 1.4.1.

WinBUGS is a Markov chain Monte Carlo (MCMC) sampling program, which allows users to draw a large number of samples from a probability distribution. In the case of this study, the target probability distribution is the updated joint probability distribution. Thus, WinBUGS draws a large number of samples from the joint probability distribution, and simulates the probability distribution of software development effort $P(Y)$ using these samples. The sampling algorithm used in WinBUGS is the Gibbs sampler [10].

In the MCMC sampling, its sampling distribution is known to converge at the target probability distribution given a sufficient amount of time. Thus, it is important to ensure that the target probability distribution is simulated only after the convergence has been reached. One way to assess whether the convergence has been reached is to monitor time series of the samples. Then, once the convergence is confirmed, the samples drawn before the convergence must be discarded. This study drew 150000 samples from the updated joint probability distribution, and discarded the first 75000 samples. That is, the effort probability distribution was simulated using only the second 75000 samples.

Hence, in this Bayesian statistical approach, the primary task for predicting software development effort is to investigate how the likelihood of data and the prior probability distributions can be defined appropriately for the system of interest. A number of different Bayesian statistical models have been so far developed by the author and their predictive accuracy has been evaluated and reported in other publications with different software datasets [33], [34].
2) **Bayesian multivariate normal distribution (BMVN) model:** The Bayesian multivariate normal distribution (BMVN) model assumes correlations between software development effort and \( k \) predictor variables. Then, these correlations are modeled as a multivariate normal distribution. The multivariate normal distribution is a normal distribution on a multi-dimensional space, defined by multiple variables. The assumed multivariate normal distribution has unknown mean vector \( \mu_{ji} \) and unknown variance matrix \( 1/\tau_{j,m} \):

\[
\text{EFFORT}_i, X_{1i}, \ldots, X_{ki} \sim \text{MV N} \left( \mu_{ji}, 1/\tau_{j,m} \right) \\
\text{for} \quad j = 1, \ldots, k + 1 \\
m = 1, \ldots, k + 1
\]

(5)

where \( \mu_{1i} \) is the mean value of effort, and \( \mu_{ji} \) for \( j = 2, \ldots, k + 1 \) is the mean value of the \( h \)-th predictor variable \( X_h \) for \( h = 1, \ldots, k \), respectively. The values of \( \mu_{ji} \) are calculated from calibration data.

The variance matrix is defined as:

\[
\begin{pmatrix}
1/\tau_{1,1} & \cdots & 1/\tau_{1,k+1} \\
\vdots & \ddots & \vdots \\
1/\tau_{k+1,1} & \cdots & 1/\tau_{k+1,k+1}
\end{pmatrix}
\]

Then, the prior probability distributions of the unknown means \( \mu_{ji} \) are assumed as a normal distribution with mean 0 and large variance, while \( \tau_{j,m} \), a Wishart distribution with the scale matrix equal to a \((k + 1) \times (k + 1)\) identity matrix:

\[
\begin{align*}
\mu_{ji} &\sim N \left( 0, 1000000 \right) \\
\tau_{j,m} &\sim \text{Wishart} \left( C_{j,m}, k + 1 \right)
\end{align*}
\]

(6)

(7)

where \( C_{j,m} = \begin{cases} 
1 & \text{if } j = m \\
0 & \text{otherwise} 
\end{cases} \)

The normal distributions of \( \mu_{ji} \) are assumed to have the mean value of 0, since prior to calibration, we have no knowledge of the mean value of effort as well as the mean values of predictor variables. The large variance value of 1000000 is chosen to ensure that the normal distributions are almost flat prior to calibration. The flat normal distribution can make it equally
possible for \( \mu_{ji} \) to take any value, and therefore, minimizes the effect of the defined prior probability distribution.

The prior probability distribution of \( \tau_{j,m} \) is a Wishart distribution, since the variance matrix of a multivariate normal distribution is known to be a Wishart distribution. The scale matrix of the Wishart distribution is then assumed as an identity matrix, since prior to calibration, we have absolutely no knowledge about the variance and covariance components of the matrix.

In addition, in order to achieve the convergence in the MCMC sampling as early as possible, the initial values of \( \tau_{j,m} \) in the BMVN model are set to the corresponding component of a \((k + 1) \times (k + 1)\) identity matrix. The initial values of \( \mu_{ji} \) are also set to the corresponding mean value, calculated from the calibration data. Setting initial values is not mandatory for the sampling algorithm to reach convergence. However, appropriate initial values reduce the time required for the convergence. If no initial value is set, the sampling algorithm automatically generates initial values to start from.

According to the joint probability distribution described above, the probability distribution of effort of a new software system also becomes a multivariate normal distribution:

\[
\text{EFFORT}_{\text{new}}, X_{1,\text{new}}, \ldots, X_{k,\text{new}} \sim \text{MVN} \left( \mu_{j,\text{new}}, 1/\tau_{j,m} \right) \quad \text{for} \quad j = 1, \ldots, k + 1 \quad m = 1, \ldots, k + 1
\]  

(8)

In this distribution, the values of \( 1/\tau_{j,m} \) are estimated from the updated joint probability distribution, and the values of \( \mu_{j,\text{new}} \), calculated from the calibration data and the predictor values of the new system. Once these values have been known, the target effort probability distribution is obtained from the multivariate normal distribution 8, by directly sampling a large number of \( \text{EFFORT}_{\text{new}} \) values. The predicted effort value is then obtained by calculating the mean of the effort probability distribution. This is because the BMVN model assumes the effort probability distribution is multivariate normal, and a multivariate normal distribution is symmetrical with a single peak at the mean.

3) **Bayesian multivariate normal distribution & regression (BMVR) model:** The Bayesian multivariate normal distribution & regression (BMVR) model assumes a linear relationship of software development effort with \( k \) predictor variables. However, this model, like the BMVN model, also assumes correlations among the predictor variables, and incorporates these correla-
tions into the linear relationship. That is, in the BMVR model, software development effort is
assumed to have a normal distribution with unknown mean $\mu_{yi}$ and unknown variance $1/\tau_y$:

$$\text{EFFORT}_i - \text{mean EFFORT} \sim N(\mu_{yi}, 1/\tau_y)$$

(9)

Then, the unknown mean $\mu_{yi}$ is assumed to be determined by the linear equation of $k$ predictor
variables $X_j$:

$$\mu_{yi} = \beta_1(X_{1i} - \bar{X}_1) + \cdots + \beta_k(X_{ki} - \bar{X}_k)$$

(10)

In addition, $k$ predictor variables $X_j$ are together assumed to have a multivariate normal
distribution with null mean vector and unknown variance matrix $1/\tau_{xj,m}$:

$$X_{1i} - \bar{X}_1, \ldots, X_{ki} - \bar{X}_k \sim \text{MVN}(0, 1/\tau_{xj,m})$$

for $j = 1, \ldots, k$

$$m = 1, \ldots, k$$

(11)

where the variance matrix is defined as:

$$
\begin{pmatrix}
1/\tau_{1,1} & \cdots & 1/\tau_{1,k} \\
\vdots & \ddots & \vdots \\
1/\tau_{k,1} & \cdots & 1/\tau_{k,k}
\end{pmatrix}
$$

Then, the prior probability distribution of $\tau_y$ is assumed as a gamma distribution with the
large scale parameter of 10000 and the small shape parameter of 0.0001. The prior probability
distribution of $\beta_j$ is assumed as a normal distribution with mean 0 and large variance, and of
$\tau_{xj,m}$, a Wishart distribution with the scale matrix equal to a $k \times k$ diagonal matrix with the
diagonal elements of 0.00001:

$$\tau_y \sim \text{Gamma}(10000, 0.0001)$$

(12)

$$\beta_j \sim N(0, 10000) \quad \text{for} \quad j = 1, \ldots, k$$

(13)

$$\tau_{xj,m} \sim \text{Wishart}(C_{j,m}, k)$$

where

$$C_{j,m} = \begin{cases}
0.00001 & \text{if } j = m \\
0 & \text{otherwise}
\end{cases}$$

(14)

The flat gamma distribution with the large scale parameter value of 10000 makes it equally
possible for $\tau_y$ to take any value, and therefore, minimizes the effect of the defined prior
probability distribution. The small shape parameter value of 0.0001 ensures that the normal distribution of software development effort has very large variance prior to calibration. Similarly, the flat normal distributions of $\beta_j$ with the large variance value of 10000 make it equally possible for $\beta_j$ to take any value, whereas the mean value of 0 assumes it is equally possible for $\beta_j$ to take either a positive or negative value prior to calibration.

The Wishart distribution assumes no knowledge about the variance and covariance components of the scale matrix prior to calibration, but ensures that at least the $k$ predictor variables $X_j$ have large variance.

In the case of the BMVR model, in order to achieve the convergence as early as possible, the initial value of $\tau_y$ is set to 1, while $\tau_{xj,m}$, the corresponding component of a $k \times k$ identity matrix. The initial values of $\beta_j$ are also set to 0.

According to the joint probability distribution described above, the probability distribution of effort of a new software system becomes a joint probability distribution:

$$\text{EFFORT}_{\text{new}} = \mu_{y,\text{new}} + \text{mean \ EFFORT}$$

where

$$\mu_{y,\text{new}} = \beta_1(X_{1,\text{new}} - \bar{X}_1) + \cdots + \beta_k(X_{k,\text{new}} - \bar{X}_k)$$

In Equation 15 the mean EFFORT is the average effort value calculated from the calibration data. In Equation 16 $X_{j,\text{new}}$ denotes the value of the predictor variable $X_j$ observed in the new system. The values of $\beta_j$ are estimated from the updated joint probability distribution. Once these values have been known, the target effort probability distribution is obtained from the above joint probability distribution 15 and 16, by directly sampling a large number of EFFORT_{new} values. Then, the predicted effort value is obtained by calculating the median of the effort probability distribution. This is because the effort probability distribution of the BMVR model can be skewed, and median is usually a better measure of the central tendency than mean when the distribution is skewed.

4) Random effect BMVR (RBMVR) model: The random effect BMVR (RBMVR) model is a modified BMVR model, which takes account of a random effect of linear coefficients $\beta_j$ in Equation 10. The RBMVR model was developed to improve predictive accuracy of the BMVR model, particularly when $k$, the number of predictor variables, is larger than the number of
software systems in the calibration data. Such a situation can often arise when the local dataset consists of fewer than 10 software systems.

When the number of the predictor variables becomes larger than that of the systems, it becomes no longer possible to determine the values of linear coefficients $\beta_j$ in Equation 10 from the data. This means that although some $\beta_j$s are still estimated from the data, the others are left undetermined. These undetermined $\beta_j$s make the BMVR model unstable and consequently, reduce its predictive accuracy. The RBMVR model extends the BMVR model by adding a random effect of $\beta_j$. And because of the additional random effect, even if some $\beta_j$ s are left undetermined from the data, the RBMVR model can still estimate the values of these undetermined $\beta_j$s using the values of the determined $\beta_j$s. Hence, the RBMVR model remains stable and consequently, its predictive accuracy is expected to be greater than that of the BMVR model.

If the number of the predictors is not larger than that of the systems, the random effect of $\beta_j$ still allows all $\beta_j$s to be determined from the data. This means that in this case, predictive accuracy of the RBMVR model is at least equivalent to that of the BMVR model, if not any better.

In the RBMVR model, software development effort is assumed to have a normal distribution with unknown mean $\mu_{yi}$ and unknown variance $1/\tau_y$:

$$\text{EFFORT}_i - \text{mean EFFORT} \sim N(\mu_{yi}, 1/\tau_y)$$

(17)

where the unknown mean $\mu_{yi}$ is assumed to be determined by the linear equation of $k$ predictor variables $X_j$:

$$\mu_{yi} = \beta_1 \text{Std.X}_{1i} + \cdots + \beta_k \text{Std.X}_{ki}$$

(18)

In Equation 18 the $\text{Std.X}_{ji}$ denotes the standardized value of $X_{ji}$, which is calculated from calibration data using the formula:

$$\text{Std.X}_{ji} = \frac{X_{ji} - \bar{X}_j}{\text{standard deviation of } X_j}$$

(19)

Then, $\text{Std.X}_j$ are together assumed to have a multivariate normal distribution with null mean vector and unknown variance matrix $1/\tau_{xj,m}$:

$$\text{Std.X}_{1i}, \ldots, \text{Std.X}_{ki} \sim MVN(0, 1/\tau_{xj,m})$$

for $j = 1, \ldots, k$

$$m = 1, \ldots, k$$

(20)
In addition, the linear coefficients $\beta_j$ in Equation 18 are assumed to have a normal distribution with unknown mean $\mu_{\beta}$ and unknown variance $1/\tau_{\beta}$:

$$\beta_j \sim N(\mu_{\beta}, 1/\tau_{\beta})$$ (21)

Then, the prior probability distribution of $\tau_y$ is assumed as a gamma distribution with the large scale parameter of 10000 and the small shape parameter of 0.0001, while $\tau_{x,j,m}$, a Wishart distribution with the scale matrix equal to a $k \times k$ diagonal matrix with the diagonal elements of 0.00001. The prior probability distribution of $\mu_{\beta}$ is assumed as a normal distribution with mean 0 and large variance, and of $\tau_{\beta}$, the same gamma distribution as that of $\tau_y$:

$$\tau_y \sim \text{Gamma}(10000, 0.0001)$$ (22)

$$\tau_{x,j,m} \sim \text{Wishart}(C_{j,m}, k)$$

where

$$C_{j,m} = \begin{cases} 0.00001 & \text{if } j = m \\ 0 & \text{otherwise} \end{cases}$$ (23)

$$\mu_{\beta} \sim N(0, 10000)$$ (24)

$$\tau_{\beta} \sim \text{Gamma}(10000, 0.0001)$$ (25)

For the RBMVR model, the initial values of $\tau_y$ and $\tau_{x,j,m}$ are the same as those of the BMVR model, whereas the initial value of $\tau_{\beta}$ is set to 1. Then, the initial value of $\mu_{\beta}$ is set to 0.

According to the joint probability distribution described above, the probability distribution of effort of a new software system becomes a joint probability distribution:

$$\text{EFFORT}_{\text{new}} = \mu_{y,\text{new}} + \text{mean \ EFFORT}$$ (26)

where

$$\mu_{y,\text{new}} = \beta_1 \text{Std}.X_{1,\text{new}} + \cdots + \beta_k \text{Std}.X_{k,\text{new}}$$ (27)

In Equation 26 the mean EFFORT is the average effort value calculated from the calibration data. In Equation 27 Std.$X_{j,\text{new}}$ denotes the standardized value of the predictor variable $X_j$ observed in the new system. The values of $\beta_j$ are estimated from the updated joint probability distribution, through the estimated values of $\mu_{\beta}$ and $\tau_{\beta}$. Once all these values have been known, the target effort probability distribution is obtained from the above joint probability distribution 26 and 27, by directly sampling a large number of $\text{EFFORT}_{\text{new}}$ values. Then, the
The predicted effort value is obtained by calculating the median of the effort probability distribution due to the same reason as the BMVR model.

**B. Multiple linear regression models**

The multiple linear regression models:

\[
\text{EFFORT} = \beta_0 + \beta_1 X_{1i} + \cdots + \beta_k X_{ki}
\]  

are constructed using a statistical software package SPSS11.0, with the same calibration data as that of the Bayesian statistical models. The multiple linear regression models are referred to as R models in tables presented in this paper.

**C. Case-based reasoning/analogy-based (CBR) models**

The case-based reasoning/analogy-based (CBR) models are constructed using AngelPlus [29], with the same calibration data as that of the above two models.

AngelPlus calculates the Euclidean distance between each software system in calibration data and a new system, and uses a number of software systems nearest to the new system for predicting the effort. The predicted effort is calculated as the unweighted average value of the nearest systems chosen.

The number of nearest systems used for prediction must be specified by the user. It is known that the predictive accuracy of the CBR model varies depending on the number of nearest systems chosen. However, since there exists no theory to determine such a number, this study identifies an appropriate number of nearest systems for each prediction as follows.

Assuming the calibration data contains \( n \) software systems, the CBR model is first calibrated using \( n - 1 \) systems. Secondly, the calibrated model predicts the effort of the one remaining system, using all possible numbers of nearest systems from 1 to \( n - 1 \). This results in \( n - 1 \) predicted values for this particular system. Thirdly, \( n - 1 \) magnitude of relative error (MRE) values are calculated from these predicted values. Fourthly, the previous process is repeated \( n \) times, to obtain \( n - 1 \) MRE values for each system in the data. In other words, \( n \) systems in the data now have \( n - 1 \) MRE values each. Finally, the mean MRE (MMRE) value of the \( n \) systems is calculated, one for each set of the \( n - 1 \) MRE values. The appropriate number of
nearest systems is then determined as the one that provides the smallest MMRE value among these \( n - 1 \) MMRE values.

A setback of this procedure is that finding the appropriate number for one prediction using a calibration dataset of \( n \) systems requires \( n \times (n - 1) \) trials. This is very labour-intensive, and becomes even more so as \( n \), the number of the systems, increases. From this point of view, the CBR models may not always be practical for predicting software effort, if the calibration data consists of a large number of systems.

**III. Predictive accuracy measures**

This paper evaluates and compares the predictive accuracy of the above models in terms of the absolute residual (Ab.Res.), the magnitude of relative error (MRE) and pred measures.

The Ab.Res. is the absolute value of residual given by:

\[
Ab.\text{Res.} = |\text{actual value} - \text{predicted value}|
\]  

The sum of the absolute residuals (Sum Ab.Res.), the median of the absolute residuals (Med.Ab.Res.) and the standard deviation of the absolute residuals (SD Ab.Res.) are used. The Sum Ab.Res. measures the total residuals over the validation subset. Since each model was validated using the same data consisting of the same number of software systems, the Sum Ab.Res. is equivalent to the mean of the absolute residuals when comparing the models. The Med.Ab.Res. is chosen for a measure of the central tendency of the residual distribution since the residual distribution is usually skewed with software data. The SD Ab.Res. measures the dispersion of the residual distribution.

MRE is a normalized measure of the discrepancy between actual values and predicted values, given by [16]:

\[
MRE = \frac{|\text{actual value} - \text{predicted value}|}{\text{actual value}}
\]  

In this study, the maximum value of MRE (Max.MRE) and the mean of MRE, the mean magnitude of relative error (MMRE), are used. The Max.MRE measures the maximum relative discrepancy, which is equivalent to the maximum error relative to the actual effort in the prediction. The MMRE measures the average relative discrepancy, which is equivalent to the average error relative to the actual effort in the prediction. The MMRE is given by:

\[
MMRE = \frac{1}{n} \sum_{i=1}^{n} MRE_i
\]
Sometimes MMRE is expressed in %. However, this study follows the definition above, and does not express MMRE in %.

Pred is a measure of what proportion of the predicted values have MRE less than or equal to a specified value, given by [4]:

\[ \text{Pred}(q) = \frac{k}{n} \]  

where \( q \) is the specified value, \( k \) is the number of systems whose MRE is less than or equal to \( q \), and \( n \) is the total number of systems in the dataset. In this study, \( \text{pred}(0.25) \) and \( \text{pred}(0.30) \) are used since these two pred measures are commonly used in the software effort prediction literature.

In order for an effort prediction model to be considered accurate, \( \text{MMRE} \leq 0.25 \) [3] and/or either \( \text{pred}(0.25) \geq 0.75 \) [3] or \( \text{pred}(0.30) \geq 0.70 \) [18] is suggested in the literature. On the other hand, there is a growing concern about MRE, since MRE is biased [32] and not always reliable as a predictive accuracy measure [5]. However, MRE has been the de facto standard in the software effort prediction literature and no alternative standard exists at present. Hence, MRE is still used in this study. However, this study also uses the absolute residual measures, since the absolute residual measures, in particular the SD Ab.Res., are shown to be a better measure than MRE for model comparison [5].

IV. CASE STUDY 1: DATA-CENTRED 4GL SYSTEMS

A. Software dataset

1) Description: In order to compare predictive accuracy of the models described in the previous section, a first case study was conducted using a dataset collected by the author. The dataset consists of 25 small or medium sized data-centred fourth-generation language (4GL) software systems, developed by a single organization. The data-centred 4GL systems are database-oriented transaction processing systems (TPS) and/or management information systems (MIS), developed using particular 4GL software development tools. The 4GL tools are Oracle Corporation’s Designer 6i and Developer 6i.

Each of the systems were developed by a team of final year undergraduates at a university in New Zealand in 2002 and 2003. Most of the individuals involved were taking Information Science or Computer Science as a major and the development was undertaken during the final
semester of their degree course, on behalf of business clients outside the university in a manner similar to commercial development. The development teams followed the software development methodology, taught in their course.

2) Variables: In order to build an accurate software effort prediction model, it is important to select predictor variables that are influential in the effort. It is also important to choose these variables from those available at an early stage of development, so that the benefit of the prediction can be maximized.

It is known that software effort depends on the size of the system. And a size metric available at the earliest stage of the development is a specification-based metric. Hence, this study chose a number of specification-based software size metrics for the models.

The specifications of the data-centred 4GL software systems described above were expressed in a form of Entity Relationship Diagram (ERD) and Function Hierarchy Diagram (FHD). The ERD was used to show the system’s database entities and their relationships. Under the software development methodology employed, the FHD was used to show the system’s data entry forms, summary reports, and how they were organized. Bearing these in mind, the following three specification-based software size metrics: FORMNUM, REPTNUM and ENTFORM were considered as candidate predictor variables.

The FORMNUM is the total number of data entry forms in a system, whereas REPTNUM is the total number of data summary reports produced by the system. The ENTFORM is the total number of times the system’s data entry forms access the database entities. If two different forms have access to the same entity, it is counted as two. The values of these variables are manually collected from the system’s ERD and FHD.

In addition to these variables, DNUM and SKILL were used as predictor variables. The DNUM indicates the number of individuals in a development team. The number of individuals involved in developing a software system is often influential in the effort since more individuals generally require more time for communicating. The effect of such communication overload can become statistically significant for certain software systems, including the systems used in this study, since the particular development methodology employed requires continual and intensive communication between all developers in a team.

The SKILL indicates the skill level of a development team regarding the particular 4GL tool used for the development. It is measured as the average of marks of the individual developers
in the team, awarded in a practical assessment carried out at the time of the development. In the assessment, each individual was asked to independently build an identical software system within a period of two hours using the same 4GL tools as the development. At the beginning of the assessment, the specifications of the system were given in a form of ERD and FHD to make the assessment environment as close as to that of the development. Because of these similarities, the average assessment mark is considered to appropriately reflect the team’s average skill level at the time of the development. The SKILL value is a ratio expressed in % by taking the full mark of the assessment as 100%.

The inclusion of SKILL is expected to improve predictive accuracy of the models, since software development effort is known to vary depending on the skill of developers. For example, it is said in the literature that the productivity of highly skilled developers could be up to 30 times higher than low-skilled developers [6].

The EFFORT variable measures the total effort spent by a development team in hours. That is, the sum of the effort spent by the individuals in the team. During the development, each team member was required to record their individual effort in hours each day into their own log book, under the regulations of their course. The EFFORT values were collected from these log books. The recorded individual effort was checked by the team leader for accuracy. Because their academic achievements in the course would not be influenced by the amount of recorded effort, the EFFORT values collected from these log books are considered reasonably accurate.

Descriptions of these variables are listed in Table I and their descriptive statistics are shown in Table II.

3) Variable selection method: In order to investigate if each of the above five candidate predictor variables is influential in the EFFORT, a series of statistical tests were performed. The procedure used is the F-test on the extra sum of squares. The F-test on the extra sum of squares is a standard procedure for linear statistical models to select important variables from multiple candidates. The F-test is carried out by comparing the extra sum of squares obtained from one variable to the corresponding residual sum of squares. The extra sum of squares is the sum of squares of the differences between the estimated and observed values remaining after subtracting the contributions of all the other variables in the model. The F statistic value is calculated by dividing the extra sum of squares by the residual sum of squares. If the F statistic value is statistically significant, the variable is identified as influential. The set of linear statistical
<table>
<thead>
<tr>
<th>Variable</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>FORMNUM</td>
<td>number of data entry forms</td>
</tr>
<tr>
<td>REPTNUM</td>
<td>number of data summary reports</td>
</tr>
<tr>
<td>ENTFORM</td>
<td>number of access made by the forms to database entities</td>
</tr>
<tr>
<td>DNUM</td>
<td>number of individuals in a development team</td>
</tr>
<tr>
<td>SKILL</td>
<td>average mark of four developers in a practical assessment</td>
</tr>
<tr>
<td>EFFORT</td>
<td>observed effort (hours)</td>
</tr>
</tbody>
</table>

**TABLE I**

**VARIABLES OF THE DATA-CENTRED 4GL SOFTWARE SYSTEMS**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Mean</th>
<th>Median</th>
<th>Std Dev.</th>
<th>Min.</th>
<th>Max.</th>
</tr>
</thead>
<tbody>
<tr>
<td>FORMNUM</td>
<td>12.16</td>
<td>11</td>
<td>3.63</td>
<td>7</td>
<td>20</td>
</tr>
<tr>
<td>REPTNUM</td>
<td>6.88</td>
<td>8</td>
<td>3.35</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>ENTFORM</td>
<td>31.64</td>
<td>32</td>
<td>12.01</td>
<td>11</td>
<td>62</td>
</tr>
<tr>
<td>DNUM</td>
<td>4.24</td>
<td>4</td>
<td>0.60</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>SKILL</td>
<td>61.15</td>
<td>60.4</td>
<td>9.67</td>
<td>42.5</td>
<td>78.6</td>
</tr>
<tr>
<td>EFFORT</td>
<td>463.0</td>
<td>432.3</td>
<td>138.55</td>
<td>258.2</td>
<td>757.5</td>
</tr>
</tbody>
</table>

**TABLE II**

**DESCRIPTIVE STATISTICS**

models analyzed here comprises all models consisting of a subset of the five predictor variables and their two-way interactions. The software package used is SPSS11.0.

The two-way interactions between certain pairs of the variables are included, since these interactions may influence effort, even though each individual variable alone may not. The interactions tested are: SKILL*FORMNUM, SKILL*REPTNUM, SKILL*ENTFORM and SKILL*DNUM. These interactions take account of the possibility that the effect of SKILL may differ according to
the values of other predictor variables. For example, a statistically significant \textit{SKILL*FORMNUM} value indicates that the effect of \textit{SKILL} on \textit{EFFORT} changes as the \textit{FORMNUM} value increases. Since the \textit{FORMNUM} measures the size of the system, this means that the effect of \textit{SKILL} changes as the system becomes larger.

The results of the analysis are shown in Table III. The models in this table are compared in sequential F-tests, starting with the full model that consists of the five individual predictors and four two-way interactions. The first column indicates the number of steps taken to select the most appropriate model. In each step, only the least statistically significant variable, emphasized in italics, is removed from the model selected in the previous step. The 5\% level of significance is used to identify the final model.

Table III leads us to conclude that the most appropriate model consists of four predictors: \textit{ENTFORM}, \textit{SKILL}, \textit{DNUM} and \textit{SKILL*ENTFORM}. The final model requires not only \textit{DNUM} and \textit{SKILL*ENTFORM} but also \textit{ENTFORM} and \textit{SKILL}, since an interaction with the two individual variables involved must always coexist.

The above F-tests are often performed in multiple regression to select influential predictors in the predicted variable. Alternatively, Bayesian statistics offers a number of different variable selection procedures to identify important variables for the models. This study used one of these Bayesian variable selection procedures to see if it produces the same result as that of the above F-tests. The particular procedure used is based on Deviance Information Criterion (DIC). This was chosen because the Bayesian statistical models in this study use flat prior probability distributions. Two other frequently used Bayesian criteria: Bayes Factor and Bayesian Information Criterion, are not appropriate for flat prior distributions. The DIC is the Bayesian statistical equivalent to the Akaike Information Criterion (AIC). That is, the optimal model is selected from a set of candidate models, by choosing the DIC value closest to 0. This study calculated the DIC value of a model using a function provided by WinBUGS.

The results are shown in Table IV. The models in this table are compared, starting with the full model consisting of the five individual predictors and four two-way interactions. The first column indicates the number of steps taken for selecting the optimal model. In each step, the model with the smallest DIC value is selected. The selected model is emphasized in italics. The procedure continues until all models in a step have a DIC value greater than that of the model selected in the previous step. This means that the optimal model is the model selected in the
<table>
<thead>
<tr>
<th>Step</th>
<th>Variable</th>
<th>P-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><strong>SKILL*FORMNUM</strong></td>
<td>0.991</td>
</tr>
<tr>
<td></td>
<td>SKILL*REPTNUM</td>
<td>0.554</td>
</tr>
<tr>
<td></td>
<td>SKILL*ENTFORM</td>
<td>0.258</td>
</tr>
<tr>
<td></td>
<td>SKILL*DNUM</td>
<td>0.790</td>
</tr>
<tr>
<td>2</td>
<td>SKILL*REPTNUM</td>
<td>0.502</td>
</tr>
<tr>
<td></td>
<td>SKILL*ENTFORM</td>
<td>0.105</td>
</tr>
<tr>
<td></td>
<td><strong>SKILL*DNUM</strong></td>
<td>0.775</td>
</tr>
<tr>
<td></td>
<td>FORMNUM</td>
<td>0.499</td>
</tr>
<tr>
<td>3</td>
<td>SKILL*REPTNUM</td>
<td>0.469</td>
</tr>
<tr>
<td></td>
<td>SKILL*ENTFORM</td>
<td>0.092</td>
</tr>
<tr>
<td></td>
<td><strong>FORMNUM</strong></td>
<td>0.486</td>
</tr>
<tr>
<td></td>
<td>DNUM</td>
<td>0.001</td>
</tr>
<tr>
<td>4</td>
<td><strong>SKILL*REPTNUM</strong></td>
<td>0.466</td>
</tr>
<tr>
<td></td>
<td>SKILL*ENTFORM</td>
<td>0.094</td>
</tr>
<tr>
<td></td>
<td>DNUM</td>
<td>0.000</td>
</tr>
<tr>
<td>5</td>
<td>SKILL*ENTFORM</td>
<td>0.074</td>
</tr>
<tr>
<td></td>
<td>DNUM</td>
<td>0.000</td>
</tr>
<tr>
<td></td>
<td><strong>REPTNUM</strong></td>
<td>0.589</td>
</tr>
<tr>
<td>6</td>
<td>SKILL*ENTFORM</td>
<td>0.041</td>
</tr>
<tr>
<td></td>
<td>DNUM</td>
<td>0.000</td>
</tr>
</tbody>
</table>

**TABLE III**

*Variable Selection based on F-tests*

second to last step.

Table IV shows that the optimal model is Model 18, which consists of ENTFORM, SKILL, DNUM and SKILL*ENTFORM. This result is identical to the one obtained from the F-tests on
<table>
<thead>
<tr>
<th>Step</th>
<th>Model</th>
<th>DIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Full model</td>
<td>1019.480</td>
</tr>
<tr>
<td>1</td>
<td>1: Full model - <em>SKILL</em>FORMNUM</td>
<td>1016.780</td>
</tr>
<tr>
<td></td>
<td>2: Full model - <em>SKILL</em>REPTNUM</td>
<td>1017.360</td>
</tr>
<tr>
<td></td>
<td>3: Full model - <em>SKILL</em>ENTFORM</td>
<td>1018.870</td>
</tr>
<tr>
<td></td>
<td>4: Full model - <em>SKILL</em>DNUM</td>
<td>1016.830</td>
</tr>
<tr>
<td>2</td>
<td>5: Model 1 - <em>SKILL</em>REPTNUM</td>
<td>1014.850</td>
</tr>
<tr>
<td></td>
<td>6: Model 1 - <em>SKILL</em>ENTFORM</td>
<td>1018.310</td>
</tr>
<tr>
<td></td>
<td>7: Model 1 - <em>SKILL</em>DNUM</td>
<td>1014.230</td>
</tr>
<tr>
<td></td>
<td>8: Model 1 - FORMNUM</td>
<td>1014.940</td>
</tr>
<tr>
<td>3</td>
<td>9: Model 7 - <em>SKILL</em>REPTNUM</td>
<td>1012.480</td>
</tr>
<tr>
<td></td>
<td>10: Model 7 - <em>SKILL</em>ENTFORM</td>
<td>1027.550</td>
</tr>
<tr>
<td></td>
<td>11: Model 7 - FORMNUM</td>
<td>1012.490</td>
</tr>
<tr>
<td></td>
<td>12: Model 7 - DNUM</td>
<td>1029.160</td>
</tr>
<tr>
<td>4</td>
<td>13: Model 9 - <em>SKILL</em>ENTFORM</td>
<td>1014.730</td>
</tr>
<tr>
<td></td>
<td>14: Model 9 - FORMNUM</td>
<td>1010.760</td>
</tr>
<tr>
<td></td>
<td>15: Model 9 - REPTNUM</td>
<td>1010.540</td>
</tr>
<tr>
<td></td>
<td>16: Model 9 - DNUM</td>
<td>1027.810</td>
</tr>
<tr>
<td>5</td>
<td>17: Model 15 - <em>SKILL</em>ENTFORM</td>
<td>1014.020</td>
</tr>
<tr>
<td></td>
<td>18: Model 15 - FORMNUM</td>
<td>1008.840</td>
</tr>
<tr>
<td></td>
<td>19: Model 15 - DNUM</td>
<td>1028.900</td>
</tr>
<tr>
<td>6</td>
<td>20: Model 18 - <em>SKILL</em>ENTFORM</td>
<td>1013.340</td>
</tr>
<tr>
<td></td>
<td>21: Model 18 - DNUM</td>
<td>1025.560</td>
</tr>
</tbody>
</table>

**TABLE IV**

**Variable Selection based on DIC**
the extra sum of squares.

Once the optimal model has been selected, the model’s goodness-of-fit to given data is also assessed statistically using a Bayesian goodness-of-fit test. This test ensures that the selected model appropriately fits the data. Pearson’s test statistic $T$, as shown below, was used as a test statistic.

$$T = \sum_{i=1}^{k} \frac{(d_i - e_i)^2}{e_i}$$

(33)

where $d_i$ indicates the observed value in data, and $e_i$ indicates its expected value calculated from the model. Then, the associated p-value was calculated directly from the above statistic’s probability distribution, by drawing a large number of samples from the distribution using WinBUGS.

The results show that the Bayesian p-value of the BMVN model obtained from the 25 software systems is 0.458, and of the BMVR model is 0.544. These p-values confirm that both the BMVN and BMVR models appropriately fit the data.

Since the above two different variable selection procedures have identified the identical model consisting of ENTFORM, SKILL, DNUM and SKILL*ENTFORM as the most appropriate model; and since appropriate goodness-of-fit of the Bayesian statistical models consisting of these same variables has been assured, only these four variables are included in the models to predict the effort for the target system in the reminder of this section.

4) Validation method: Predictive accuracy of the models was evaluated using four different-sized subsets. These subsets were randomly sampled from 25 systems in the original dataset without replacement, using a function provided by SPSS11.0.

The first and largest subset was the original dataset. Since the dataset consisted of only 25 software systems, a leave-one-out validation method was applied to this subset. The leave-one-out validation allows predictive accuracy of the model to be evaluated using the largest possible calibration subset. Using this method, each model was calibrated using 24 systems sampled from the dataset. The model then predicted the effort of the one remaining system. This procedure was repeated 25 times, until the effort of all 25 systems in the dataset had been predicted. The model’s predictive accuracy was then evaluated as the average accuracy of these 25 systems.

The second subset consisted of six software systems. Eight samples of this subset were independently drawn from the original dataset. Each remaining group of 19 systems formed
the corresponding validation sample, from which predictive accuracy of the calibrated model was evaluated. Eight independent samples instead of one were used to reduce the possibility of obtaining a particular result by chance, as well as increasing the number of systems used in the model’s evaluation. That is, the model’s predictive accuracy was evaluated from each of the eight validation samples, resulting in a total of $19 \times 8 = 152$ systems for the evaluation. Predictive accuracy of the model was then calculated as the average value of these 152 systems.

The third subset consisted of four systems. Again eight independent samples were drawn. Then, each remaining group of 21 systems formed the corresponding validation sample, resulting in evaluating the model’s predictive accuracy using $21 \times 8 = 168$ systems.

The fourth and smallest subset consisted of two systems. Like the previous subsets, eight independent samples were drawn. Then, each remaining group of 23 systems formed the corresponding validation sample, so that the model’s predictive accuracy was evaluated using $23 \times 8 = 184$ systems.

B. Model evaluation

1) Results from the leave-one-out validation of 25 systems: Table V shows the results obtained from the leave-one-out validation of the original dataset of 25 systems. As was explained in the previous subsection, these values are the average accuracy calculated from the 25 systems. The models were calibrated using 24 systems.

The CBR model is not included in this table, since it is too labour-intensive to find the appropriate number of nearest systems for predicting each of the 25 system. As was explained in subsection II-C, finding the number from a calibration subset of 24 systems requires $24 \times (24 - 1) = 552$ trials. Since the leave-one-out validation uses 25 different calibration subsets of 24 systems, a total of $25 \times 552 = 13800$ trials would be required simply to find the number here, which is not really practical.

Table V shows that the accuracy of the BMVR model is slightly lower than the BMVN and regression models, whereas the accuracy of the BMVN and regression models is similar. Wilcoxon signed-rank tests performed on the MRE values and absolute residuals have confirmed evidence of a statistically significant difference between the BMVR and other two models, however, no such difference between the BMVN and regression models. Hence, it is concluded
that both the BMVN and regression models are able to predict software development effort better than the BMVR model when calibrated using 24 systems in this dataset.

However, the MMRE, $\text{pred}(0.25)$ and $\text{pred}(0.30)$ values achieved by the three models all satisfy the criteria for an accurate model suggested in the literature: $\text{MMRE} \leq 0.25$, $\text{pred}(0.25) \geq 0.75$ and $\text{pred}(0.30) \geq 0.70$. Hence, according to these criteria, even the least accurate BMVR model is concluded to be sufficient in practice.

2) Results from eight calibration samples of six systems: Table VI shows the results obtained from the eight independent calibration samples of six systems. As explained in the previous subsection, the values in this table are the average accuracy calculated from 152 systems, in the corresponding eight validation samples.

In Table VI the accuracy of the first three models is lower than that in the previous table. This accuracy deterioration is not surprising since the number of systems used in the calibration is now reduced from 24 to six. Consequently, the models no longer satisfy the suggested criteria for an accurate model.

Table VI also shows that the accuracy of the four models is different. Using the 5% level of significance, Wilcoxon signed-rank tests confirmed a statistically significant difference between each pair of models. This means that the accuracy of the BMVR model is now greater than that of the BMVN and regression models, when calibrated using the eight samples of six software systems. However, it is not as good as that of the CBR model. The table also shows that the BMVN model has greater accuracy than the regression model, however, less than the CBR

<table>
<thead>
<tr>
<th>Model</th>
<th>MRE</th>
<th>Pred</th>
<th>Ab.Res.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Max.</td>
<td>Mean</td>
<td>0.25</td>
</tr>
<tr>
<td>BMVN</td>
<td>0.590</td>
<td>0.170</td>
<td>0.840</td>
</tr>
<tr>
<td>BMVR</td>
<td>0.600</td>
<td>0.187</td>
<td>0.760</td>
</tr>
<tr>
<td>R</td>
<td>0.640</td>
<td>0.167</td>
<td>0.840</td>
</tr>
</tbody>
</table>

TABLE V

Predictive Accuracy from Leave-one-out Validation of 25 Systems

2) Results from eight calibration samples of six systems: Table VI shows the results obtained from the eight independent calibration samples of six systems. As explained in the previous subsection, the values in this table are the average accuracy calculated from 152 systems, in the corresponding eight validation samples.

In Table VI the accuracy of the first three models is lower than that in the previous table. This accuracy deterioration is not surprising since the number of systems used in the calibration is now reduced from 24 to six. Consequently, the models no longer satisfy the suggested criteria for an accurate model.

Table VI also shows that the accuracy of the four models is different. Using the 5% level of significance, Wilcoxon signed-rank tests confirmed a statistically significant difference between each pair of models. This means that the accuracy of the BMVR model is now greater than that of the BMVN and regression models, when calibrated using the eight samples of six software systems. However, it is not as good as that of the CBR model. The table also shows that the BMVN model has greater accuracy than the regression model, however, less than the CBR
model.

3) Results from eight calibration samples of four systems: Table VII shows the results obtained from the eight independent samples of four systems. The values in this table are the average accuracy calculated from 168 systems.

The RBMVR model has replaced the BMVR model in this table, since the number of systems in the calibration sample is now the same as the number of predictors. The regression model is no longer included, since a regression model with four predictors cannot be applied to a dataset consisting of fewer than six observations.

In Table VII the accuracy of the BMVN and CBR models is generally lower than that in
Table VI. The fewer systems in the calibration samples explains this deterioration. However, the accuracy of the RBMVR model seems very close to that of the BMVR model in Table VI, despite the fewer systems in the calibration data. Consequently, the accuracy of the RBMVR model is now not only greater than that of the BMVN model, but also approaches that of the CBR model.

Using the 5% level of significance, Wilcoxon signed-rank tests confirmed statistically significant differences of the BMVN model from the others, but no such difference was confirmed between the RBMVR and CBR models. Hence, it is concluded that the RBMVR model is able to predict the effort of the target system more accurately than the BMVN model, and as accurately as the CBR model, when calibrated using the eight samples of four systems.

4) Results from eight calibration samples of two systems: Table VIII shows the results obtained from the eight independent samples of two systems. The values in this table are the average accuracy of 184 systems.

This table includes neither the BMVN model nor the regression model, as the results in Tables VI and VII have provided sufficient evidence that the BMVR and RBMVR models are more accurate than the BMVN model, when calibrated using fewer than 10 systems. The regression model cannot be applied to these samples, since these samples have only two systems for the calibration.

In Table VIII the accuracy of the RBMVR and CBR models is lower than that in Table VII, due to the fewer systems in the calibration data. However, the accuracy deteriorates less with the RBMVR model than with the CBR model. Consequently, the accuracy of the RBMVR model
now becomes slightly greater than that of the CBR model, although the Wilcoxon signed-rank test confirms no statistically significant difference between the two models. It is therefore concluded that the RBMVR model is able to predict the effort of the target system at least as accurately as the CBR model, when calibrated using only two systems.

C. Discussion

The above results show that the two Bayesian statistical models and the multiple linear regression models are accurate in predicting the effort of data-centred 4GL software systems studied, provided that the calibration subset consists of 24 systems. However, when the calibration subset consists of as few as six systems, these three models have achieved lower accuracy and no longer meet the criteria for accuracy suggested in the literature. A further reduction of systems in the calibration data, in general, has resulted in further accuracy deterioration. Similar consecutive accuracy deterioration was observed in the CBR models when the calibration data was reduced from six systems. These results thus confirm that the predictive accuracy of the models studied generally deteriorates, when the calibration dataset consists of fewer systems.

The rate and level of the observed accuracy deterioration vary between the models. The accuracy of the BMVR model is initially not as good as that of the multiple linear regression model, when calibrated using 24 systems, whereas the accuracy of the BMVN model is equivalent to that of the regression model. However, when the calibration subset is reduced to as few as six systems, the accuracy of the BMVR and BMVN models becomes greater than that of the regression model. This suggests that the Bayesian statistical models are able to predict the effort of software systems studied more accurately than the regression model, when the calibration dataset consists of fewer systems.

In addition, the accuracy of the BMVN and BMVR models is not as good as that of the CBR model, when calibrated using the eight samples of six systems. However, when the calibration subset is reduced to four systems, the accuracy of the RBMVR model becomes closer to that of the CBR model. Then, as the calibration subset is further reduced to two systems, the accuracy of the RBMVR model becomes greater than that of the CBR model, although the difference is not identified as statistically significant. This indicates that the accuracy deterioration of the RBMVR model is smaller than that of the CBR model, when the size of the calibration dataset decreases from six systems.
When compared to each other, the BMVN model initially outperforms the BMVR model when calibrated using 24 systems. However, when the calibration dataset is reduced to six systems and further, the BMVR and RBMVR models outperform the BMVN model. The BMVN model assumes correlations between predictor variables, whereas the BMVR and RBMVR models not only assume the correlations but also incorporate the correlations into a linear relationship with the effort. Hence, the better accuracy of the BMVR and RBMVR models over the BMVN model may suggest that when the calibration dataset consists of fewer systems, software development effort can be more accurately predicted by incorporating correlations among predictors into their linear relationship, rather than the linear relationship alone.

V. CASE STUDY 2: DESHARNAIS SYSTEMS

A. Software dataset

1) Description: The second case study was conducted to predict effort for another type of software systems, developed by another organization. The dataset used is known as the Desharnais dataset. The Desharnais dataset is in the public domain and available on line from Promise Software Engineering Repository [27]. It consists of 77 software systems, which were developed by a Canadian software house between 1983 and 1988. From these 77 systems, this study used a subset of 44 systems that were developed using the same programming language. The subset of 44 systems was chosen in order to create a local dataset consisting only of similar software systems. This subset is referred to as the original dataset in this paper.

2) Variables: The original dataset contains the observed software development effort of the 44 systems and the following seven variables:

- Experience of equipment (years): \texttt{EXEQP}
- Experience of project manager (years): \texttt{EXMAN}
- Number of transactions: \texttt{TRANS}
- Adjusted Function Points: \texttt{ADFP}
- Number of entities: \texttt{ENT}
- Duration (months)
- Year finished (categorical variable: 83 \textasciitilde 88)

From these seven variables, this study selected the first five as candidate predictor variables for the models. This is because the values of the last two variables: Duration and Year finished,


<table>
<thead>
<tr>
<th>Variable</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXEQP</td>
<td>experience of equipment (years)</td>
</tr>
<tr>
<td>EXMAN</td>
<td>experience of project manager (years)</td>
</tr>
<tr>
<td>TRANS</td>
<td>number of transactions</td>
</tr>
<tr>
<td>ADFP</td>
<td>adjusted Function Points</td>
</tr>
<tr>
<td>ENT</td>
<td>number of entities</td>
</tr>
<tr>
<td>EFFORT</td>
<td>observed effort (hours)</td>
</tr>
</tbody>
</table>

**TABLE IX**

**VARIABLES OF DESHARNAIS SOFTWARE SYSTEMS**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Mean</th>
<th>Median</th>
<th>Std Dev.</th>
<th>Min.</th>
<th>Max.</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXEQP</td>
<td>2.43</td>
<td>2</td>
<td>1.39</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>EXMAN</td>
<td>2.30</td>
<td>2.5</td>
<td>1.59</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>TRANS</td>
<td>169.52</td>
<td>143.0</td>
<td>143.43</td>
<td>33</td>
<td>886</td>
</tr>
<tr>
<td>ADFP</td>
<td>277.91</td>
<td>259.0</td>
<td>179.73</td>
<td>83</td>
<td>1116</td>
</tr>
<tr>
<td>ENT</td>
<td>118.30</td>
<td>97.5</td>
<td>77.43</td>
<td>7</td>
<td>332</td>
</tr>
<tr>
<td>EFFORT</td>
<td>5413.00</td>
<td>3993.5</td>
<td>4365.73</td>
<td>805</td>
<td>23940</td>
</tr>
</tbody>
</table>

**TABLE X**

**DESCRIPTIVE STATISTICS**

become available only after the system has been completed, that is, at the same time as the effort.

The observed effort values are measured in hours. Descriptions of the six selected variables and their descriptive statistics are shown in Tables IX and X.

Since the distributions of TRANS, ADFP, ENT and EFFORT appear skewed, the values of these four variables were log-transformed into LogTRANS, LogADFP, LogENT and LogEF-
FORT for the Bayesian statistical models and multiple linear regression models. This is because these models assume that data is approximately normally distributed, and hence, a significant departure from this assumption results in poor predictive accuracy. The log-transformation is frequently used to bring skewed data closer to the normal distribution.

However, in order to see if data is skewed or not, the dataset requires a sufficient number of systems. If the dataset consists only of fewer than 10 systems, whether the data is skewed or not may not always be apparent. In addition, the log-transformation of such a small number of systems may not significantly improve the predictive accuracy of the models. Bearing this in mind, this study constructed Bayesian statistical models and multiple linear regression models using the LogTRANS, LogADFP, LogENT and LogEFFORT values, as well as using the TRANS, ADFP, ENT and EFFORT values. Predictive accuracy of these models is then compared to determine whether the log-transformation leads to better accuracy. The -L notation is added to the models in the tables in this section to indicate the use of the log-transformed data. For example, the RBMVR-L model means that the RBMVR model is applied to the EXEQP, EXMAN, LogTRANS, LogADFP, LogENT and LogEFFORT values.

3) Variable selection method: In this second case study, from the above five candidate predictor variables each effort prediction model further selected an optimal set of important predictors by applying their own variable selection procedure. In addition, the variable selection procedure was performed separately for each calibration subset. Since each calibration subset consists of different systems, this leads to the optimal predictors for one calibration subset often differ from those for another subset in each model. Although this procedure is different from that used in the first case study, in which each model includes the same four predictor variables, this procedure is considered closer to the reality. This is because different predictive models offer different variable selection procedures, often leading to identifying a different set of important variables using the identical data. However, software practitioners and project managers cannot know which model is most accurate until the development is completed and the actual effort is obtained.

The Bayesian statistical models used a Bayesian variable selection procedure based on the DIC. The procedure is the same as that in the first case study. The goodness-of-fit of each optimal model was then assured by the test using the Bayesian p-value in the same way as the first case study.
The multiple linear regression models used two different variable selection procedures provided by SPSS11.0. One is backward elimination, and the other, stepwise selection. Although these two procedures use the F-test on the extra sum of squares, they apply the F-tests in a different way, often resulting in a different set of important variables. Hence, when these two procedures resulted in different models, this study compared predictive accuracy of the models, and selected the most accurate model as the optimal model.

The CBR model first identified the most appropriate number of nearest software systems to be used for prediction using the procedure described in II-C. Then, the Best Attribute Subset function provided by AngelPlus was used to select the optimal set of predictors.

4) Validation method: Predictive accuracy of the models was evaluated using five different-sized subsets.

The first and largest subset was the original dataset. A leave-one-out validation method was applied to this subset. That is, each model was calibrated using 43 systems sampled from the dataset. Then, the model predicted the effort of the one remaining system. This procedure was repeated 44 times, until the effort of all 44 systems in the dataset had been predicted. Predictive accuracy of the model was evaluated as the average accuracy of these 44 systems.

The second subset consisted of 11 systems. The original dataset of 44 systems was randomly divided into four samples of this subset, using a function provided by SPSS11.0. This splitting ensures that these four samples completely cover the original dataset, allowing each system in the original dataset to be used for the model’s calibration. When a model was calibrated using 11 systems in one sample, the remaining group of 33 systems formed the corresponding validation sample. This process was repeated four times using a different set of calibration and validation samples, so that predictive accuracy of the model was evaluated as the average value of $33 \times 4 = 132$ systems.

The third subset consisted of eight systems. First, the original dataset of 44 systems was randomly divided into six samples of this subset, using a function provided by SPSS11.0. Then, each of the four samples with only seven systems added one more system, different from those already included in the sample. This last system was randomly drawn from the 37 systems not included in the sample, using a function provided by SPSS11.0. This process ensures that these six samples completely cover the original dataset, allowing each system in the original dataset to be used at least once for the model’s calibration. When a model was calibrated using eight
systems in one sample, the remaining group of 36 systems formed the corresponding validation sample. This process was repeated six times, so that predictive accuracy of the model was evaluated as the average value of $36 \times 6 = 216$ systems.

The fourth subset consisted of five systems. Like the previous subset, the original dataset of 44 systems was first randomly divided into nine samples of this subset, using a function provided by SPSS11.0. After that, the one sample with only four systems added one more system, different from those already included in the sample. This last system was randomly drawn from the 40 systems not included in the sample, using a function provided by SPSS11.0. This process again ensures that these nine samples completely cover the original dataset, allowing each system in the original dataset to be used at least once for the model’s calibration. Once a model was calibrated using five systems in one sample, the remaining group of 39 systems formed the corresponding validation sample, allowing the model’s predictive accuracy to be evaluated using $39 \times 9 = 351$ systems.

The final subset consisted of two systems. The original dataset of 44 systems was randomly divided into 22 samples of this subset, using a function provided by SPSS11.0. This splitting again ensures that these 22 samples completely cover the original dataset, allowing each system in the original dataset to be used for the model’s calibration. When a model was calibrated using two systems in one sample, the remaining group of 42 systems formed the corresponding validation sample. This resulted in predictive accuracy of the model being evaluated as the average value of $42 \times 22 = 924$ systems.

B. Model evaluation

1) Results from the leave-one-out validation of 44 systems: Table XI shows the results obtained from the leave-one-out validation of the original dataset of 44 systems. As was explained in the previous subsection, these values are the average accuracy calculated from the 44 systems. The models were calibrated using 43 systems.

The BMVN model is no longer used in this case study since the previous case study has provided sufficient evidence that the BMVN model is not as accurate as the BMVR and RBMVR models when the calibration dataset consists of fewer systems. In addition, the BMVR model is completely replaced with the RBMVR model since the RBMVR model is more stable than the BMVR model, particularly when the calibration dataset consists of a smaller number of software
systems than the number of predictor variables.

The CBR model is not included in Table XI, since it is too labour-intensive to find the appropriate number of nearest systems for predicting each of the 44 system.

Table XI shows that the accuracy of the RBMVR and RBMVR-L models is slightly lower than that of the respective R and R-L models. In order to see if these small differences are statistically significant, Wilcoxon signed-rank tests were performed on the MRE values and absolute residuals. Using the 5% level of significance, the results confirmed a statistically significant difference between the RBMVR and R models, and the RBMVR-L and R-L models. Hence, it is concluded that the Bayesian statistical model is not able to predict the effort of the target system as accurately as the multiple linear regression model, when the calibration dataset consists of 43 systems.

Table XI also shows that the accuracy of the RBMVR and R models is improved by the log-transformed data, as indicated by the greater accuracy of the respective RBMVR-L and R-L models. The Wilcoxon signed-rank tests confirmed these improvements as statistically significant.

2) Results from four calibration samples of 11 systems: Table XII shows the results obtained from the four calibration samples of 11 systems. As was explained in the previous subsection, the values in this table are the average accuracy calculated from 132 systems.

In Table XII the accuracy of the Bayesian statistical models and the multiple linear regression models is lower than that obtained from the leave-one-out validation of 43 systems in the previous

<table>
<thead>
<tr>
<th>Model</th>
<th>MRE</th>
<th>Pred</th>
<th>Ab.Res.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Max.</td>
<td>Mean</td>
<td>0.25</td>
</tr>
<tr>
<td>RBMVR</td>
<td>1.353</td>
<td>0.414</td>
<td>0.455</td>
</tr>
<tr>
<td>RBMVR-L</td>
<td>1.237</td>
<td>0.383</td>
<td>0.477</td>
</tr>
<tr>
<td>R</td>
<td>1.255</td>
<td>0.391</td>
<td>0.432</td>
</tr>
<tr>
<td>R-L</td>
<td>1.036</td>
<td>0.355</td>
<td>0.477</td>
</tr>
</tbody>
</table>

TABLE XI

PREDICTIVE ACCURACY FROM LEAVE-ONE-OUT VALIDATION OF 44 SYSTEMS
This accuracy deterioration is explained by the reduction in the number of systems used in the calibration.

Table XII also shows that the accuracy of the RBMVR and RBMVR-L models is lower than that of the R and R-L models. Using the 5% level of significance, the Wilcoxon signed-rank tests confirmed a statistically significant difference between the RBMVR-L and R-L models, but no such difference between the RBMVR and R models. This result suggests that the optimal RBMVR model is able to predict the effort of the target system as accurately as the optimal multiple linear regression model when calibrated using 11 systems. However, when the data is log-transformed, although the improvement is confirmed as statistically significant by Wilcoxon signed-rank tests, the improvement of the RBMVR model is not as good as that of the regression model, resulting in the lower accuracy of the RBMVR-L model than that of the R-L model.

The accuracy of the RBMVR and RBMVR-L models is also lower than that of the CBR model. However, the Wilcoxon signed-rank tests confirmed no statistically significant difference between the RBMVR and CBR models, or the RBMVR-L and CBR models. Hence, it is concluded that the optimal Bayesian statistical models are able to predict the effort of the target system as accurately as the optimal CBR model, when the calibration dataset consists of 11 software systems.
3) Results from six calibration samples of eight systems: Table XIII shows the results obtained from the six calibration samples of eight systems. The values in this table are the average accuracy calculated from 216 systems.

In Table XIII the accuracy of the R, R-L and CBR models is lower than that in Table XII. Fewer systems in the calibration samples explains this accuracy deterioration.

In contrast, the accuracy of the RBMVR and RBMVR-L models in Table XIII is generally greater than that in Table XII, despite the fewer systems in the calibration samples. One possible explanation of this counter-intuitive result is the fact that the predictive accuracy in Table XIII is the average of 216 systems, in comparison to 132 systems in Table XII. The average value of 132 systems can vary more than the average of 216 systems. On the other hand, the accuracy deterioration of the Bayesian statistical models caused by the reduction of the systems from 11 to eight may be very small. If the deterioration is smaller than the variation in the average values, it is possible to observe greater accuracy, as shown in Table XIII.

Table XIII also shows that the accuracy of the RBMVR and RBMVR-L models is now greater than that of the R and R-L models. Using the 5% level of significance, the Wilcoxon signed-rank tests confirmed a statistically significant difference between the RBMVR and R models, and the RBMVR-L and R-L models. Hence, it is concluded that the optimal Bayesian statistical models are able to predict the effort of the target system more accurately than the optimal multiple linear
The accuracy of the RBMVR and RBMVR-L models is now also greater than that of the CBR model. The Wilcoxon signed-rank tests confirmed a statistically significant difference between the RBMVR and CBR models, and the RBMVR-L and CBR models. Hence, it is concluded that the optimal Bayesian statistical models are able to predict the effort of the target system more accurately than the optimal CBR model, when the size of the calibration dataset is reduced to eight systems.

In addition, in Table XIII the accuracy of the RBMVR and R models is again generally improved by the log-transformed data, although the values of pred measures and Ab.Res. measures of the RBMVR-L model indicate otherwise. The Wilcoxon signed-rank tests confirmed these differences as statistically significant. Hence, it is again confirmed that the predictive accuracy of both the optimal Bayesian statistical model and the optimal regression model generally improves, if the calibration data of eight systems is log-transformed.

4) Results from nine calibration samples of five systems: Table XIV shows the results obtained from the nine calibration samples of five systems. The values in this table are the average accuracy calculated from 351 systems.

This table includes neither the R nor R-L model since a regression model with five candidate predictors cannot be applied to a calibration dataset consisting of only five observations.

In Table XIV the accuracy of all models is generally lower than that in Table XIII, due to fewer systems in the calibration data. The counter-intuitive result of the Bayesian statistical models is
no longer observed in this table. This seemingly inconsistent result supports the explanation that the variation in the average values is the likely cause.

Table XIV also shows that the accuracy of the RBMVR-L model is again greater than that of the CBR model. Although the accuracy of the RBMVR model is very similar to that of the CBR model, the log-transformed data has substantially improved the accuracy of the RBMVR-L model. The Wilcoxon signed-rank tests confirmed statistically significant differences between the RBMVR and RBMVR-L models, and between the RBMVR-L and CBR models, but no such difference between the RBMVR and CBR models. Hence, these results lead us to conclude that the optimal Bayesian statistical model is able to predict the effort of the target system as accurately as the optimal CBR model, when calibrated using only five systems. However, if the data is log-transformed, the predictive accuracy of the optimal Bayesian statistical model improves. Consequently, the optimal Bayesian statistical model becomes more accurate than the optimal CBR model in predicting the effort of the target system.

5) Results from 22 calibration samples of two systems: Table XV shows the results obtained from the 22 calibration samples of two systems. The values in this table are the average accuracy calculated from 924 systems.

Neither the R nor R-L model is included in this table, since a regression model with five candidate predictors cannot be applied to a calibration dataset of only two systems.

In Table XV the accuracy of all models is lower than that in Table XIV, due to fewer systems in the calibration data. The counter-intuitive result of the Bayesian statistical models is not

<table>
<thead>
<tr>
<th>Model</th>
<th>MRE</th>
<th>Pred</th>
<th>Ab.Res.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Max.</td>
<td>Mean</td>
<td>0.25</td>
</tr>
<tr>
<td>RBMVR</td>
<td>5.696</td>
<td>0.987</td>
<td>0.242</td>
</tr>
<tr>
<td>RBMVR-L</td>
<td>9.694</td>
<td>1.116</td>
<td>0.260</td>
</tr>
<tr>
<td>CBR</td>
<td>6.414</td>
<td>1.098</td>
<td>0.224</td>
</tr>
</tbody>
</table>

**TABLE XV**

**Predictive Accuracy from 22 Calibration Samples of Two Systems**
observed in this table, either.

Table XV also shows that the accuracy of the RBMVR model is now greater than that of the CBR model. On the other hand, the accuracy of the RBMVR-L model is very similar to that of the CBR model. This means that the log-transformed data has decreased the predictive accuracy of the RBMVR model, instead of improving it. The Wilcoxon signed-rank tests confirmed a statistically significant difference between the RBMVR and CBR models, the RBMVR-L and CBR models, and the RBMVR and RBMVR-L models. Hence, it is concluded that the optimal Bayesian statistical model is able to predict the effort of the target system more accurately than the optimal CBR model, when calibrated using only two systems.

However, if the data is log-transformed, the predictive accuracy of the optimal Bayesian statistical model deteriorates. Consequently, the optimal Bayesian statistical model is no longer able to predict the effort of the target system more accurately than the optimal CBR model.

C. Discussion

The above results show that predictive accuracy of all the models studied generally deteriorates when the calibration dataset consists of fewer systems.

However, the rate and level of the observed accuracy deterioration vary between the models, as was observed in the first case study. The predictive accuracy of the optimal RBMVR and RBMVR-L models is initially not as good as that of the optimal multiple linear regression model, when calibrated using as few as 11 systems. However, when the size of the calibration dataset is further reduced to eight systems, the accuracy of the RBMVR and RBMVR-L models becomes greater than that of the regression model. This finding is consistent with the finding of the previous case study. Thus, this study provides further evidence that the optimal Bayesian statistical model is able to predict the effort of software systems more accurately than the optimal multiple linear regression model, when calibrated using fewer than 10 systems.

The predictive accuracy of the optimal RBMVR and RBMVR-L models is equivalent to that of the optimal CBR model when calibrated using 11 systems. However, when the size of the calibration subset is reduced to eight systems, the accuracy of the CBR model deteriorates more than that of the RBMVR and RBMVR-L models, leading to statistically significantly better performance of the RBMVR and RBMVR-L models. Further reducing the calibration subset to five systems results in consistently greater accuracy of the RBMVR-L model than the
CBR model, however, similar accuracy for the RBMVR and CBR models. Then finally, when the calibration subset is reduced to only two systems, the RBMVR model again statistically significantly outperforms the CBR model, whereas the accuracy of the RBMVR-L model instead becomes similar to that of the CBR model. These results show that the optimal Bayesian statistical model can outperform the CBR model statistically significantly, when calibrated using fewer than 10 systems, at least for the systems studied.

In addition, the above results also show that the log-transformed data generally improves the predictive accuracy of the RBMVR model, provided that the calibration dataset consists of as few as five systems. However, if the size of the dataset becomes fewer than five systems, the benefit of the log-transformation seems questionable, as indicated by the lower accuracy of the RBMVR-L model than that of the RBMVR model when calibrated with two systems.

Overall, the above results show that the optimal Bayesian statistical model, the RBMVR model, is most accurate in predicting the effort of the target system, when the calibration dataset consists of fewer than 10 systems.

VI. CONCLUSIONS

Predictive accuracy of three different types of software development effort prediction models applicable to a small dataset, is compared in two separate case studies with a different software dataset. The models compared are: (1) Bayesian statistical models, (2) multiple linear regression models and (3) case-based reasoning/analogy-based (CBR) models built using AngelPlus [29]. One dataset studied consists of database-oriented transaction processing and/or management information systems developed by a single organization, using particular fourth-generation language software development tools. The other dataset, which is in the public domain, consists of another type of software systems developed by another organization.

The results of the comparison have shown that the predictive accuracy of the models generally deteriorates as the number of software systems in the calibration data decreases. However, the rate and level of the accuracy deterioration vary between the models. The Bayesian statistical models studied are able to predict effort of the software systems studied consistently more accurately than the multiple linear regression model, when the calibration data consists of fewer than 10 software systems.

In comparison to the CBR model, the best Bayesian statistical model studied, the random effect
Bayesian multivariate normal distribution & regression (RBMVR) model, is able to statistically significantly outperform the CBR model, at least for the systems studied in the second case study.

These findings suggest that the Bayesian statistical models, particularly the RBMVR model, would be a better choice in effort prediction when software practitioners and project managers have only a very small local dataset, consisting of fewer than 10 systems similar to their system of interest.

The results presented in this paper are distinguishable from the results reported in previous comparative studies of software development effort prediction models, due to the particular interest in utilizing only a small amount of local data available, and identifying the most accurate model under such restricted information.

However, future studies using different software datasets are required, in order to further investigate predictive accuracy of the Bayesian statistical models, as well as other models, under the similar restricted information. The author would like to invite more software practitioners and researchers to participate in these studies. Although software effort prediction has a history of over 30 years to date, it still has a long and difficult journey ahead. Further consistent cooperation between software industry and academic research community should also be encouraged, so as to advance the status of software effort prediction practice further.

REFERENCES


**Chikako van Koten** The author had studied toward a PhD in the Department of Information Science, University of Otago, New Zealand, since 2004. Her main research area had been software effort prediction, with strong interest in Bayesian statistical modelling. The research so far resulted in two other publications in other Software Engineering academic journals. She completed the research and submitted a thesis in January, 2007, and is currently working in the Department of Mathematics and Statistics in the same university.