A Simulation Study of the Model Evaluation Criterion MMRE

Tron Foss, Erik Stensrud, *Member*, *IEEE*,
Barbara Kitchenham, *Member*, *IEEE Computer Society*, and Ingunn Myrtveit

Abstract—The Mean Magnitude of Relative Error, *MMRE*, is probably the most widely used evaluation criterion for assessing the performance of competing software prediction models. One purpose of *MMRE* is to assist us to select the best model. In this paper, we have performed a simulation study demonstrating that *MMRE* does not always select the best model. Our findings cast some doubt on the conclusions of any study of competing software prediction models that used *MMRE* as a basis of model comparison. We therefore recommend not using *MMRE* to evaluate and compare prediction models. At present, we do not have any universal replacement for *MMRE*. Meanwhile, we therefore recommend using a combination of theoretical justification of the models that are proposed together with other metrics proposed in this paper.

Index Terms—Mean magnitude of relative error, software metrics, simulation, regression analysis, prediction models, software cost estimation, software engineering, empirical software engineering, prediction accuracy.

1 Introduction

Software cost estimates and defect rate estimates are important deliverables of software projects. As a consequence, researchers have proposed and evaluated a plethora of prediction systems. There are a number of empirical studies including studies on *generic* model-based methods [12], [14], [18], [28], [30], [36] as well as on *specific* model-based methods. The latter methods include CART (Classification and Regression Trees) [4], [5], [7], [33], [48], [46], OSR (Optimized Set Reduction) [2], [3], [29], Stepwise ANOVA (Analysis of Variance) [33], OLS (Ordinary Least Squares) regression (more than 30 studies, see [10] for an account), Robust regression [23], [24], [26], [38], [42], [43], Composite Estimation models (like COBRA) [4], Analogy-based models [19], [27], [39], [40], [47], [49], [52], [46] and, finally, artificial neural network-based models [45], [48]. (We have adopted the classification scheme proposed in the *Encyclopedia of Software Engineering* [10] except possibly for neural networks.)

The most widely used evaluation criterion to assess the performance of software prediction models is the *mean magnitude of relative error*, *MMRE* [10]. This is usually computed following standard evaluation processes such as cross-validation [6]. Conte et al. [13] consider $MMRE \leq 0.25$ as acceptable for effort prediction models.

MMRE is used for many purposes. One important use of *MMRE* is to select the best model among two or more *competing* prediction models, e.g., compare an estimation-by-analogy model with a linear regression model. The model obtaining the lowest *MMRE* is deemed "best." Examples of such studies include [40] and [47].

In this paper, performing a simulation study, we investigate whether *MMRE* is a reliable selection criterion or not. The findings suggest that *MMRE* is an *unreliable* selection criterion; in many cases, *MMRE* will select the *worst* candidate out of two competing models; in particular, *MMRE* will tend to prefer a model that *underestimates* to a model that estimates the expected value; in fact, *MMRE* may be lower (i.e., "better") for a bad model than for a good model even when the good model happens to be the *true* model. Miyazaki et al. [38] have pointed out that "*MMRE* underestimates," but neither they nor anybody else have seriously questioned the far reaching implications of this fact.

The consequences of our findings cast doubts on the results of all studies that have relied on *MMRE* to compare the accuracy of predictive cost models. Furthermore, this remains a problem because *MMRE* is still considered the de facto standard [10].

As an aside, we are not aware of *MMRE* being used to evaluate prediction models (like regression analysis, estimation by analogy, etc.) in disciplines other than computer science and software engineering. We are, however, aware of its use in time series analysis. See, for example, Makridakis et al. [35].

The paper is organized as follows: Section 2 illustrates the potential problem with *MMRE*. Section 3 presents and discusses alternative evaluation metrics that we investigate in this study together with *MMRE*. Section 4 presents related work. The main objective of this section is to position our study relative to other related studies and, specifically, point out that no other studies have done the investigation that this paper undertakes. Section 5 describes the simulation method, the simulation model, and the

- *T. Foss and I. Myrtveit are with the Norwegian School of Management BI, Elias Smiths vei 15, Box 580, N-1301 Sandvika, Norway. E-mail: {tron.foss, ingunn.myrtveit}@bi.no.*
- *E. Stensrud is with Myrtveit og Stensrud ANS, Austliveien 30, 0752 Oslo, Norway. E-mail: erik.stensrud@ieee.org.*
- *B. Kitchenham is with the Department of Computer Science, Keele University, Keele Staffordshire, UK ST5 5BG. E-mail: barbara@cs.keele.ac.uk.*

Manuscript received 14 Jan. 2002; revised 30 Jan. 2003; accepted 25 May 2003.
Recommended for acceptance by J. Knight.
For information on obtaining reprints of this article, please send e-mail to: tse@computer.org, and reference IEEECS Log Number 115704.

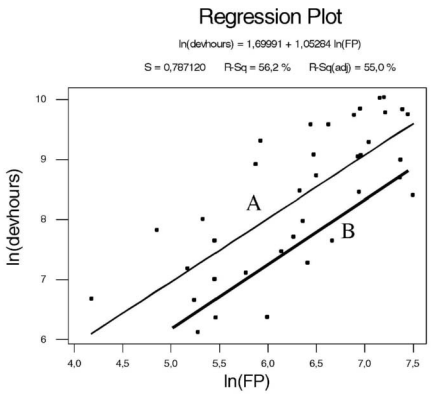

Fig. 1. Two regression models A and B.

"good" and "bad" models that we use as competing models. The section also highlights some of the advantages of simulation versus using real data sets. Whereas Section 5 describes how the simulation model is created, Section 6 describes how the simulation model is used in this study to evaluate the metrics that are in scope. Section 7 presents the results followed up by a discussion in Section 8. Section 9 concludes and, in Section 10, we briefly outline some possible directions for future work.

2 Illustrating the Problem

To illustrate the problem of *MMRE*, let us consider two prediction models *A* and *B*, respectively. If *MMRE* of model *B* is significantly lower than *MMRE* of model *A*, one would conclude that model *B* is better than model *A* (*B* is "more accurate" than *A* in current software engineering terminology). *MRE* is the basic metric in *MMRE* and is defined in (1) as follows [13] (where $y = \text{actual}$, $\hat{y} = \text{prediction}$):

$$MRE = \frac{|y - \hat{y}|}{y}. \quad (1)$$

To be able to draw the correct conclusion with regard to whether model *A* or model *B* is best, it is crucial that the model evaluation metric *selects the model that is closest to the true model most of the time*. (The true, or population, model is the model we would obtain if our data set comprised all the relevant past and future data points, e.g., the population of all past and future software projects that are similar to the project to be predicted). This seems like a reasonable, common sense requirement of an evaluation criterion. Otherwise, the evaluation criterion may lead you to wrongly choose model *B* when model *A* ought to have been chosen.

Consider the two models *A* and *B* in Fig. 1. Model *A* is fitted to the data by OLS log-linear regression. (The data is the Finnish data set, see [32] for details on the data). Next, assume that model *B* is fitted by some other method (Say, the novel method "SuperX" which we have recently developed and which we believe holds great promise.) We observe that model *B* has the same slope as *A* but a different intercept than *A*. (Model *B* Constant = 0.7, Model *A* Constant = 1.7, see Table 1 and Table 2, respectively). Thus, the intercept of model *B* is *underestimated* (compared with *A*).

TABLE 2
Model B

Predictor	Coef
Constant	0.7
ln(FP)	1.05

By visual examination, *A* seems to represent the central tendency of the data reasonably well, at least far better than model *B*. Since *A* has been fitted using OLS, the estimates from *A* equal the expected value (or mean). Also, model *A* is a good regression model in terms of the commonly used criteria *SE* (standard error) and R^2 (See Table 3). Both coefficients of model *A* are significant ($p < 0.10$).

Nevertheless, in terms of *MMRE*, model *B* is preferred to model *A* (Table 3). (*MRE* may be calculated using the formula derived in Appendix A.) As a consequence, we would be misled by *MMRE* to identify model *B* as better (or more "accurate") than model *A*. This is a serious flaw of the evaluation criterion *MMRE* when it is used to select between competing prediction systems. As an evaluation criterion, it therefore clearly does not comply with common sense (nor with statistical science) with regard to identifying which model is the better one.

In this study, we perform a simulation study to demonstrate the extent of the *MMRE* problem. We also take the opportunity to propose and evaluate other goodness of fit statistics.

3 Alternative Evaluation Metrics

In this section, we present other potential evaluation metrics that we evaluate in this study. Another measure akin to *MRE*, the magnitude of error relative to the *estimate*, *MER*, has been proposed by Kitchenham et al. [31]. Intuitively, it seems preferable to *MRE* since it measures the error relative to the estimate. *MER* is a measure of the dispersion of the variable $y/\hat{y}$ if the mean of $y/\hat{y}$ is approximately 1. They further pointed out that researchers should look at the full distribution of the variable and not just the dispersion. *MER* is defined as

$$MER = \frac{|y - \hat{y}|}{\hat{y}}. \quad (2)$$

TABLE 1
Model A (Fitted by OLS Regression)

Predictor	Coef	SE Coef	T	P
Constant	1.7	0.99	1.7	0.096
ln(FP)	1.05	0.16	6.8	0.000

TABLE 3
MMRE, *SE*, and R^2 of Models A and B

Model	N	MMRE	SE	R^2
A	38	0.79	0.79	0.56
B	38	0.59	N/A	N/A

We use the notation *MMRE* to denote the mean *MER*. Another, and simple, measure of residual error is the standard deviation, *SD*. It is computed as follows:

$$SD = \sqrt{\frac{\sum (y_i - \hat{y}_i)^2}{n-1}}. \quad (3)$$

We also propose and evaluate two other measures. They are the relative standard deviation *RSD* and the logarithmic standard deviation *LSD*. *RSD* is defined as follows:

$$RSD = \sqrt{\frac{\sum \left(\frac{y_i - \hat{y}_i}{x_i}\right)^2}{n-1}}. \quad (4)$$

The variable x is function points, *FP*, in our case. The rationale behind *RSD* is to measure the dispersion relative to the x value (e.g., FP) rather than relative to the y value (effort) to avoid one of the problems with *MMRE*. One of *MMRE*'s problems is that small actuals (small ys) will have a (too) large influence on the mean *MRE* since a number divided by a small number tends to be a large number. Contrary to *MRE*, which is almost uncorrelated with size [50], *SD* is positively correlated with size because software project data sets are often heteroscedastic. As opposed to *SD*, *RSD* is almost uncorrelated with size.

We observe that *RSD* is limited to models with a single predictor variable. In many software studies, this is, however, not a serious limitation since it is common to create prediction models based on *FP* and effort. More important, we can provide a rationale for choosing this metric as well as an interpretation of its meaning. As for the rationale, let us assume that we have the following model:

$$y = \alpha + \beta x + x\varepsilon, \quad (5)$$

where ε is normally distributed: $E(\varepsilon) = 0$ and $\mathrm{var}(\varepsilon) = \sigma^2$. This model will generate data where the variance increases with x. Dividing (5) by x gives:

$$\frac{y}{x} = \alpha \cdot \frac{1}{x} + \beta + \varepsilon. \quad (6)$$

The error term in this regression model (6), ε, is normal: $E(\varepsilon) = 0$ and $\mathrm{var}(\varepsilon) = \sigma^2$. OLS will, therefore, be an *efficient* estimating method. Let $\hat{\alpha}$ and $\hat{\beta}$ be estimates of α and β. Our prognosis (sample prediction model) for effort is then:

$$\hat{y} = \hat{\alpha} + \hat{\beta} x. \quad (7)$$

But, then:

$$\frac{\hat{y}}{x} = \hat{\alpha}\frac{1}{x} + \hat{\beta}. \quad (8)$$

$\frac{y}{x} - \frac{\hat{y}}{x}$ is, therefore, an estimate, e, of the error term ε. Since we also have that

$$e = \frac{y}{x} - \frac{\hat{y}}{x} = \frac{y - \hat{y}}{x}. \quad (9)$$

RSD is, therefore, an estimate of the standard deviation of the error term ε in the regression equation. Thus, *RSD* is a relevant measure of how good the prediction model is. It remains to give *RSD* an interpretation making sense since x and y are measured in different units (hours vs. *FP*).We can interpret y/x as the effort per *FP*, that is to say, the productivity. If α is close to zero or if the project is large (in terms of x), we observe that y/x will approximate β.

TABLE 4
Descriptive Statistics for DMR Data Set

Variable	N	Mean	Median	StDev	Min	Max
FP Adj	81	289	255	186	62	1116
Effort	81	5046	3647	4419	546	23940

We note that *RSD* is based on an additive model and many software effort estimation models use an exponential model. Thus, *RSD* may be less effective as a goodness of fit statistic for an exponential model if the exponent is significantly different from one. (However, these two models are very close to each other, see Section 5.2. The exponent is 0.943 and, thus, very close to 1, see Table 6. Therefore, *RSD* is an appropriate measure in this single-predictor case.)

LSD is defined as follows:

$$LSD = \sqrt{\frac{\sum \left(e_i - \left(-\frac{s^2}{2}\right)\right)^2}{n-1}}. \quad (10)$$

The term s^2 is an estimator of the variance of the residual e_i, where e_i is given by

$$e_i = \ln y_i - \ln \hat{y}_i. \quad (11)$$

The rationale behind *LSD* is as follows: Data sets with a large heteroscedasticity like the DMR data set (Table 4) will be very influenced by the large projects. Thus, the usual *SD* is more sensitive to large projects than to small projects and it may therefore not be a stable, reliable measure for such data sets. On the other hand, *LSD* lends itself well to data sets that comply with a log-linear model because the residual error is independent of size (i.e., homoscedastic) on the log scale. In fact, we use a log-linear transformation for our simulation (see Section 5.2), so *LSD* should theoretically be more reliable than *SD* in this case. (The reason for the $-s^2/2$ term will become clearer in Section 5.2. See also Appendix B.) To summarize, *LSD* is useful for comparing multiplicative models but it may be inappropriate for comparing additive models.

Finally, we evaluate the mean of the balanced relative error *BRE* and the inverted balanced relative error *IBRE* proposed by Miyazaki et al. [38]:

$$BRE = \frac{(\hat{y} - y)}{y}, \hat{y} - y \geq 0, \quad (12)$$

$$BRE = \frac{(\hat{y} - y)}{\hat{y}}, \hat{y} - y < 0, \quad (13)$$

$$IBRE = \frac{(\hat{y} - y)}{y}, \hat{y} - y < 0, \quad (14)$$

$$IBRE = \frac{(\hat{y} - y)}{\hat{y}}, \hat{y} - y \geq 0. \quad (15)$$

The mean of the absolute values of *BRE* and *IBRE* are termed *MBRE* and *MIBRE*, respectively.

4 Related Work

There are two categories of work that may be considered related to this study: studies on evaluation metrics as well as simulation studies. Evaluation of the evaluation metrics themselves seems to have received little attention since Conte et al. [13] publicized *MMRE* and other measures. The only related work we are aware of is Miyazaki et al. [38], Kitchenham et al. [31], Foss et al. [20], and Stensrud et al. [50].

Miyazaki et al. identified that *MMRE* is lower for models that underestimate and suggested other summary statistics they believed would be better behaved but they did not investigate the implications of *MMRE* bias, nor the properties of their proposed replacement metrics.

Kitchenham et al. take a different approach. They attempt to understand what *MMRE* (and other measures) really measure. They note that *MMRE* is a measure of the variance of the variable $y/\hat{y}$ and that *PRED* (20) is related to kurtosis. They advise against using a single summary statistic to assess goodness of fit and suggest looking at box plots both of the $y/\hat{y}$ variables and of the simple residual $y - \hat{y}$.

Stensrud et al. found that *MRE* and size are virtually uncorrelated, which is a positive asset of *MRE*. None of the previous studies have investigated the reliability of *MMRE* when used as a criterion to select between *competing* prediction systems.

Regarding simulation studies in software engineering, several papers have recently used simulation in software engineering: Rosenberg [44], El-Emam [22], Briand and Pfahl [9], Angelis and Stamelos [1], Strike et al. [51], Shepperd and Kadoda [46], and Pickard et al. [43]. Therefore, simulation is becoming an accepted research method in software engineering.

Pickard et al. note that simulation is useful both to check whether or not empirically observed relationships result from data analysis procedures rather than genuine relationships and to assess the implications of proposed statistical analysis methods particularly when there are likely to be interactions between the analysis technique and the data set. In our case, we are concerned with assessing the implication of a summary statistic, i.e., confirming that *MMRE* exhibits an undesirable property rather than investigating the data set conditions that affect the property. Since we need only to demonstrate that the undesirable property *exists*, we have used the same technique as El Emam, and Briand and Pfahl, and based our simulation on a single real data set. This contrasts with the approach taken by Pickard et al. and Shepperd and Kadoda who investigated a variety of analysis techniques on artificial data sets constructed to exhibit a variety of different conditions because they were interested in the interaction between the data set properties and the analysis method results.

5 Simulation Method

Since the 1950s, various computer simulation techniques have become increasingly important research tools across a wide range of sciences. Software packages based on these techniques are also widely used in more applied fields such as engineering, finance, or environmental management, often in connection with computerized databases and electronic gathering devices.

There are several advantages of simulation compared with using real data. One advantage is that we can create a large number of samples rather than having to use only one, single sample. Thus, we obtain the *distributions* for statistical parameters that are estimated (the estimators). Using a single sample of real data, we can obtain the distribution only when it can be calculated analytically. In cases where we cannot calculate the distribution analytically, we would obtain one single value for the estimator, not its distribution. In such cases, simulation adds value. Clearly, we obtain more information and can draw more reliable conclusions based on a distribution than based on a single value.

Another advantage of simulation is that we know the true answer. For example, we may study missing data techniques by removing data from a complete (simulated) data set in a controlled manner and study the effects of the missing data techniques by comparing results with the true, complete data set. Strike et al. [51] have used this approach.

Still another advantage with simulation is that it enables the studying of phenomena that are too complex to describe and solve analytically, e.g., the behavior of oil reservoirs.

In this study, we use simulation (combined with regression models) to investigate *MMRE* and alternative evaluation metrics. Using simulation, we demonstrate that, in many cases, *MMRE* fails to select the best prediction model among competing models. The simulation study therefore is a more conclusive demonstration of the conjecture stated in Sections 1 and 2 and illustrated in Fig. 1.

5.1 Data "Template"

It is important that a simulation method for generating data generates data that are as close as possible to actual, and representative, software data sets. In this study, we have primarily used the Desharnais (DMR) data set as a model for our simulation model. The Desharnais data set exhibits properties that we consider representative of other data sets of software projects with respect to (non)linearity and heteroscedasticity. Four more software data sets, presented in [50], have similar properties with DMR in this respect. In addition, it is a reasonably large sample. (It contains 81 projects, see Table 4). For more details on DMR, see [15]. The DMR data come from a single company but have used three different language types. (For some kinds of analysis, it may therefore require partitioning based on the language type. For this analysis, it is, however, unnecessary to partition the data into more homogeneous subsets since it is only used as model for generating a simulation model). The projects span from 1,116 to 23,940 workhours.

Overall, we believe that a simulation model approximating the DMR data set should be close to reality and representative of software cost estimation data sets. (We make, however, no claim that defect estimation data sets have similar characteristics to cost estimation data sets, so our results cannot be assumed to apply to defect estimation models.)

TABLE 5
The Linear Model (17)

Predictor	Coef	SE Coef	T	P
Constant	-40	620	-0.06	0.949
FP	17.6	1.8	9.73	0.000

5.2 Model Specification

It is common, and a reasonable starting point, to assume a linear model of the following form: (Recent research using genetic algorithms as a flexible method of model fitting did not find any significant deviations from a linear model [16].)

$$eff = \alpha + \beta \cdot FP + u. \tag{16}$$

If we apply (16) to the DMR data set [15], we obtain the following OLS linear regression model (Table 5):

$$eff = -40 + 17.6 \cdot FP. \tag{17}$$

On closer inspection of the DMR data set in Fig. 2, we observe that it seems reasonably linear but exhibits a pronounced heteroscedasticity (increasing variance). OLS regression analysis assumes that the data are homoscedastic (equal variance). Model (2) is therefore not sufficiently correct. Therefore, we need to transform the data in an attempt to make them better comply with the assumptions of the OLS method, in particular, the homoscedasticity assumption. There exist several alternatives for transforming the data.

One alternative is to perform a log-linear regression where we assume that the relationship between effort and *FP* is of the form

$$eff = e^{\alpha}(FP)^{\beta} \cdot I, \tag{18}$$

where I is lognormal with mean equal to 1. Thus, $I = e^u$ with u normally distributed. It has been proved that if $Var(u) = \sigma^2$ and $E(u) = -\frac{\sigma^2}{2}$, then $E(I) = E(e^u) = 1$ ([17, vol. 5, p. 134]).

Stating (18) in the form

$$eff = e^{\alpha} \cdot (FP)^{\beta} \cdot e^{u} \tag{19}$$

and taking the logarithm of (19), we obtain

$$\ln(eff) = \alpha + \beta \ln(FP) + u. \tag{20}$$

A plot of the transformed data when applying (20) is presented in Fig. 3. Inspecting the plot, the data exhibit a reasonable homoscedasticity and linearity.

Applying model (20) to the DMR data set, we get the following OLS regression model (21) or (22) or in table form in Table 6:

TABLE 6
The Log-Linear Model (21), (22)

Predictor	Coef	SE Coef	T	P
Constant	3.03	0.59	5.11	0.000
ln(FP)	0.943	0.11	8.77	0.000

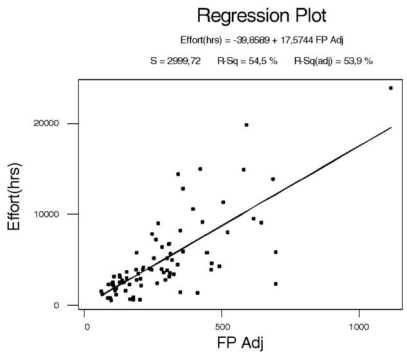

Fig. 2. DMR data set: FP versus Effort.

$$\ln(eff) = 3.03 + 0.943 \cdot \ln(FP), \mathrm{SD} = 0.6000. \tag{21}$$

Backtransforming (21), we get

$$eff = e^{3.03} \cdot (FP)^{0.943} = 20.6972 \cdot (FP)^{0.943}. \tag{22}$$

Comparing (16) with (18), we can state that in (16), we believe in a linear relationship between effort and *FP* whereas in (18), we believe in an exponential relationship between these two variables. We observe, however, that model (22) fitted to the DMR data set is not particularly exponential since the exponent is close to 1 (0.943 and with a standard error of 0.11, see Table 6). From this, we cannot draw any conclusions regarding returns to scale, i.e., whether to assume increasing, decreasing, like COCOMO (Constructive Cost Model) [12], or constant returns to scale. See [41] for an account of returns to scale. However, none of the two models reject the assumption of constant returns to scale. Therefore, a linear model seems to be a good first order approximation of reality.

Given that the data seem reasonably linear, we could have used (16) except for the fact that the data are heteroscedastic. Therefore, it is interesting to investigate a third model that is linear rather than log-linear but corrects the heteroscedasticity of (16).

$$eff = \alpha + \beta \cdot (FP) + (FP)u. \tag{23}$$

In model (23), we assume a linear relationship between $FP$and effort as we do in (16), but unlike (16), we transform the data in an attempt to obtain a more constant variance. Equation (23) may be restated as

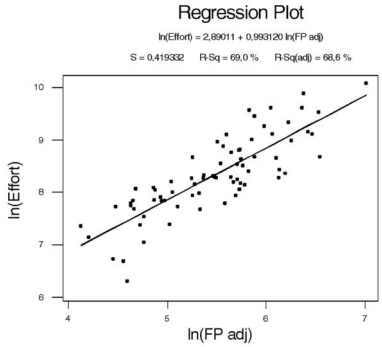

Fig. 3. *DMR* data set: *ln(FP)* versus *ln(Effort)*.

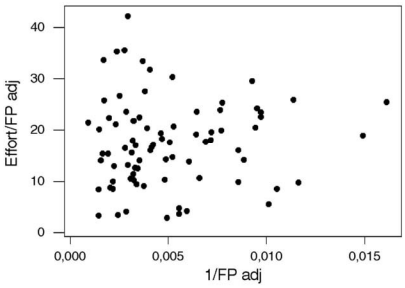

Fig. 4. DMR data set: 1/FP versus Effort/FP.

$$\frac{eff}{(FP)} = \alpha \cdot \frac{1}{(FP)} + \beta + u. \quad (24)$$

Thus, in (24), $\frac{eff}{(FP)}$ is the dependent variable and $\frac{1}{(FP)}$ is the independent variable. The OLS regression model applying (24) on the DMR data is

$$\frac{eff}{(FP)} = 16.9 + 127 \cdot \frac{1}{(FP)}. \quad (25)$$

The standard deviation of the residual in (25) is 8.4. Model (25) may alternatively be stated as

$$eff = 127 + 16.9 \cdot (FP). \quad (26)$$

A plot of model (25) is provided in Fig. 4. In (25), the residual u seems less heteroscedastic than the residual of (16). Unfortunately, it is not perfectly homoscedastic either. This is observed by comparing the plots of Fig. 2 and Fig. 4. Compared with the log-linear model (18), model (25) seems no better. In addition, comparing Table 6 and Table 7, we observe that the intercept of the log-linear model (18) has a higher t-value than the intercept of model (25), and that the slope coefficients have similar t-values. Overall, the log-linear model seems the best choice.

In the simulation study that follows, we have therefore opted for model (18), the log-linear model. The log-linear model also has an additional benefit compared with the two other models in that it corrects heteroscedasticity as well as forces the line through the origin. (That is, when zero FP is delivered, zero effort is expended.)

We find it useful to have performed and reported this explorative exercise because, unfortunately, there is no strong theory to guide us in software engineering. We have therefore chosen (18) based on empirical evidence of the DMR data set. This empirical evidence weakly suggests a multiplicative model rather than an additive model.

5.3 Simulation Model and Procedure

Let us assume that we have estimated the regression parameters based on a *large* sample (or alternatively, suppose that we happen to know the population) and that we therefore know the *true* (or population) regression model. Assume that the true model is exponential of the form (18) and with coefficients and standard deviation σ identical to model (20):

$$\ln(eff) = 3.03 + 0.943 \cdot \ln(FP) + u, \sigma = 0.600. \quad (27)$$

TABLE 7
Model (25), (26)

Predictor	Coef	SE Coef	T	P
Constant	127	298	0.43	0.671
FP	16.9	1.8	9.6	0.000

Model (27), i.e., the regression model including the error term u and the standard deviation σ, is our simulation model describing the population. The parameters describing the population should describe aspects of the population that are relevant to the simulation study. For this study, it is relevant to include a model for the variance of the population. The error term u accounts for the stochastics (e.g., variables not captured in our model). Some projects use more effort, and some less effort, than the expected effort based on FP counts. u is normal with mean equal to $-\sigma^2/2$ and variance equal to σ^2 ([17, vol. 5, p. 134]). This simulation model generates data sets with characteristics similar to the DMR data set (and presumably similar to other software project data sets).

If the population is given by model (27), we may simulate sampling from this population. Let us assume that we have conducted 30 projects of different size, say, $FP_i = 50 \cdot i$, $i = 1, 2, \ldots, 30$. Then, the smallest project will have $FP_1 = 50$ and the largest in the sample will have $FP_{30} = 1,500$. (This is close to the span of the DMR data set, see Table 4. Only, we draw 30 rather than 81 observations. This is because many software data sets have around 30 observations.) For each project i, we draw a random number u_i from a normal distribution with mean equal to -0.18 $(-\sigma^2/2)$ and standard deviation σ^2 equal to -0.600. (This is standard functionality in statistical packages, e.g., Minitab [37]: Menu tree: Calc.random.normal). Thus, we loop from $i = 1, 2, \ldots, 30$ and generate FP_i each time using the formula $FP_i = 50 \cdot i$. Next, we compute the effort eff_i using (27) with FP_i as input as well as input of an u_i value. This procedure gives us a data set of 30 observations with characteristics similar to the DMR data set. In this way, we may create as many samples of 30 observations as we wish. For the simulation study, we created 1,000 samples of 30 observations each. We observe that the simulation procedure is a reverse procedure of the ordinary regression analysis procedure. (In regression analysis, we start with the observations and estimate the regression model whereas in this simulation we start with a know population regression model and create a number of observations).

5.4 Simulating Competing Models

Let us assume that we have created a variety of prediction models based on a variety of procedures and based on samples drawn from the population described by the simulation model (27). (These procedures could have been classification and regression trees, estimation by analogy, neural nets, or some novel method X.) For simplicity, assume that we have created four different multiplicative models based on different samples and different methods X1, X2, X3, and X4 as follows:

TABLE 8
Results of True Model versus Model (28)

	True	Model (28)
Best MMRE	22	978
Best MdMRE	494	506
Best MMER	999	1
Best SD	922	78
Best RSD	970	30
Best LSD	982	18
Best MBRE	783	217
Best MIBRE	729	271

TABLE 9
Results of True Model versus Model (29)

	True	Model (29)
Best MMRE	0	1000
Best MdMRE	328	672
Best MMER	952	48
Best SD	656	344
Best RSD	686	314
Best LSD	716	284
Best MBRE	175	825
Best MIBRE	265	735

$$eff = e^{2.50} \cdot (FP)^{0.943}, \quad (28)$$

$$eff = e^{3.03} \cdot (FP)^{0.920}, \quad (29)$$

$$eff = e^{3.50} \cdot (FP)^{0.943}, \quad (30)$$

$$eff = e^{3.03} \cdot (FP)^{0.970}. \quad (31)$$

We observe that (28) has a smaller intercept than the true model (22) but equal β. That is, model (28) underestimates. We should therefore expect it to be considered as superior to the true model terms of *MMRE* since *MMRE* favors models that underestimate. Model (29) has intercept equal to the true model whereas the slope is smaller than the slope of the true model. Thus, model (29) also underestimates, so we should expect *MMRE* to identify this model as superior to the true model. Although model (29)'s underestimates are more pronounced for larger size values, it never produces as extreme underestimates as model (28), over the range of size values used in our simulation. Thus, model (28) is an example of a severely underestimating model, and model (29) is an example of moderately underestimating model.

Model (30) has a greater intercept than the true model and equal slope. We should therefore expect this model to be identified as inferior to the true model in terms of *MMRE*. Model (31) has a greater slope than the true model and equal intercept. Thus, model (31) overestimates, and we expect *MMRE* to identify this model as inferior to the true model. Although model (31)'s overestimates are more pronounced for large size values, it never produces as extreme overestimates as model (30), over the size range used in our simulation. Thus, model (30) is an example of a severely overestimating model, and model (31) is an example of moderately overestimating model.

6 Research Procedure

The research procedure consisted of creating 1,000 samples based on the simulation model (11). Thereafter, we computed *MMRE*, *MdMRE* (median *MRE*), *MMER*, *SD*, *RSD*, *LSD*, *MBRE*, and *MIBRE* values for the five models (true (27), (28), (29), (30), and (31)) on all the 1,000 samples. We compared models pairwise with the true model. For each pairwise comparison, we counted the number of times each model obtained the best *MMRE*, *MdMRE*, *MMER*, *SD*, *RSD*, *LSD*, *MBRE*, and *MIBRE* values, respectively. For example, suppose we compare the true model and model (28) on 1,000 samples by computing *MMRE* for each model on each sample. Suppose the results are that (28) obtains the lowest (best) *MMRE* on 900 of the samples and the true model obtains the lowest *MMRE* on the remaining 100 samples. Then, we report 900 for model (28) and 100 for the true model. This should be interpreted as follows: When *MMRE* is used to select between competing prediction systems, the estimated probability that we select (28) is 0.9 whereas the estimated probability of selecting the true model is 0.1.

MRE may be computed using the formula (A8) in Appendix A, or it may be computed directly using the multiplicative form of the function. Similar formulas may be used to compute the other measures (not reported).

7 Results

The results for the comparison are presented in Tables 8, 9, 10, and 11. We have reported the number of times each model obtains the best score (i.e., lowest *MMRE*, etc.). We reiterate that a sensible evaluation criterion ought to have a high probability of identifying the true model as best.

Focussing on *MMRE* in the four tables, we observe that *MMRE* identifies models that underestimate as superior to the true model and identifies models that overestimate as inferior to the true model. *MdMRE* exhibits a similar pattern to *MMRE* but the effect is less severe and appears to be influenced by the strength of the overestimate, i.e., it is more likely to select the correct model if the underestimate is severe. *MMRE* identifies the true model as best in three out of four cases. It is therefore not sufficiently consistent in identifying the true model as best. *SD* identifies the true model as best in all cases and is therefore consistent. *RSD* also identifies the true model as better than the four other models and is therefore consistent, too. Compared with *SD*, *RSD* also identifies the true model as best with a higher probability than *SD*. *LSD*, too, consistently identifies the true model as best with a reasonably high probability (> 0.7). *MBRE* and *MIBRE* perform similarly to *MMER*. They both identify the true model as best in three out of four cases.

MMER does not perform particularly well in this study. In theory, however, *MMER* might perform better for data sets exhibiting a more pronounced heteroscedasticity than the DMR data. Like *SD*, and unlike *MMRE*, it favors models with a good fit to the data, and unlike *MMRE*, it is not sensitive to small actuals. Therefore, given the choice

TABLE 10
Results of True Model versus Model (30)

	True	Model (30)
Best MMRE	1000	0
Best MdMRE	996	4
Best MMER	590	410
Best SD	943	57
Best RSD	981	19
Best LSD	964	34
Best MBRE	1000	0
Best MIBRE	998	2

TABLE 11
Results of True Model versus Model (31)

	True	Model (31)
Best MMRE	1000	0
Best MdMRE	870	130
Best MMER	260	740
Best SD	734	266
Best RSD	765	235
Best LSD	746	254
Best MBRE	988	12
Best MIBRE	924	76

between *MMRE* and *MMER*, we argue that *MMER* is to be preferred.

Among all the measures evaluated in this study, *MMRE* is probably the *worst choice*. It clearly favors models that underestimate, and it is extremely sensitive to small actuals.

As an aside, we also observe that *MMRE* does not take the number of observations, *N*, into account. Common sense dictates that we generally should have more confidence in a model based on 100 observations than a model based on two observations. The latter model would likely obtain a better *MMRE* but would, nevertheless, inspire less confidence, in general.

To conclude, *MMRE*, *MdMRE*, *MMER*, *MBRE*, and *MIBRE* are substantially more unreliable as evaluation criteria than *SD*, *RSD*, and *LSD*. All of the latter three criteria are consistent, and *RSD* and *LSD* seem slightly better than *SD*. It seems that a good selection criterion ought to prefer the true model, the truth, most of the time. (Ideally, it ought to prefer the truth all of the time, but this is, of course, infeasible for evaluation criteria based on statistical methods.)

8 Threats to the Study Validity

The major threat to the validity of our study is the validity of the simulation exercise. In this section, we discuss some of the several critical decisions we made when undertaking our study.

In the simulation study, we generated 1,000 samples. This is common in simulation studies. Each sample contained 30 observations. This is a sample size that reasonably well reflects average software project data sets. See, e.g., [50] for a presentation of five different data sets.

We performed significance tests of the difference in *MMRE* values for each pairwise comparison (not reported). Testing for statistical significance is, however, not so meaningful because this is a simulation study where we have fitted the four "bad" models as badly as possible in order to obtain as significant results as possible. Therefore, we may obtain as significant results as we wish. This situation is different from a situation where we want to compare two *given* fitting methods (for example OLS regression versus estimation by analogy). In the latter study, it is important to test for significance.

We also compared two models at a time. In most software engineering (SE) studies, a new, proposed model is compared against a baseline model like OLS regression. Therefore, our pairwise comparisons reflect the majority of SE studies. There are, however, cases where more than two models are compared simultaneously. This would require a different set-up of the simulation study.

Last, but not least, the simulation model was based on one data set. However, we selected a data set exhibiting many typical characteristics of cost estimation data sets, in particular, the presence of heteroscedasticity. This property is relevant in the context of evaluating a relative error measure like *MRE* since *MRE* is intended to correct for heteroscedasticity. Thus, we believe a simulation model based on our single data set is sufficient to confirm the *existence* of any problems with *MRE* unless one can argue that this particular data set is atypical among cost estimation data sets.

9 Conclusion

MMRE has for many years been, and still is, the de facto criterion to select between competing prediction models in software engineering. It is therefore important that we can rely on *MMRE* in selecting the best model among two or more choices. This study suggests that we cannot rely on *MMRE* for this purpose. The conclusions that we can draw from the empirical results of this study are the following:

1. *MMRE* is an unreliable criterion when used to select between competing prediction models. There is a high probability that *MMRE* will prefer a model with a bad fit to a model with a good fit to the data. In particular, there is a high probability that *MMRE* will select a model that provides an estimate *below the mean* (i.e., "underestimates") to a model that predicts the mean. Given that we prefer information on a precisely defined statistic as the *mean* to some nondefined, optimistic prediction somewhere between the mean and zero effort, *MMRE* is inadequate.
2. *LSD* is appropriate to evaluate multiplicative models. Given that other software engineering data sets exhibit characteristics similar to the DMR data set, multiplicative models fit well with reality.
3. *RSD* is also appropriate to evaluate models that fit with data similar to the DMR data, that is to say, data that are fairly linear as well as heteroscedastic. However, *RSD* is limited to data with a single predictor variable.

4. *SD* is appropriate to evaluate linear, additive models, i.e., where the data are homoscedastic.
5. *SD*, *RSD*, and *LSD* are the only criteria that select the true model with a probability $p > 0.5$. The other metrics, *MMER*, *MBRE*, and *MIBRE*, do not. As for *MMER* versus *MMRE*, from a theoretical perspective, we would prefer *MMER* to *MMRE* because *MER*, unlike *MRE*, measures the inaccuracy relative to the estimate.
6. All the metrics investigated, including *SD*, *RSD*, and *LSD*, suffer from either a flaw or a limitation and are therefore not universal solutions to the problem of selecting between competing models of different types, e.g., comparing a linear model with a nonlinear arbitrary function approximator like estimation-by-analogy. As for *RSD*, it is limited to univariate data. *SD* requires homoscedastic data to be meaningful, and *LSD* requires data that are homoscedastic on the log-log scale to be meaningful.

10 Alternatives for Evaluating Prediction Models

Our results have shown that *MMRE* is not suitable for comparing software effort prediction models and that the other statistics proposed in this study either are unsuitable or suffer from a limitation, so the question remains as to how we should compare such models. Although researchers in the past have sometimes reported several measures, typically *MMRE*, *MdMRE*, and *PRED(k)*, we do not believe that reporting several measures that are all based on *MRE* would improve matters.

Actually, this study suggests that it probably is futile to search for the Holy Grail: a single, simple-to-use, universal goodness-of-fit kind of metric, which can be applied with ease to compare a linear regression model with a nonlinear, arbitrary function approximator (for example, an estimation-by-analogy model). Given that we are right in this assumption, what alternatives can we envisage? In the following, we speculate on some alternatives.

One alternative is to apply established evaluation procedures from statistics. Empirical software engineering is a multidisciplinary field with links to software engineering as well as to statistics. The latter discipline is older and presumably more mature with respect to data analysis than empirical software engineering, and it would therefore not come as any surprise if this discipline already has adequate solutions to our problems. It therefore would seem wise to investigate what is the state-of-the-art in statistics before trying to (re-)invent statistical analysis methods that are proprietary to empirical software engineering.

Statistical science has developed a number of concepts and methods to evaluate prediction models and to select the best among two competing models. For example, to evaluate which fitting method is the better to fit linear models to a sample of observations, the concept of best linear unbiased estimator (BLUE) has been developed in statistics [25]. Using the BLUE criterion, we can decide whether the ordinary least squares (OLS) method or the least absolute deviation (LAD) method is the most efficient fitting method for a particular data set [21]. (*Efficient* is a reserved word in statistics meaning *best unbiased*. That is, an efficient method provides estimators, i.e., estimates of the coefficients, with smaller variance than any competing method (therefore, *best*) and a point estimate equal to the mean (therefore, *unbiased*). In short, an efficient method results in a model closer to the truth than any competitor fitting method. See, e.g., [25, Appendix A.7] for a definition of the term efficiency.) According to statistics, it is the characteristics of the data set that decides which fitting method is efficient. For example, given that the OLS assumptions are fulfilled, the OLS method will be efficient. On the other hand, if the kurtosis is high, LAD may be more efficient than OLS. Thus, it seems we ought to have theories as well as data to defend our use of a particular type of fitting method (e.g., OLS or LAD).

We do not know whether statistics offers a solution for selecting between different types of arbitrary function approximators, and we do not know if there exists a solution for selecting between a linear OLS model and an arbitrary function approximator. If statistical science does not have a solution for such comparisons, we ought to ask ourselves why.

In addition, we should investigate the statistical properties of other prediction models, such as arbitrary function approximators, in more detail, particularly with respect to obtaining prediction intervals with quantified probabilities and well-defined properties of the point estimates. For example, the point estimate from an OLS model is unbiased, i.e., it is a well-defined statistic.

Appendix A

Calculation of MRE in Log-Linear Regression Models

This appendix shows how the formula for calculating *MRE* is derived when one applies a log-linear regression model to predict effort. Let y be the actual and $\hat{y}$ be the prediction. Further, let the log-linear population model be

$$\ln y = \ln \alpha + \beta \ln X + \ln u. \tag{A1}$$

Then, the sample model is

$$\ln \hat{y} = \alpha + n \ln X. \tag{A2}$$

The residual is given by

$$residual = \ln y - \ln \hat{y}, \tag{A3}$$

which is equal to

$$residual = \ln\left(\frac{y}{\hat{y}}\right). \tag{A4}$$

This may be transformed to

$$e^{-residual} = \frac{\hat{y}}{y}. \tag{A5}$$

Thus,

$$1 - e^{-residual} = \frac{y - \hat{y}}{y}. \tag{A6}$$

By definition, *MRE* is

$$MRE = \left|\frac{y - \hat{y}}{y}\right|. \quad \text{(A7)}$$

From (A6) and (A7), we may restate *MRE* as

$$MRE = \left|1 - e^{-residual}\right|. \quad \text{(A8)}$$

APPENDIX B

THE ISSUE OF $-\Sigma^2/2$

In Sections 5.2 and 5.3, we state that the error term of the exponential model (18), u, is normal with mean equal to $-\sigma^2/2$ and equal variance σ^2. Therefore, when we apply OLS to the log-transformed regression equation

$$\ln y = \alpha + \beta x + u, \quad \text{(A9)}$$

we obtain the estimator of α, $\hat{\alpha}^*$, where

$$E(\hat{\alpha}^*) = \left(\alpha - \frac{\sigma^2}{2}\right). \quad \text{(A10)}$$

Thus, $\hat{\alpha}^*$ is not an unbiased estimator of α. We would have to add $\frac{s^2}{2}$ to obtain an unbiased estimator, $\hat{\alpha}$, for α (s^2 is an estimator for σ^2) with $\hat{\alpha}$, thus, given as

$$\hat{\alpha} + \hat{\alpha}^* + \frac{\sigma^2}{2}. \quad \text{(A11)}$$

To obtain unbiased predictions, we ought to use the unbiased estimator $\hat{\alpha}$. However, we have not used the models to make any predictions in this paper. On the contrary, we have used the true multiplicative model (18) solely to generate data samples.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous referees for a number of useful suggestions to improve the paper.

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.C. Briand, V.R. Basili, and W.M. Thomas, "A Pattern Recognition Approach for Software Engineering Data Analysis," *IEEE Trans. Software Eng.*, vol. 18, no. 11, pp. 931-942, Nov. 1992.
[3] L.C. Briand, V.R. Basili, and C.J. Hetmanski, "Developing Interpretable Models with Optimized Set Reduction for Identifying High-Risk Software Components," *IEEE Trans. Software Eng.*, vol. 19, no. 11, pp. 1028-1044, Nov. 1993.
[4] 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.
[5] L.C. Briand, K. El-Emam, and I. Wieczorek, "A Case Study in Productivity Benchmarking: Methods and Lessons Learned," *Proc. Ninth European Software Control and Metrics Conf.*, pp. 4-14, 1998.
[6] L.C. Briand, K. El-Emam, and I. Wieczorek, "Explaining the Cost of European Space and Military Projects," *Proc. 21st Int'l Conf. Software Eng.*, pp. 303-312, 1999.
[7] L.C. Briand, K. El-Emam, K. Maxwell, D. Surmann, and I. Wieczorek, "An Assessment and Comparison of Common Cost Software Project Estimation Methods," *Proc. 21st Int'l Conf. Software Eng.*, pp. 313-322, 1999.
[8] L.C. Briand, T. Langley, and I. Wieczorek, "A Replicated Assessment and Comparison of Common Software Cost Modeling Techniques," *Proc. Int'l Conf. Software Eng.*, pp. 377-386, 2000.
[9] L.C. Briand and D. Pfahl, "Using Simulation for Assessing the Real Impact of Test Coverage on Defect Coverage," *IEEE Trans. Reliability*, vol. 49, no. 1, pp. 60-70, 2000.
[10] L.C. Briand and I. Wieczorek, "Resource Modeling in Software Engineering," *Encyclopedia of Software Eng.*, second ed., J. Marciniak, ed., Wiley, in press.
[11] A.F. Chalmers, *What is This Thing Called Science?* second ed. Buckingham: Open Univ. Press, 1982.
[12] The COCOMO II Suite, http://sunset.usc.edu/research/cocomosuite/index.html, 2002.
[13] S.D. Conte, H.E. Dunsmore, and V.Y. Shen, *Software Engineering Metrics and Models.* Menlo Park, Calif.: Benjamin/Cummings, 1986.
[14] A.M.E. Cuelenare, M.J.I. van Genuchten, and F.J. Heemstra, "Calibrating a Software Cost Estimating Model: Why and How," *Information and Software Technology*, vol. 29, no. 10, pp. 558-569, 1987.
[15] J.M. Desharnais, "Analyse Statistique de la Productivite des Projects de Developpement en Informatique a Partir de la Technique des Points de Fonction," Master's thesis, Univ. du Quebec a Montreal, 1989.
[16] J.J. Dolado, "On the Problem of the Software Cost Function," *Information Software Technology*, vol. 43, no. 1, pp. 61-72, 2001.
[17] *Encyclopedia of Statistical Sciences*, S. Kotz, N.L. Johnson, and C.B. Read, eds., New York: Wiley, 1982-1998.
[18] D.V. Ferens, "Software Cost Estimation in the DoD Environment," *Am. Programmer*, pp. 28-34, July 1996.
[19] G.R. Finnie, G.E. Wittig, and J.M. Desharnais, "A Comparison of Software Effort Estimation Techniques: Using Function Points with Neural Networks, Case-Based Reasoning and Regression Models," *J. Systems and Software*, vol. 39, no. 3, pp. 281-289, 1997.
[20] T. Foss, I. Myrtveit, and E. Stensrud, "MRE and Heteroscedasticity," *Proc. 12th European Software Control and Metrics Conf.*, pp. 157-164, 2001.
[21] T. Foss, I. Myrtveit, and E. Stensrud, "A Comparison of LAD and OLS Regression for Effort Prediction of Software Projects," *Proc. 12th European Software Control and Metrics Conf.*, pp. 9-15, 2001.
[22] K. El-Emam, "The Predictive Validity Criterion for Evaluating Binary Classifiers," *Proc. Second Int'l Software Metrics Symp.*, pp. 235-244, 1998.
[23] T. Foss, I. Myrtveit, and E. Stensrud, "A Comparison of LAD and OLS Regression for Effort Prediction of Software Projects," *Proc. 12th European Software Control and Metrics Conf.*, pp. 9-15, 2001.
[24] A.R. Gray and S.G. MacDonell, "Software Metrics Data Analysis —Exploring the Relative Performance of Some Commonly Used Modeling Techniques," *Empirical Software Eng.*, vol. 4, pp. 297-316, 1999.
[25] D.N. Gujarati, *Basic Econometrics*, third ed. New York: McGraw-Hill, 1995.
[26] R. Jeffery, M. Ruhe, and I. Wieczorek, "Using Public Domain Metrics to Estimate Software Development Effort," *Proc. Fifth Int'l Software Metrics Symp.*, pp. 16-27, 2001.
[27] R. Jeffery and F. Walkerden, "Analogy, Regression and Other Methods for Estimating Effort and Software Quality Attributes," *Proc. European Conf. Optimizing Software Development and Maintenance*, pp. 37-46, 1999.
[28] R.L. Jenson and J.W. Bartley, "Parametric Estimation of Programming Effort: An Object-Oriented Model," *J. Systems and Software*, vol. 15, pp. 107-114, 1991.
[29] M. Jørgensen, "Experience With the Accuracy of Software Maintenance Task Effort Prediction Models," *IEEE Trans. Software Eng.*, vol. 21, no. 8, pp. 674-681, 1995.
[30] C.F. Kemerer, "An Empirical Validation of Cost Estimation Models," *Comm. ACM*, vol. 30, no. 5, pp. 416-429, 1987.
[31] B.A. Kitchenham, S.G. MacDonell, L.M. Pickard, and M.J. Shepperd, "What Accuracy Statistics Really Measure," *IEE Proc. Software*, vol. 148, no. 3, pp. 81-85, 2001.
[32] B.A. Kitchenham and K. Kansala, "Inter-Item Correlations Among Function Points," *Proc. First METRICS Conf.*, pp. 11-14, 1993.
[33] B.A. Kitchenham, "A Procedure for Analyzing Unbalanced Datasets," *IEEE Trans. Software Eng.*, vol. 24, no. 4, pp. 278-301, Apr. 1998.
[34] A. Koutsoyiannis, *Theory of Econometrics*, second ed. London: MacMillan, 1977.

[35] S. Makridakis, S.C. Wheelwright, and R.J. Hyndman, *Forecasting Methods and Applications,* third ed. John Wiley & Sons Inc., 1998.
[36] R. Martin, "Evaluation of Current Software Costing Tools," *ACM SIGSOFT Software Eng. Notes,* vol. 13, no. 3, pp. 49-51, 1988.
[37] Minitab Statistical Software Release 13, www.minitab.com, 2000.
[38] Y. Miyazaki, M. Terakado, K. Ozaki, and H. Nozaki, "Robust Regression for Developing Software Estimation Models," *J. Systems and Software,* vol. 27, pp. 3-16, 1994.
[39] T. Mukhopadhyay, S.S. Vicinanza, and M.J. Prietula, "Examining the Feasibility of a Case-Based Reasoning Model for Software Effort Estimation," *MIS Quarterly,* pp. 155-171, June 1992.
[40] I. Myrtveit and E. Stensrud, "A Controlled Experiment to Assess the Benefits of Estimating with Analogy and Regression Models," *IEEE Trans. Software Eng.,* vol. 25, no. 4, pp. 510-525, 1999.
[41] I. Myrtveit and E. Stensrud, "Benchmarking COTS Projects Using Data Envelopment Analysis," *Proc. Sixth Int'l Software Metrics Symp.,* pp. 269-278, 1999.
[42] P. Nesi and T. Querci, "Effort Estimation and Prediction for Object-Oriented Systems," *J. Systems and Software,* vol. 42, pp. 89-102, 1998.
[43] L. Pickard, B. Kitchenham, and S.J. Linkman, "Using Simulated Data Sets to Compare Data Analysis Techniques Used for Software Cost Modelling," *IEE Proc. Software,* vol. 148, no. 6, pp. 165-174, Dec. 2001.
[44] J. Rosenberg, "Some Misconceptions about Lines of Code," *Proc. First Int'l Software Metrics Symp.,* pp. 137-142, 1997.
[45] B. Samson, D. Ellison, and P. Dugard, "Software Cost Estimation Using and Albus Perceptron (CMAC)," *Information and Software Technology,* vol. 39, pp. 55-60, 1997.
[46] M. Shepperd and G. Kadoda, "Using Simulation to Evaluate Predition Techniques," *Proc. Proc. Fifth Int'l Software Metrics Symp.,* pp. 349-359, 2001.
[47] M. Shepperd and C. Schofield, "Estimating Software Project Effort Using Analogies," *IEEE Trans. Software Eng.,* vol. 23, no. 12, pp. 736-743, Dec. 1997.
[48] R. Srinivasan and D. Fisher, "Machine Learning Approaches to Estimating Software Development Effort," *IEEE Trans. Software Eng.,* vol. 21, no. 2, pp. 126-137, Feb. 1995.
[49] E. Stensrud and I. Myrtveit, "Human Performance Estimating with Analogy and Regression Models: An Empirical Validation," *Proc. Second Int'l Software Metrics Symp.,* pp. 205-213, 1998.
[50] E. Stensrud, T. Foss, B. Kitchenham, and I. Myrtveit, "A Further Empirical Investigation of the Relationship between MRE and Project Size," *Empirical Software Eng.,* 2002.
[51] K. Strike, K. El-Emam, and N. Madhavji, "Software Cost Estimation with Incomplete Data," *IEEE Trans. Software Eng.,* vol. 27, no. 10, pp. 890-908, Oct. 2001.
[52] F. Walkerden and R. Jeffery, "An Empirical Study of Analogy-based Software Effort Estimation," *Empirical Software Eng.,* vol. 4, no. 2, pp. 135-158, 1999.

Tron Foss received the MS degree in mathematics from the University of Oslo in 1970. He is an associate professor in multivariate statistics and mathematics at the Norwegian School of Management. His main research interests are multivariate statistics and econometrics.

Erik Stensrud received the MS degree in physics from the Norwegian Institute of Technology in 1982, the MS degree in petroleum economics from the Institut Francais du Petrole in 1984 and a Dr. philos. in software engineering from the University of Oslo in 2000. He is an independent IT consultant who has developed and managed software projects for more than 15 years serving with major consultancy companies including Accenture and Ernst & Young. His interdisciplinary research interests include software engineering, software economics, software metrics, and applied statistics. He is a visiting professor at Bournemouth University and a former associate professor at the Norwegian School of Management. He is a member of IEEE and IEEE Computer Society.

Barbara Kitchenham received the PhD degree from the University of Leeds. She is a professor of quantitative software engineering at Keele University. Her main research interest is software metrics and its application to project management, quality control, risk management, and evaluation of software technologies. She is particularly interested in the limitations of technology and the practical problems associated with applying measurement technologies and experimental methods to software engineering. She is a chartered mathematician and a fellow of the Institute of Mathematics and Its Applications. She is also a fellow of the Royal Statistical Society. She is a visiting professor at both the University of Bournemouth and the University of Ulster. She is a member of the IEEE Computer Society.

Ingunn Myrtveit received the MS degree in management from the Norwegian School of Management in 1985 and the PhD degree in economics from the Norwegian School of Economics and Business Administration in 1995. She is an associate professor in business economics at the Norwegian School of Management. She has also been a senior manager at Accenture's World Headquarters R&D Center in Chicago. Her research interests include managerial economics, empirical studies, software engineering economics, and software metrics.