Comparing the performance of fault prediction models which report multiple performance measures: recomputing the confusion matrix

David Bowes  
Science and Technology Research Institute  
University of Hertfordshire  
College Lane  
Hatfield, AL10 9AB  
United Kingdom  
d.h.bowes@herts.ac.uk

Tracy Hall  
Department of Information Systems and Computing  
Brunel University  
Uxbridge  
Middlesex, UB8 3PH  
United Kingdom  
tracy.hall@brunel.ac.uk

David Gray  
Science and Technology Research Institute  
University of Hertfordshire  
College Lane  
Hatfield, AL10 9AB  
United Kingdom  
d.gray@herts.ac.uk

ABSTRACT
There are many hundreds of fault prediction models published in the literature. The predictive performance of these models is often reported using a variety of different measures. Most performance measures are not directly comparable. This lack of comparability means that it is often difficult to evaluate the performance of one model against another. Our aim is to present an approach that allows other researchers and practitioners to transform many performance measures back into a confusion matrix. Once performance is expressed in a confusion matrix alternative preferred performance measures can then be derived. Our approach has enabled us to compare the performance of 601 models published in 42 studies. We demonstrate the application of our approach on several case studies, and discuss the advantages and implications of doing this.

1. INTRODUCTION
Imagine the following simplified scenario:

You are a practitioner thinking about starting to use fault prediction models. You hope that such models will help you to identify the most fault prone parts of your system. You then plan to target your test effort on those parts of the system. You think that doing this may reduce the faults delivered to your users and reduce the cost of your system. You are not an expert in fault prediction models yourself, but you have seen many such models published in the literature. You identify several published models that have been developed in a similar software development context to your own. You decide to evaluate the performance of these models with a view to trying out the top three models in your project. However when you look at the model performance figures they are reported using a variety of different performance measures.

Several studies report Precision\(^1\) and Recall. Some report Error Rate. Some report pd and pf. Others report Popt. A few report Area Under the Curve of the Receiver Operator Curve. One provides a confusion matrix. It is beyond your expertise to identify a comparative point of reference amongst these different measures. You struggle to understand how the overall performance of a model compares to the others. And so you decide that fault prediction models are too complicated to use and abandon the idea.

This type of scenario may partially explain why the uptake of fault prediction models is low in industry. This low uptake is important as finding and fixing faults in code costs the software industry many millions of dollars every year. Predicting effectively where faults are in code occupies many researchers and practitioners. Our previous work [14] showed that 208 software fault prediction studies were published between January 2000 and December 2010. These 208 studies contained many hundreds of fault prediction models.

Despite this significant research effort it remains difficult or inconvenient to compare the performance of these models. The difficulty in comparing predictive performance means that identifying which fault prediction models perform best in a given context is complex. This complexity in comparing the performance of models is not only a likely barrier to practitioners using fault prediction models, but also makes it difficult for researchers to meta-analyse fault prediction studies [8]. This lack of opportunity to meta-analyse limits the ability of the fault prediction community to mature, as we are not building an evidence base that is as useful as it should be.

One of the difficulties when comparing the performance of fault prediction models stems from the many performance measurement schemes devised, used and reported by studies. Many of the schemes used by studies highlight different aspects of predictive performance. For example, Menzies et al. [25] use pd and pf to highlight standard predictive performance, while Mende and Koschke [23] use Popt to assess effort-awareness. The different performance measurement schemes used mean that directly comparing the\(^1\)Definitions of particular measures are given in Section Two.
performance reported by individual studies is difficult and potentially misleading. Such comparisons cannot compare like with like as there is no adequate point of comparison.

It is perfectly legitimate for studies to report different performance measures. Studies may be interested in reporting prediction models with particular qualities. Some studies may be interested in reporting models which reduce the amount of effort wasted on code predicted as faulty which turns out not to be faulty. In these cases, measures based on the number of false positives will be of most interest. Other studies may be developing models focused on identifying the maximum number of faults in the system. In which case measures related to the number of true positives are likely to be the performance focus. The qualities needed in a fault prediction model depend on, for example, application domain. Models used in the safety critical domain are likely to need different predictive qualities to those in other domains. However developers and potential users of models may want to compare performance in terms of a particular predictive quality. This requires a conversion of performance figures from those reported to those reflecting the predictive quality of interest. The ability to convert predictive measures in this way allows the predictive performance of a wide range of models to be compared.

We previously found [14] that Precision and Recall were the most commonly reported predictive performance measures used with binary\(^2\) fault prediction models (e.g. [1, 6, 11, 20]). However, many studies provide only limited predictive performance data, often only reporting performance using their preferred performance measures. This preferred data often represents the performance of specific models in the most positive light. An issue also highlighted by Zeller et al. [34]. This preferred measurement data may be unusual and rarely reported in other studies. For example, only a few studies report the use of Error Rate [19, 29, 33] or Popt [23]. Without additional performance data that is more commonly reported by studies, it is difficult to satisfactorily compare the predictive performance of such models. A common point of comparison is needed.

The confusion matrix is usually at the centre of measuring the predictive performance of models (the confusion matrix is discussed in detail in Section Two). Most other predictive performance measures are calculated from the confusion matrix. The confusion matrix is a powerful point of comparative reference. All models reporting binary results can have their predictive performance expressed via a confusion matrix [26]. This means that it is a relatively universal comparative basis. It is also a simple and understandable way to show predictive performance. More sophisticated measures of predictive performance can be calculated from a confusion matrix. The confusion matrix provides measurement flexibility as specific measures may be derived from the confusion matrix which evaluate particular model qualities. The importance of the confusion matrix is discussed in detail by Pizzi et al. [28].

In this paper we present a process by which we transform a variety of reported predictive performance measures back to a confusion matrix [5]. These measures cover most of those reported by the 208 fault prediction studies we previously reviewed [14]. We illustrate this process by constructing the confusion matrix for a number of published models. From these confusion matrices we compute a range of alternative performance measures. We finally evaluate the use of our transformation process.

In Section Two we describe the measurement of predictive performance by discussing in detail the basis of the confusion matrix and related compound measures of performance. In Section Three we present our method of transforming a variety of performance measures to the confusion matrix and explain how alternative measures can then be derived from this matrix. Section Four provides the results of worked examples from the literature in which we transform the reported performance measures back to the confusion matrix. Section Five identifies the threats to the validity of the study. Section Six discusses the implications of transforming performance measures. We conclude and summarise in Section Seven.

2. MEASURING PREDICTIVE PERFORMANCE

This section is based on several previous studies which provide an excellent overview of measuring the predictive performance of fault models (e.g. [26], [16] and [22]).

2.1 The Confusion Matrix

The measurement of predictive performance is often based on the analysis of data in a confusion matrix (see [26]). Pizzi et al. [28] discuss the confusion matrix in more detail. This matrix reports how the model classified the different fault categories compared to their actual classification (i.e. predicted versus observed). This is represented by four pieces of data:

- True Positive (TP): An item is predicted as faulty and it is faulty
- False Positive (FP): An item is predicted as faulty and it is not faulty
- True Negative (TN): An item is predicted as not faulty and it is not faulty
- False Negative (FN): An item is predicted as not faulty and it is faulty

Table 1 shows the structure of a confusion matrix.

<table>
<thead>
<tr>
<th>predicted true</th>
<th>observed true</th>
<th>observed false</th>
</tr>
</thead>
<tbody>
<tr>
<td>predicted false</td>
<td>FN</td>
<td>TN</td>
</tr>
</tbody>
</table>

\(^2\)Binary models are those predicting that code units (e.g. modules or classes) are either fault prone (fp) or not fault prone (nfp). Binary models do not predict the number of faults in code units. In this paper we restrict ourselves to considering only binary models that are based on machine learning techniques.
In a confusion matrix, it is normal for the sum of the instances of each possibility to be reported, see Table 2.

Few studies report complete confusion matrices for their experiments. Studies that do include [27], [36] and [18]. Most studies prefer to report a sub-set of the compound performance measures shown in Table 3.

2.2 Compound Measures

Many performance measures are related to components of the confusion matrix. Table 3 shows how some commonly used performance measures are calculated relative to the confusion matrix.

Table 3 shows that Accuracy is the proportion of units correctly classified. Table 3 also shows that Recall (otherwise known as the true positive rate, probability of detection (pd) or Sensitivity) describes the proportion of faulty code units (usually files, modules or packages) correctly predicted as such. Precision describes how reliable a prediction is in terms of what proportion of code predicted as faulty actually was faulty. Both Recall and Precision are important when test sets are imbalanced (see the following sub-section), but there is a trade-off between these two measures (see [16] for a more detailed analysis of this trade-off). An additional composite measure is the false positive rate (pf) which describes the proportion of erroneously predicted faulty units. The optimal classifier would achieve a pd of 1, Precision of 1, a pf of 0 and an f-measure of 1. The performance measure balance combines pd and pf. A high Balance value (near 1) is achieved with a high pd and low pf. Balance can also be adjusted to factor in the cost of false alarms which typically do not result in fault fixes. Matthews Correlation Coefficient (MCC) is a measure rarely used in software fault prediction [3]. MCC is more commonly used in medical research and bioinformatics e.g. [3, 30]. It is a Chi Square based performance measure on which all four quadrants of the confusion matrix are included in the calculation. MCC results are the equivalent of reporting $R^2$ in regression modelling and results range from -1 to 1 (with 0 indicating random results). Popt defined by Mende and Koschke [23] is a an effort aware performance measure which ranges between 0 and 1 with 1 being desirable.

The Receiver Operator Curve (ROC) is an important measure of predictive performance. When the combinations of Recall and pf for a series of experiments are plotted they produce a ROC. It is usual to report the area under the curve (AUC) as varying between 0 and 1, with 1 being the ideal value. Because the AUC is a result of a series of experiments where the meta-parameters are varied, it is not possible to compute the confusion matrix from AUC and visa versa.

Previous studies have critiqued the use of these various measures of performance. For example, Zhang and Zhang [35], Menzies et al. [24] and Gray et al. [13] discuss the use of precision. However such a critique is beyond the scope of the work reported here.

2.3 Imbalanced Data

Substantially imbalanced data sets are commonly used in fault prediction studies (i.e. there are usually many more non-faulty units than faulty units in the data sets used in fault prediction) [7], [35]. An extreme example of this is seen in the NASA data set PC2, which has only 0.4% of data points belonging to the faulty class (23 out of 5589 data points). This distribution of faulty and non-faulty units has important implications in fault prediction. Imbalanced data can strongly influence the suitability of predictive performance measures. Measures which favour the majority class (such as Accuracy and Error Rate) are not sufficient by themselves [15]. More appropriate measures for imbalanced data sets include: Precision, Recall, f-measure, MCC and G-mean [15]. Consequently data imbalance is an important consideration in our method of recomputing the confusion matrix.

3. OUR METHOD OF RECOMPUTING THE CONFUSION MATRIX

To compare the results of one study with the results of another we recompute the confusion matrix for each study and then calculate the preferred compound measures from this. Zhang and Zhang [35] did something similar to this by re-computing Precision for Menzies et al.’s [25] study which originally reported pd and pf. Our approach is motivated by Zhang and Zhang’s [35] work. We now describe the process by which transformation from a variety of compound measures to the confusion matrix can be achieved.

3.1 Creating a Frequency-Based Confusion Matrix

The precise method needed to recompute the confusion matrix varies slightly depending upon the original measures reported. In most cases the first thing that needs to be done is that we produce a frequency-based confusion matrix. These confusion matrices are different from instance based confusion matrices (an example of which was shown in Table 2). Table 4 shows the frequencies (or proportions for each confusion matrix quadrant) based on the instances in Table 2. These frequencies are derived by dividing the instances in each quadrant by the total number of instances in the matrix. This shows the relative proportion each quadrant represents of the whole confusion matrix. From now on we will append $f$ to $TP$, $TN$, $FP$ and $FN$ to distinguish frequency values from instance based values.

3.2 Calculating Faulty and Non-Faulty Data Distributions

beyond the scope of our work as it is not possible to construct a confusion matrix from AUC data.

$^4$Data imbalance also has serious implications for the training of prediction models. Discussion of this is beyond the scope of this work (instead see [13], [35], [31], [4] and [17]).
Table 3: Compound Performance Measures

<table>
<thead>
<tr>
<th>Measures</th>
<th>Defined As</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy / Correct Classification Rate (CCR)</td>
<td>(\frac{TP + TN}{TP + TN + FP + FN})</td>
</tr>
<tr>
<td>Error Rate</td>
<td>(\frac{FP + FN}{TP + TN + FP + FN})</td>
</tr>
<tr>
<td>Recall / True Positive Rate / Sensitivity / Probability of Detection (pd)</td>
<td>(\frac{TP}{TP + FN})</td>
</tr>
<tr>
<td>True Negative Rate / Specificity</td>
<td>(\frac{TN}{TN + FP})</td>
</tr>
<tr>
<td>False Positive Rate / Type I Error Rate / Probability of False Alarm (pf)</td>
<td>(\frac{FP}{TN + FP})</td>
</tr>
<tr>
<td>False Negative Rate / Type II Error Rate</td>
<td>(\frac{FN}{FN + TP})</td>
</tr>
<tr>
<td>Precision</td>
<td>(\frac{TP}{TP + FP})</td>
</tr>
<tr>
<td>F-Measure / F-Score</td>
<td>(\frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}})</td>
</tr>
<tr>
<td>Balance</td>
<td>(1 - \frac{\sqrt{(0 - pf)^2 + (1 - pd)^2}}{\sqrt{2}})</td>
</tr>
<tr>
<td>G-mean</td>
<td>(\sqrt{\text{Recall} \times \text{Precision}})</td>
</tr>
<tr>
<td>Matthews Correlation Coefficient (MCC)</td>
<td>(\frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}})</td>
</tr>
</tbody>
</table>
Constructing a frequency based confusion matrix is possible when the class distribution (i.e., the proportion of faulty versus non-faulty units) is reported\(^5\). To do this we use \(d\) as the frequency of the faulty units, where:

\[
d = \frac{TN + FN}{TN + TP + FP + FN}
\]

(1)

Applying (1) to the example instances reported in Table 2 would result in:

\[
n = 33 + 17 + 2 + 98, \quad d = \frac{33 + 17}{n} = 0.3333
\]

This shows that given the confusion matrix shown in Table 2, 33% of the units in the data set on which the model was applied, were faulty.

### Table 4: Frequency Confusion Matrix

<table>
<thead>
<tr>
<th>predicted true</th>
<th>observed true</th>
<th>0.2200</th>
</tr>
</thead>
<tbody>
<tr>
<td>observed false</td>
<td>0.1133</td>
<td>0.6533</td>
</tr>
</tbody>
</table>

\[TN_f + TP_f + FP_f + FN_f = 1\]

\[d = 0.2200 + 0.1133 = 0.3333\]

3.3 Transforming Specific Compound Measures

A wide variety of compound measures are reported by studies. Our approach is successful when a particular sub-set of these measures is reported by studies. Table 5 shows the prerequisite combinations of performance measures that must be available.

Each of these combinations of measures requires a specific method by which to recompute the confusion matrix. Formulae for the most common measures reported are now described.

1. **Transforming Precision, Recall and \(pf\)**

   We first need to know the frequency of the true class \(d\).

   \[
   1 = TP_f + TN_f + FP_f + FN_f
   \]

   \[
d = TP_f + FN_f
   \]

   It then becomes possible to calculate \(TP_f, FN_f, TN_f\) and \(FP_f\) as follows:

   Given \(pf\) and \(d\):

   \[
   TN_f = (1 - d)(1 - pf)
   \]

   \[
   FP_f = (1 - d)pf
   \]

   Given \(Recall(r)\) and \(d\):

   \[
   TP_f = d \times r
   \]

   \[
   FN_f = d(1 - r)
   \]

\(^5\)When this class distribution is not provided it is often possible to calculate the proportion of faulty units in a data set.

Given \(FNR(TypeII(t2))\), \(pf\) and \(d\) we already have (2), (4) and (5)

\[
FN_f = t2 \times d
\]

(8)

\[
TP_f = 1 - FN_f - TN_f - FP_f
\]

(9)

Given \(Precision(p), Recall(r)\) and \(d\) we already have (2), (6) and (7)

\[
FP_f = \frac{TP_f(1 - p)}{p} = \frac{d(1 - p)r}{p}
\]

(10)

\[
TN_f = 1 - FP_f - FN_f - TP_f
\]

(11)

2. **Transforming ErrorRate(\(er\)), TypeII(\(t2\)) and \(pf\)**

\[
d = \frac{er - pf + pf \times er}{t2}
\]

(12)

which can then be used with (4),(5),(8) and (9)

3. **Transforming Precision(\(p\)), Recall(\(r\)) and Accuracy(\(a\))**

\[
d = \frac{p(1 - a)}{p - 2pr + r}
\]

(13)

which can then be used with (6),(7),(10) and (11)

4. **Transforming Accuracy(\(a\)), \(pf\) and \(FNR(TypeII(t2))\)**

\[
er = 1 - a
\]

(14)

\[
d = \frac{er - pf}{t2 - pf}
\]

(15)

which can be used with (8) to give \(FN_f\) and (5) to give \(FP_f\).

\[
TP_f = d(1 - t2)
\]

(16)

which can be used with (11) to give \(TN_f\).

We have automated these conversations by developing a tool\(^6\). This tool allows individual performance measurement data to be input and will automatically recompute the confusion matrix by iterating over the the equations until no extra performance measures can be derived.

4. Constructing the Confusion Matrix for Some Example Studies

We have transformed the predictive performance data produced by 601 models reported in 42 published studies. A list of these studies is provided in the Appendix. Space restrictions make it impossible to report the detail for all these transformations. Consequently in this section we present transformations for four examples. We chose these four examples to illustrate recomputing the confusion matrix from a range of different original measures.

4.1 Case Studies

Table 6 illustrates the original performance measurement data reported by our four case study papers. Table 6 shows that a wide range of different measurement data is reported.

---

\(^6\)This tool is available at: https://bugcatcher.stca.herts.ac.uk/JConfusion/
Table 5: Pre-Requisite Combinations of Performance Measures for Re-Computing the Confusion Matrix.

<table>
<thead>
<tr>
<th>Fault Frequency</th>
<th>Type I</th>
<th>Type II</th>
<th>Precision</th>
<th>Recall</th>
<th>Accuracy</th>
<th>pf</th>
<th>Error Rate</th>
<th>Specificity</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

NB this is not an exhaustive list. For example, it is possible to calculate $d$ by dividing the number of defective instances by the total number of instances.

Table 6: Reported Performance Measurement Data

<table>
<thead>
<tr>
<th>Study</th>
<th>pd</th>
<th>pf</th>
<th>Error Rate</th>
<th>Type I</th>
<th>Type II</th>
<th>Precision</th>
<th>Recall</th>
<th>Accuracy</th>
<th>Total Instances</th>
<th>Faulty Instances</th>
</tr>
</thead>
<tbody>
<tr>
<td>[19]</td>
<td>0.3134</td>
<td>0.3127</td>
<td>0.2826</td>
<td></td>
<td></td>
<td>0.682</td>
<td>0.621</td>
<td>0.641</td>
<td>520</td>
<td>106</td>
</tr>
<tr>
<td>[6]</td>
<td>0.471</td>
<td>0.0834</td>
<td>0.1615</td>
<td>0.1304</td>
<td>0.2830</td>
<td>0.8515</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[21]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[29]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

by these four papers. Given this range it is difficult to evaluate how the performance of these models compares against each other.

We have recomputed the confusion matrix for these four case studies (shown in Table 7). Based on this confusion matrix data, we have computed the f-measure and MCC data for each case study (also shown in Table 7). It is now possible to comparatively evaluate the predictive performance of these case studies using this common set of data.

5. THREATS TO VALIDITY

There are internal and external validity issues that need to be considered when using our approach to recomputing the confusion matrix.

5.1 Internal Validity

The impact of cross-validation. Performance data reported are usually based on some form of cross-validation. This means that the numbers reported are usually average figures across a specific number of folds and / or experiments. This averaging process may introduce some minor inaccuracies into our calculations. The study by Elish and Elish [12] provide a clearer understanding of the variation in performance values across a series of experiments. The results of rounding errors and variations in performance values helps to explain the negative $TN_f$ value computed in Table 11.

Divide by zero problems. Several of our formulas are based on divisions. Where some figures are very similar we encounter divide by zero problems. This division problem is exacerbated by rounding of very small numbers. These small numbers may be very different, but when rounded become the same number. Such numbers suffer from divide by zero issues.

Data uncertainty. Identifying the balance of faulty and non-faulty units is an important part of our recomputing method. However in a few studies there is inconsistency in the class distribution figures. For example, although an author may have cited a particular class distribution, when we calculate the distribution figure inherent within the results reported (i.e. via the calculation of $d$), the distribution is different to that stated by the authors. Similarly in some papers where the same data set has been used the distribution varies between experiments. This inconsistency casts some uncertainty over the results in such cases. We suspect that this distribution inconsistency is partly the result of a particular machine learner dealing with the data that it is processing differently to other learners, and partly the result of studies not reporting the data pre-processing that they have applied.

5.2 External Validity

Model tuning. Some models may have been developed to maximise a particular quality (e.g. to reduce false positives). Such models are likely to perform best when their performance is expressed using measures that are sympa-
thletic to the qualities for which the model has been built. Interpreting the performance of such models via alternative performance measures should be treated with caution.

6. DISCUSSION
The process of translating the performance measures reported by studies to the confusion matrix reveals a variety of performance issues with studies that we now discuss.

6.1 Erroneous Results
In some cases our translation to the confusion matrix demonstrated that the original results reported by some studies could not have been possible. For example we found an error in [32]. This error was revealed as our transformations would not work correctly. As a result of this we emailed the authors to clarify the problem. The authors confirmed that a typographical error had crept into their final draft. False Alarms were reported instead of False Positives. It is easy for such errors to creep into published work, especially in an area as complex as fault prediction. Without very careful interpretation such errors can easily be missed and be misleading.

6.2 Definitions of Measures
While performing our transformations we have had difficulty in making sense of the figures reported in some studies. The reason for this was that a number of studies have used non-standard definitions for some well-known performance measures (e.g. [36] does not use a standard definition of Precision and [27] does not use the standard definitions of Sensitivity and Specificity (in both cases, the issues were confirmed by emailing the authors)). Although the definitions used were given in the paper, it is difficult for a reader to pick-up on the nuances of measurement definitions (usually provided via formulae). Consequent mis-understanding could have serious implications for subsequent model users.

6.3 Reporting Performance Based on Predicting Non-Faulty Units
Some papers have reported performance measures based on predicting the majority (non-faulty class) rather than the minority (faulty) class. In some of these cases it is also not made clear that the predictive performance is on the majority class. These issues can be very misleading when trying to evaluate predictive performance. For example, Elish and Elish [12] report a very influential fault prediction study using Support Vector Machines (SVM). Their study has been cited more than 60 times and is considered a pivotal paper referring to Support Vector Machines (SVM). Their study has been cited more than 60 times and is considered a pivotal paper.

Our process to recompute the confusion matrix would not work on these figures when we assumed that the values for Precision and Recall were based on the non-faulty class. Tables 9 and 10 show our workings for this recomputation. Our workings suggest that Elish and Elish have reported the performance of their SVM models based on predicting non-faulty units rather than faulty units. Since the vast majority of units in data sets are non-faulty (ranging between 84.6% and 93.7% in their case), predicting the majority class is very easy and so high performance figures are to be expected. Such models are not useful. Our findings are complementary to those of several other authors who report problems reproducing the high predictive performances reported by Elish and Elish when using their SVM settings. For example [2] reports that most papers report a far lower Recall value. [9] and [10] used the same SVM settings. [9] reported Specificity and Sensitivity values and [10] reported Precision and Recall for both the faulty and non-faulty classes which are similar to our recomputed values. Despite emailing Elish and Elish several times we were unable to get a reply confirming the basis of their performances.

Using our technique it is possible to calculate the Precision and Recall of the faulty units in Elish and Elish’s study. Table 11 shows the results of this calculation. Table 11 suggests that the performance of the SVMs in the Elish and Elish study is much less positive. Table 11 shows that f-measure ranges from 0.0 to 0.12. This is compared to their original maximum f-measure of 0.96.

<table>
<thead>
<tr>
<th>Study</th>
<th>TP</th>
<th>TN</th>
<th>FP</th>
<th>FN</th>
<th>f-measure</th>
<th>MCC</th>
</tr>
</thead>
<tbody>
<tr>
<td>[19]</td>
<td>0.0163</td>
<td>0.6710</td>
<td>0.3063</td>
<td>0.0064</td>
<td>0.0944</td>
<td>0.1288</td>
</tr>
<tr>
<td>[6]</td>
<td>0.3335</td>
<td>0.3075</td>
<td>0.1555</td>
<td>0.2035</td>
<td>0.6501</td>
<td>0.2845</td>
</tr>
<tr>
<td>[21]</td>
<td>0.0575</td>
<td>0.7940</td>
<td>0.0732</td>
<td>0.0646</td>
<td>0.4549</td>
<td>0.3755</td>
</tr>
<tr>
<td>[29]</td>
<td>0.1422</td>
<td>0.6963</td>
<td>0.1000</td>
<td>0.0615</td>
<td>0.6377</td>
<td>0.5381</td>
</tr>
</tbody>
</table>

Table 7: Computed Performance Measurement Data

<table>
<thead>
<tr>
<th>Dataset</th>
<th>TP</th>
<th>FN</th>
<th>FP</th>
<th>TN</th>
</tr>
</thead>
<tbody>
<tr>
<td>cm1</td>
<td>0.9037</td>
<td>0.0000</td>
<td>0.0931</td>
<td>0.0032</td>
</tr>
<tr>
<td>pc1</td>
<td>0.9261</td>
<td>0.0049</td>
<td>0.0641</td>
<td>0.0049</td>
</tr>
<tr>
<td>kc1</td>
<td>0.8412</td>
<td>0.0047</td>
<td>0.1490</td>
<td>0.0051</td>
</tr>
<tr>
<td>kc3</td>
<td>0.9330</td>
<td>0.0039</td>
<td>0.0633</td>
<td>-0.0002*</td>
</tr>
</tbody>
</table>

* demonstrates that rounding errors occur.

Table 10: SVM Confusion Matrix of the Majority Class

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>f-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>cm1</td>
<td>0.9069</td>
<td>1.0000</td>
<td>0.0332</td>
<td>0.0643</td>
</tr>
<tr>
<td>pc1</td>
<td>0.9310</td>
<td>0.5000</td>
<td>0.0710</td>
<td>0.1244</td>
</tr>
<tr>
<td>kc1</td>
<td>0.8463</td>
<td>0.5204</td>
<td>0.0331</td>
<td>0.0622</td>
</tr>
<tr>
<td>kc3</td>
<td>0.9328</td>
<td>-0.0541</td>
<td>-0.0032</td>
<td>-0.0060</td>
</tr>
</tbody>
</table>

Table 11: Performance Measures for the Faulty Class using the Values from Table 10
Table 8: Accuracy, Precision and Recall [12]

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>cm1</td>
<td>0.9069</td>
<td>0.9066</td>
<td>1.0000</td>
</tr>
<tr>
<td>pc1</td>
<td>0.9310</td>
<td>0.9353</td>
<td>0.9947</td>
</tr>
<tr>
<td>kc1</td>
<td>0.8459</td>
<td>0.8495</td>
<td>0.9940</td>
</tr>
<tr>
<td>kc3</td>
<td>0.9328</td>
<td>0.9365</td>
<td>0.9958</td>
</tr>
</tbody>
</table>

Table 9: Frequency of the Class Identified as “True” and the Frequency of the Faulty class

<table>
<thead>
<tr>
<th>Dataset</th>
<th>computed d</th>
<th>1 - computed d</th>
<th>Reported fault frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>cm1</td>
<td>0.9037</td>
<td>0.0963</td>
<td>0.097</td>
</tr>
<tr>
<td>pc1</td>
<td>0.9311</td>
<td>0.0689</td>
<td>0.069</td>
</tr>
<tr>
<td>kc1</td>
<td>0.8462</td>
<td>0.1538</td>
<td>0.154</td>
</tr>
<tr>
<td>kc3</td>
<td>0.9370</td>
<td>0.0630</td>
<td>0.063</td>
</tr>
</tbody>
</table>

7. CONCLUSION

The predictive performance of published fault prediction models is expressed using a variety of different performance measures. This makes it difficult to compare the performance of published prediction models. We have presented an approach that enables the recomputation of the confusion matrix for studies originally reporting a variety of performance measures. From the confusion matrix a range of other performance measures can be calculated. Expressing the performance of fault prediction models using a consistent set of measures allows comparative analysis. Our approach has several advantages, including that it:

- allows comparative analysis of a set of fault prediction models in terms of a preferred predictive quality.
- makes meta-analysis possible across the many fault prediction studies published.
- enables the validation of the performance figures reported in published studies.

The advantages of our approach have benefits for fault prediction researchers, practitioners and reviewers. Researchers can use our approach to evaluate predictive performance across sets of models and perform meta-analysis of these models. An evidence base of fault prediction can be built by researchers that will enable more informed future model building research. Practitioners can express model performance to reflect the qualities that they are interested in, for example practitioners wanting a model that values finding as many faults as possible might predominantly focus on Recall. Practitioners are then in a more informed position to select a model that is appropriate for their development context. Reviewers of fault prediction studies can use our process as a relatively easy way to check that no errors have crept into fault prediction studies. Without our ‘ready reckoner’ checking performance figures in studies submitted for review is difficult. Model builders could themselves use our process as a ‘ready reckoner’ to check their own figures are correct. Model builders and reviewers doing this checking could improve the quality of the fault prediction work that is published.

Overall the approach that we present could significantly improve the quality of fault prediction studies and enable meta-analysis across studies. Achieving this is very important as it will help this research area to mature and grow. Such maturation could ultimately expand the industrial uptake of fault prediction modelling.

8. REFERENCES

APPENDIX

List of Papers from which we have Recomputed Confusion Matrices.


