DATA MINING FOR SOFTWARE DEFECT PREDICTION WITH OPEN SOURCE AND COMMERCIAL SOFTWARES

OLUFEMI SUNDAY ADEOYE
PhD Student, Information Management Technology Department, Federal University of Technology, Owerri, Nigeria

ABSTRACT

There has been an emerging open source data mining software development in current times. In this work I carry out an empirical study to compare the performances of open source and commercial data mining software. The results clearly reveals that before and after data preprocessing of the two NASA datasets used in this work, commercial data mining software (MATLAB) outperforms open source data mining software (WEKA) in overall accuracy, probability of false alarm, Area Under ROC Curve and Probability of Detection.

Keywords: Data Mining, Software Defect Prediction, Open Source Software, Commercial Software, Confusion Matrix.

1.0 Introduction

With the emergent of open source data mining software in recent times, it is important to do an empirical study in order to know how well open source data mining software perform in classification problems. This study compares the performance of open source data mining tools with that of commercial data mining software to determine which one should be deployed for safety critical systems.

The term “open source” refers to something that can be modified because its design is publicly accessible. It is software whose source code is available for modification or enhancement by anyone. In general, open source projects, products, or initiatives are those that embrace and celebrate open exchange, collaborative participation, rapid prototyping, transparency, meritocracy, and community development. Open source software can be downloaded freely on the internet. “Source Code” is the part of software that most computer users don’t ever see: it is the code computer programmers can use to change how a piece of software works. Programmers who have access to a computer program’s source code can improve that program by adding features to it or fixing parts that do not always work correctly. Some notable open source data mining tools are:

i. Scikit-learn: Scikit-learn is an open source machine learning library for the python programming language.

ii. KNIME – Is an open source data analytics, reporting and integration platform that integrates various components for machine learning and data mining.

iii. OpenNN – Is a software library which is able to learn from both datasets and mathematical models. The software can be used to solve pattern recognition problems.

iv. Orange – It is a component-based data mining and machine learning software suite. It includes a set of components for data preprocessing, modeling, model evaluation, and exploration techniques.

v. R - It is a free software programming language and software environment for statistical computing and graphics.
vi. **RapidMiner** – Is a software platform that provides an integrated environment for machine learning, data mining, text mining, predictive analytics and business analytics.

vii. **WEKA** – WEKA is a popular suite of machine learning software written in Java, developed at the University of Waikato, New Zealand. It contains a collection of visualization tools and algorithms for analysis and predictive modeling.

viii. **Apache Mahout** – It is a project to produce free implementation of distributed or otherwise scalable machine learning algorithms focused primarily in the areas of collaborative filtering, clustering and classification [1], [4].

On the other hand, some softwares have source code that cannot be modified by anyone but the person, team, or organization that created it. This kind of software is frequently called “proprietary software” or “commercial software” or “closed source” software because its source code is the property of its original authors, who are the only ones legally allowed to copy or modify it.

Notable commercial predictive analytic tools are:

i. **Alpine Data Labs** – This is an advanced analytics interface working with Apache Hadoop and big data.

ii. **BIRT Analytics** – Is a self service predictive analytics tool that allows non-technical business users to engage in visual data mining with Big Data from multiple sources without IT help.

iii. **IBM SPSS Modeler** – Is a data mining and text analytics software application built by IBM. It is used to build predictive and conduct other analytic tasks.

iv. **Mathematica** – Is a computational software program used in many scientific, engineering, mathematical and computing fields, based on symbolic mathematics.

v. **MATLAB** – MATLAB is a multi-paradigm numerical computing environment and fourth generation programming language. Matlab offers features to develop algorithms, and create models and applications.

vi. **Minitab** – Minitab is a statistics package developed at the Pennsylvania state university. It helps us to analyze our data and improve the products and services with the loading statistical software used for quality improvement.

vii. **Oracle Data Mining (ODM)** – Oracle data mining provides powerful data mining functionality as a native SQL functions within the oracle database. The oracle spreadsheet add-in for predictive analytics provides predictive analytics operations.

viii. **STATISTICA** – Is a statistics and analytics software package developed by StatSoft. STATISTICA provides data analysis, data management, statistics, data mining, and data visualization procedures [2], [3].

### 2.0 Methodology

In this work, data mining techniques and their applications for classifications was adapted to determine the differences in classification ability of open source and commercial data mining softwares. This study makes use of two NASA datasets CM1 and PC1 which are available in the PROMISE repository. The classification algorithms used is Multilayer Perceptron. This is used to classify the software modules using McCabe and Halstead static software features. Experiments were carried out using both imbalanced and balanced datasets. The data mining tools used are WEKA and MATLAB. Performance of the data mining technique (multilayer perceptron) is compared through the two data mining tools. The results were evaluated using the following metrics: overall accuracy, probability of detection (PD), probability of false alarm (PF), Area Under ROC curve (AUC), and Balance.
3.0 Evaluation Criteria

3.1 Confusion Matrix

Typically, the performance of a binary prediction model is summarized through confusion matrix.

<table>
<thead>
<tr>
<th>Predicted Defective</th>
<th>Predicted Defect Free</th>
</tr>
</thead>
<tbody>
<tr>
<td>Observed Defective</td>
<td>True Positive (TP)</td>
</tr>
<tr>
<td></td>
<td>False Negative (FN)</td>
</tr>
<tr>
<td>Observed Defect Free</td>
<td>False Positive (FP)</td>
</tr>
<tr>
<td></td>
<td>True Negative (TN)</td>
</tr>
</tbody>
</table>

Table 1 Confusion Matrix

True prediction (True Positive (TP) or True Negative (TN)) refers to the number of software modules which are correctly predicted as defective or non-defective software modules. While the False prediction (False Positive (FP) or False Negative (FN)) indicates the number of software modules which are incorrectly recognized as defective or non-defective software modules. A confusion matrix contains information about actual and predicted classifications done by a classification system. The performances of the models produced by the data mining tools are evaluated by using overall accuracy, sensitivity, probability of false alarm, Area Under ROC Curve, and Balance.

3.2 Overall Accuracy

The accuracy measures the chances of correctly predicting the fault proneness of individual’s modules. It ignores the data distribution and cost information. Therefore, it can be a misleading criterion as faulty modules are likely to represent a minority of the modules in the dataset.

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN}$$

3.3 Probability of Detection (PD)

This is also referred to as recall, true positive rate or sensitivity. It is the proportion of correctly predicted defective modules.

$$PD = \frac{TP}{TP + FN}$$

3.4 Probability of False Alarm (PF)

It is the ratio of the example that is negative but is classified wrongly as positive to the number of examples that are actually negative.

$$PF = \frac{FP}{FP + TN}$$

3.5 Area Under ROC Curve (AUC)

The area under the ROC curve, referred to as AUC, is a numeric performance evaluation measure directly associated with an ROC curve. AUC, also known as the C-statistic, measures the predictive power of a binary classification model. A good model’s ROC curve should sit close to the upper left corner and the area under the ROC curve should be close to 1.

$$AUC = \frac{1 + TPR - FPR}{2} \text{ or } \frac{TPR + TNR}{2}$$

The ROC curve plots the probability of detection on the y-axis and the probability of a false alarm on the x-axis.

3.6 Balance

Balance is defined as the normalized Euclidean distance from the desired point (0, 1) to (pf, pd) in a Receiver Operating Characteristic (ROC) curve. Balance combines pf and pd into one measure and is defined as the distance from the ROC ‘sweet spot’.

$$Balance = 1 - \sqrt{\left(0 - pf\right)^2 + \left(1 - pd\right)^2}$$
4.0 Experiments and Results

4.1 Datasets

The datasets used in this study (CM1 and PC1) is an embedded data from a PROMISE software engineering repository dataset made publicly available online in order to encourage repeatable, refutable, verifiable, and/or improvable predictive models of software engineering.

Data comes from McCabe and Halstead features extractors of source code. The McCabe and Halstead measures are “module” – based where a “module” is the smallest unit of functionality. (http://promise.site.uoltawa.ca/SERepository/datasets-page.html)

Table 2 Description of the Imbalanced Datasets

<table>
<thead>
<tr>
<th>Datasets</th>
<th># Instances</th>
<th>Def, Non-Def</th>
<th># Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>CM1</td>
<td>498</td>
<td>449, 49</td>
<td>22</td>
</tr>
<tr>
<td>PC1</td>
<td>1109</td>
<td>1032, 77</td>
<td>22</td>
</tr>
</tbody>
</table>

Table 3 Description of the Balanced Datasets

<table>
<thead>
<tr>
<th>Datasets</th>
<th># Instances</th>
<th>Def, Non-Def</th>
<th># Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>CM1</td>
<td>898</td>
<td>449, 449</td>
<td>22</td>
</tr>
<tr>
<td>PC1</td>
<td>2064</td>
<td>1032, 1032</td>
<td>22</td>
</tr>
</tbody>
</table>

The total number of attributes (22) consists of twenty one (21) predictor variables and one (1) response variable.

4.2 Experimental Environment

This experiment was conducted in the following hardware and software environment.

Processor: HP Mini 200 computer with Intel® Atom™ CPU N2600@ 1.60GHz processor.

Memory 2:00 GB RAM

Operating System 32 bit Windows 7

Softwares: WEKA Version 3.7.11 and MATLAB Version R2007b

4.3 Data Preparation for MATLAB Code

Before analyzing the training datasets, we need to modify the raw data files provided at PROMISE NASA repository in order to build the Neural Network properly. In the software defect prediction, the answers field (defect) in the training dataset displays a string value of either “True” or “False”. I have converted the “true” values to 1 and the “false” values to 0 to allow MATLAB recognize the answers as binary Boolean outputs.

4.4 Building the Neural Network Classifier

The next step is to create a neural network. A 2-hidden layer feed forward network is created with 8 neurons. Now the network is ready to be trained.

4.5 Testing the Classifier

The trained neural network can now be tested with the test samples. Before predicting the output the training of the neural network is required.

Fig 1 Training of Neural Network

In figure 1 neural network is trained in which 1 input layer, 2 hidden layers and 1 output layer is used.
Fig 2 shows the input to the network with 21 inputs to the network and 2 outputs. Inputs are given from the testing data. The neural network is trained using training data from the dataset.

### 4.6 Results

Table 4 Confusion Matrix Before and After Preprocessing for CM1 dataset

<table>
<thead>
<tr>
<th>DATA MINING TOOL</th>
<th>BEFORE PROCESSING</th>
<th>AFTER PROCESSING</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>T</td>
<td>N</td>
</tr>
<tr>
<td>WEKA</td>
<td>43</td>
<td>1</td>
</tr>
<tr>
<td>MATLAB</td>
<td>44</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 5 Confusion Matrix Before and After Preprocessing for PC1 dataset

<table>
<thead>
<tr>
<th>DATA MINING TOOL</th>
<th>BEFORE PROCESSING</th>
<th>AFTER PROCESSING</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>T</td>
<td>N</td>
</tr>
<tr>
<td>WEKA</td>
<td>102</td>
<td>2</td>
</tr>
<tr>
<td>MATLAB</td>
<td>103</td>
<td>1</td>
</tr>
</tbody>
</table>

### 4.7 Evaluator Metrics Results

Table 6 CM1 before Pre-Processing

<table>
<thead>
<tr>
<th>DATA MINING TOOL</th>
<th>ACCURACY</th>
<th>PD</th>
<th>PF</th>
<th>AUC</th>
<th>BALANCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>WEKA</td>
<td>87.6</td>
<td>0.0</td>
<td>0.0</td>
<td>0.4</td>
<td>0.307</td>
</tr>
<tr>
<td>MATLAB</td>
<td>91.0</td>
<td>0.0</td>
<td>0.5</td>
<td>0.351</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 1 PD, PF, AUC, and Balance of WEKA and MATLAB for CM1 compared before preprocessing.
Fig. 2 Accuracy of WEKA and MATLAB of CM1 compared before preprocessing

Table 7 CM1 after Pre-Processing

<table>
<thead>
<tr>
<th>DATA MINING TOOL</th>
<th>ACCURACY</th>
<th>PD</th>
<th>PF</th>
<th>AUC</th>
<th>BALANCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>WEKA</td>
<td>93.5</td>
<td>0.8</td>
<td>0.9</td>
<td>0.9</td>
<td>0.923</td>
</tr>
<tr>
<td>MATLAB</td>
<td>94.8</td>
<td>0.9</td>
<td>0.9</td>
<td>0.9</td>
<td>0.935</td>
</tr>
</tbody>
</table>

Fig. 3 PD, PF, AUC, and Balance of WEKA and MATLAB for CM1 compared after preprocessing

Table 8 PC1 before Pre-Processing

<table>
<thead>
<tr>
<th>DATA MINING TOOL</th>
<th>ACCURACY</th>
<th>PD</th>
<th>PF</th>
<th>AUC</th>
<th>BALANCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>WEKA</td>
<td>93.6</td>
<td>0.1</td>
<td>0.0</td>
<td>0.5</td>
<td>0.403</td>
</tr>
<tr>
<td>MATLAB</td>
<td>93.9</td>
<td>0.1</td>
<td>0.0</td>
<td>0.5</td>
<td>0.394</td>
</tr>
</tbody>
</table>

Fig. 4 Accuracy of WEKA and MATLAB compared for CM1 after preprocessing

Fig. 5 PD, PF, AUC, and Balance of WEKA and MATLAB for PC1 compared before preprocessing
Critical analysis of the result in tables 4 through 9 reveal that commercial data mining tools perform better than open source data mining tool. Commercial data mining tool performs better with high accuracy, probability of detection, Area under (ROC) curve and Balance except in the dataset PC1 before data preprocessing where open source data mining tool perform better in probability of detection, Areas under (ROC) curve and Balance. It could also be observed that commercial data mining tool perform better by producing a low probability of false alarm which will automatically lead to reduced rework cost as compared with open source data mining tool.

5.0 Conclusion

The result from this work reveals that commercial data mining tools perform better than open source data mining tools. Therefore, it is advisable that commercial data mining tools be deployed in any safety-critical system where high accurate prediction is mandatory.

References


