View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Universiti Teknikal Malaysia Melaka (UTeM) Repository

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Faculty of Information and Communication Technology

SOFTWARE DEFECT PREDICTION FRAMEWORK BASED ON HYBRID METAHEURISTIC OPTIMIZATION METHODS

Romi Satria Wahono

Doctor of Philosophy

2015

© Universiti Teknikal Malaysia Melaka

SOFTWARE DEFECT PREDICTION FRAMEWORK BASED ON HYBRID METAHEURISTIC OPTIMIZATION METHODS

ROMI SATRIA WAHONO

A thesis submitted
in fulfillment of the requirements for the degree of Doctor of Philosophy

Faculty of Information and Communication Technology

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2015

© Universiti Teknikal Malaysia Melaka

DECLARATION

I declare that this thesis entitled "Software Defect Prediction Framework based on Hybrid Metaheuristic Optimization Methods" is the result of my own research except as cited in the references. The thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature :

Name : Romi Satria Wahono

Date :

© Universiti Teknikal Malaysia Melaka

APPROVAL

I hereby declare that I have read this thesis and in my opinion this thesis is sufficient in terms of scope and quality for the award of Doctor of Philosophy.

Signature :

Supervisor Name : Prof. Dr. Nanna Suryana Herman

Date :

© Universiti Teknikal Malaysia Melaka

ABSTRACT

A software defect is an error, failure, or fault in a software that produces an incorrect or unexpected result. Software defects are expensive in quality and cost. The accurate prediction of defect-prone software modules certainly assist testing effort, reduce costs and improve the quality of software. The classification algorithm is a popular machine learning approach for software defect prediction. Unfortunately, software defect prediction remains a largely unsolved problem. As the first problem, the comparison and benchmarking results of the defect prediction using machine learning classifiers indicate that, the poor accuracy level is dominant and no particular classifiers perform best for all the datasets. There are two main problems that affect classification performance in software defect prediction: noisy attributes and imbalanced class distribution of datasets, and difficulty of selecting optimal parameters of the classifiers. In this study, a software defect prediction framework that combines metaheuristic optimization methods for feature selection and parameter optimization, with meta learning methods for solving imbalanced class problem on datasets, which aims to improve the accuracy of classification models has been proposed. The proposed framework and models that are are considered to be the specific research contributions of this thesis are: 1) a comparison framework of classification models for software defect prediction known as CF-SDP, 2) a hybrid genetic algorithm based feature selection and bagging technique for software defect prediction known as GAFS+B, 3) a hybrid particle swarm optimization based feature selection and bagging technique for software defect prediction known as PSOFS+B, and 4) a hybrid genetic algorithm based neural network parameter optimization and bagging technique for software defect prediction, known as NN-GAPO+B. For the purpose of this study, ten classification algorithms have been selected. The selection aims at achieving a balance between established classification algorithms used in software defect prediction. The proposed framework and methods are evaluated using the state-of-the-art datasets from the NASA metric data repository. The results indicated that the proposed methods (GAFS+B, PSOFS+B and NN-GAPO+B) makes an impressive improvement in the performance of software defect prediction. GAFS+B and PSOFS+B significantly affected on the performance of the class imbalance suffered classifiers, such as C4.5 and CART. GAFS+B and PSOFS+B also outperformed the existing software defect prediction frameworks in most datasets. Based on the conducted experiments, logistic regression performs best in most of the NASA MDP datasets, without or with feature selection method. The proposed methods also generated the selected relevant features in software defect prediction. The top ten most relevant features in software defect prediction include branch count metrics, decision density, halstead level metric of a module, number of operands contained in a module, maintenance severity, number of blank LOC, halstead volume, number of unique operands contained in a module, total number of LOC and design density.

© Universiti Teknikal Malaysia Melaka

ABSTRAK

Kecacatan perisian merupakan suatu ralat, kegagalan atau kesilapan dalam perisian yang menghasilkan keputusan tidak tepat atau di luar jangkaan. Kecacatan perisian menjejaskan kualiti dan melibatkan kos yang tinggi. Ramalan tepat melalui modul perisian yang terdedah kepada kecacatan membantu usaha pengujian, mengurangkan kos dan meningkatkan kualiti perisian. Algoritma klasifikasi merupakan pendekatan pembelajaran mesin yang terkenal untuk ramalan kecacatan perisian. Namun, ramalan kecacatan perisian masih kekal sebagai sebahagian daripada masalah yang tidak dapat diselesaikan. Perbandingan dan penandaarasan keputusan ramalan kecacatan menggunakan pengelas pembelajaran mesin menunjukkan bahawa tahap ketepatan adala lemah, tidak ada perbezaan prestasi yang ketara dapat dikesan di antara pengelas, dan tiada pengelas tertentu yang mempunyai pelaksanaan terbaik bagi kesemua set data. Di samping itu, terdapat dua masalah utama yang mampu menjejaskan prestasi pengelasan dalam ramalan kecacatan perisian: atribut hingar dan ketidakseimbangan kelas taburan set data, serta kesulitan pemilihan parameter optimum pengelas. Dalam kajian ini, suatu rangka kerja diwujudkan bagi perisian ramalan kecacatan yang menggabungkan kaedah pengoptimuman metaheuristik bagi pemilihan ciri dan pengoptimuman parameter, dengan kaedah meta learning bagi menyelesaikan masalah ketidakseimbangan kelas pada set data, yang bertujuan untuk meningkatkan ketepatan klasifikasi model. Rangka kerja dan model yang dicadangkan dianggap sebagai sumbangan besar dalam tesis ini ialah: 1) Rangka kerja perbandingan model klasifikasi untuk ramalan kecacatan perisian, yang dikenali sebagai CF-SDP, 2) Algoritma genetik hibrid berasaskan pemilihan ciri dan teknik pembungkusan bagi ramalan kecacatan perisian, yang dipanggil GAFS+B, 3) Pengoptimuman kumpulan zarah hibrid berdasarkan pemilihan ciri dan teknik pembungkusan untuk ramalan kecacatan perisian, yang dipanggil PSOFS+B, dan 4) Algoritma genetik hibrid berasaskan pengoptimuman jaringan saraf dan teknik pembungkusan bagi perisian ramalan kecacatan, yang dipanggil NN-GAPO+B. Bagi tujuan kajian ini, sepuluh algoritma klasifikasi telah dipilih. Pemilihan ini bertujuan bagi mencapai keseimbangan antara algoritma pengelasan yang telah mantap yang digunakan dalam ramalan kecacatan perisian. Rangka kerja dan kaedah yang dicadangkan dinilai menggunakan set data terkini dari repositori data metrik NASA. Keputusan menunjukkan bahawa kaedah yang dicadangkan (GAFS+B, PSOFS+B dan NN-GAPO+B) menghasilkan peningkatan yang memberangsangkan dalam pelaksanaan perisian ramalan kecacatan. GAFS+B dan B+PSOFS terjejas dengan ketara kepada pengelas yang mengalami ketidakseimbangan kelas, seperti C4.5 dan CART. GAFS+B dan B+PSOFS juga mengatasi rangka kerja ramalan kecacatan perisian yang sedia ada dalam kebanyakan set data. Berdasarkan eksperimen yang dijalankan, regresi logistik melakukan terbaik dalam kebanyakan dataset NASA MDP, tanpa atau dengan kaedah pemilihan ciri. Kaedah yang dicadangkan juga menghasilkan ciri-ciri yang berkaitan yang dipilih dalam ramalan kecacatan perisian. Sepuluh ciri yang paling relevan dalam ramalan kecacatan perisian termasuk metrik kiraan cawangan, ketumpatan keputusan, metrik tahap halstead dalam modul, operan yang terkandung dalam modul, keterukan penyelenggaraan, jumlah LOC kosong, jumlah halstead, bilangan operan unik yang terkandung dalam modul, jumlah LOC dan kepadatan reka bentuk.

© Universiti Teknikal Malaysia Melaka

ACKNOWLEDGEMENTS

First and foremost, I would like to express my sincere thanks to my supervisor, Professor Dr. Nanna Suryana Herman, for believing in me, also for the encouragement, support, and valuable advice during my PhD program at Faculty of Information and Communication Technology, Universiti Teknikal Malaysia Melaka. His vision and unfailing guidance at every stage has helped me in overcoming many challenges, avoiding possible dead ends all while conducting high quality research.

My second acknowledgement goes to my co-supervisor, Dr. Sabrina Ahmad, for her kind advice and useful comments. It has been a pleasure working with her and learning the different ways of doing and thinking about research.

I am also grateful to all the people that I had the pleasure to meet during my staying at Melaka: Mr. Affandy, Mr. Daniel Hartono, Dr Fahmi Arif, Mr. Sriyanto Balam and all my fellow postgraduate students for the discussions, support, and friendships.

Last but not least, I would like to thank my wife (Sulihtiani Wulandari), my sons (Irsyad, Hasan, Mahdan), my daughters (Yuka, Nana, Azka, Damia), and my parents for all the support, spirits, encouragement, and inspiration.

© Universiti Teknikal Malaysia Melaka

TABLE OF CONTENTS

	PAGE
ABSTRACT	**i**
ABSTRAK	**ii**
ACKNOWLEDGEMENTS	**iii**
TABLE OF CONTENTS	**iv**
LIST OF TABLES	**viii**
LIST OF FIGURES	**x**
LIST OF APPENDICES	**xiii**
LIST OF ABBREVIATIONS AND GLOSSARY	**xiv**
LIST OF PUBLICATIONS	**xxvi**
CHAPTER	
1. INTRODUCTION	**1**
1.1 Research Background	1
1.2 Research Problems	10
1.3 Research Questions	11
1.4 Research Objectives	12
1.5 Relationship between Research Problems, Questions and Objectives	13
1.6 Research Contributions	15
1.7 Organization of the Thesis	16
2. LITERATURE REVIEW	**21**
2.1 Introduction	21
2.2 Review Method	21
2.2.1 Research Questions	24
2.2.2 Search Strategy	26
2.2.3 Study Selection	28
2.2.4 Data Extraction	31
2.2.5 Study Quality Assessment and Data Synthesis	31
2.2.6 Threats to Validity	32

© Universiti Teknikal Malaysia Melaka

2.3 Significant Journal Publications in Software Defect Prediction 32
2.4 Most Active and Influential Researchers in Software Defect Prediction 35
2.5 Research Topics in the Software Defect Prediction Field 36
2.6 Datasets Used for Software Defect Prediction 40
2.6.1 Public Dataset vs Private Dataset 40
2.6.2 Static Code Attributes 46
2.6.3 NASA MDP Datasets 47
2.7 Methods Used in Software Defect Prediction 48
2.7.1 Review of Defect Prediction Methods 48
2.7.2 Summary of Methods Used in Software Defect Prediction 73
2.8 Most Used Methods for Software Defect Prediction 76
2.8.1 Distribution of Methods 76
2.8.2 Logistic Regression (LR) 77
2.8.3 Naïve Bayes (NB) 78
2.8.4 k-Nearest Neighbor (k-NN) 78
2.8.5 Neural Network Back Propagation (BP) 81
2.8.6 Support Vector Machine (SVM) 86
2.8.7 Decision Tree (DT) 92
2.8.8 Random Forest 97
2.9 Method Perform Best for Software Defect Prediction 98
2.10 Proposed Method Improvements for Software Defect Prediction 99
2.10.1 Feature Selection 100
2.10.2 Ensembling Machine Learning 102
2.11 Proposed Frameworks for Software Defect Prediction 103
2.11.1 Menzies *et al.*'s Framework 103
2.11.2 Lessmann *et al.*'s Framework 104
2.11.3 Song *et al.*'s Framework 106
2.11.4 Summary of the Framework Comparison Results 107
2.12 Systematic Literature Review References and the Complete Mind Map 109
2.13 Summary 113

3. RESEARCH METHODOLOGY AND THE DEVELOPMENT OF THE PROPOSED SOFTWARE DEFECT PREDICTION FRAMEWORK 115
3.1 Introduction 115
3.2 Research Phases 115
3.3 Research Scope and Design 118

© Universiti Teknikal Malaysia Melaka

3.3.1 Research Scope 118
3.3.2 Research Design 122
3.4 The Development of the Proposed Software Defect Prediction Framework 123
3.5 Data Preparation 128
3.6 Model Validation and Evaluation 133
3.6.1 Model Validation 133
3.6.2 Model Evaluation 133
3.7 Model Comparison 135
3.7.1 Comparisons of Two Classifiers 136
3.7.2 Comparisons of Multiple Classifiers 138
3.8 Experimental Settings 139
3.9 Summary 141

4. A COMPARISON FRAMEWORK OF CLASSIFICATION MODELS FOR SOFTWARE DEFECT PREDICTION 142
4.1 Introduction 142
4.2 Proposed Comparison Framework of Classification Models for Software Defect Prediction (CF-SDP) 144
4.2.1 Dataset 146
4.2.2 Classification Algorithms 146
4.2.3 Model Validation 146
4.2.4 Model Evaluation 147
4.2.5 Model Comparison 148
4.3 Experimental Results and Analysis 148
4.4 Summary 153

5. A HYBRID GENETIC ALGORITHM BASED FEATURE SELECTION AND BAGGING TECHNIQUE FOR SOFTWARE DEFECT PREDICTION 155
5.1 Introduction 155
5.2 Proposed Hybrid Genetic Algorithm based Feature Selection and Bagging Method (GAFS+B) 155
5.3 Experimental Results and Analysis 159
5.4 Summary 164

6. A HYBRID PARTICLE SWARM OPTIMIZATION BASED FEATURE SELECTION AND BAGGING TECHNIQUE FOR SOFTWARE DEFECT PREDICTION 166
6.1 Introduction 166

© Universiti Teknikal Malaysia Melaka

6.2 Proposed Hybrid Particle Swarm Optimization based Feature Selection and Bagging Method (PSOFS+B) 166
6.3 Experimental Results and Analysis 170
6.4 Comparison between GAFS+B Method and PSOFS+B Method 176
6.5 Comparison between the Proposed Methods with the Other Methods 177
6.6 Selected Relevant Features 180
6.7 Summary 183

7. A HYBRID GENETIC ALGORITHM BASED NEURAL NETWORK PARAMETER OPTIMIZATION AND BAGGING TECHNIQUE FOR SOFTWARE DEFECT PREDICTION 185
7.1 Introduction 185
7.2 Proposed Hybrid Genetic Algorithm based Neural Network Parameter Optimization and Bagging Method (NN-GAPO+BB) 187
7.3 Experimental Results and Analysis 190
7.4 Summary 195

8. CONCLUSIONS 196
8.1 Conclusions and Research Contributions 196
8.1.1 Conclusion Related to Research Objective 1 198
8.1.2 Conclusion Related to Research Objective 2 199
8.1.3 Conclusion Related to Research Objective 3 200
8.1.4 Conclusion Related to Research Objective 4 201
8.1.5 Conclusion Related to Research Objective 5 201
8.2 Future Works 202

REFERENCES 205
APPENDICES 219

© Universiti Teknikal Malaysia Melaka

LIST OF TABLES

TABLE	TITLE	PAGE
1.1	Relationship between RP, RQ and RO	14
2.1	Summary of PICOC	24
2.2	Research Questions on Literature Review	25
2.3	Inclusion and Exclusion Criteria	28
2.4	Data Extraction Properties Mapped to Research Questions	31
2.5	Scimago Journal Rank (SJR) of Selected Journals	35
2.6	Summary of the State-of-the-Art Methods in Software Defect Prediction	74
2.7	The List of Primary Studies in the Field of Software Defect Prediction	110
3.1	Frameworks Comparison	125
3.2	Description of the NASA MDP Dataset	129
3.3	Code Attributes within the MDP Datasets	131
3.4	Description of Code Attributes	132
3.5	Stratified 10 Fold Cross Validation	133
3.6	Detail of the Parameter Settings of the Classifiers	140
4.1	AUC value, Its Meaning and Symbols	148
4.2	AUC of Ten Classification Models on Nine Datasets	149
4.3	Pairwise Comparisons of the Nemenyi Post Hoc Test	150
4.4	*P*-value of the Nemenyi Post Hoc Test	150
4.5	Significant Differences of the Nemenyi Post Hoc Test	151
5.1	Detail of the Parameter Settings of Genetic Algorithm and Bagging	159

© Universiti Teknikal Malaysia Melaka

5.2 AUC of Ten Classifiers on Nine Datasets (without GA and Bagging) 160

5.3 AUC of Ten Classifiers on Nine Datasets (with GA and Bagging) 161

5.4 Paired Two-tailed t-Test of without/with GA and Bagging 164

6.1 Detail of the Parameter Settings of Particle Swarm Optimization and Bagging 171

6.2 AUC of Ten Classifiers on Nine Datasets (without PSO and Bagging) 172

6.3 AUC of Ten Classifiers on Nine Datasets (with PSO and Bagging) 173

6.4 Paired Two-tailed t-Test of without/with PSO and Bagging 175

6.5 Paired Two-tailed *t*-Test of GAFS+B and PSOFS+B 176

6.6 Detail of the Parameter Settings of InfoGain and Forward Selection 178

6.7 AUC Comparison of the Five Different Methods on Nine Datasets 178

6.8 Weight Values of Selected Relevant Features on Ten Classifiers 181

7.1 Detail of the Parameter Settings of Genetic Algorithm and Bagging 191

7.2 AUC of NN Model on Nine Datasets 192

7.3 AUC of NN-GAPO+B Model on 9 Datasets 193

7.4 AUC Comparisons of the NN Model and NN-GAPO+B Model 194

7.5 Paired Two-tailed t-Test of NN Model and NN-GAPO+B Model 195

© Universiti Teknikal Malaysia Melaka

LIST OF FIGURES

FIGURE	TITLE	PAGE
1.1	Relationship between RP, RQ, RC and Research Publications	16
1.2	Organization of the Thesis	17
2.1	Systematic Literature Review Steps	23
2.2	Basic Mind Map of the SLR on Software Defect Prediction	26
2.3	Search and Selection of Primary Studies	30
2.4	Distribution of Selected Studies over the Years	33
2.5	Journal Publications and Distribution of Selected Studies	34
2.6	Influential Researchers and Number of Studies	36
2.7	Distribution of Research Topics	40
2.8	Total Distribution of Datasets	42
2.9	Distribution of Private and Public Datasets	43
2.10	Datasets Used in Software Defect Prediction Before 2005	44
2.11	Datasets Used in Software Defect Prediction After 2005	45
2.12	Highly Used Classification Methods in Software Defect Prediction	76
2.13	Distribution of the Studies over Type of Methods	77
2.14	Neural Network Architecture (Sammut and Webb 2011)	82
2.15	Example of the Maximum Margin Separator	87
2.16	Graph of Hinge Loss	89
2.17	A Decision Tree Describing the Golf Dataset (Sammut and Webb 2011)	93
2.18	Top-Down Induction of the Decision Trees Algorithm	94

© Universiti Teknikal Malaysia Melaka

2.19 A Complex Decision Tree (Sammut and Webb 2011) 95

2.20 Menzies *et al.*'s Framework (Compiled from (Menzies *et al.* 2007)) 104

2.21 Lessmann *et al.*'s Framework (Compiled from (Lessmann *et al.* 2008)) 105

2.22 Song *et al.*'s Framework (Compiled from (Song *et al.* 2011)) 107

2.23 Complete Mind Map of the SLR on Software Defect Prediction 112

3.1 Research Methodology: Phases, Tasks and Related Chapters 116

3.2 Research Scope and Design 119

3.3 Proposed Software Defect Prediction Framework 124

3.4 Two ROC Graphs 135

4.1 Proposed Comparison Framework of Classification Models for Software Defect Prediction (CF-SDP) 145

4.2 AUC Mean (*M*) Comparison of Ten Classification Models on Nine Datasets 153

5.1 Flowchart of the Hybrid of Genetic Algorithm based Feature Selection and Bagging Method (GAFS+B) 157

5.2 AUC Comparisons of Nine Datasets Classified by Ten Classifiers 162

6.1 Flowchart of the Hybrid Particle Swarm Optimization based Feature Selection and Bagging Method (PSOFS+B) 169

6.2 AUC Comparisons of Nine Datasets Classified by Ten Classifiers (Without/With PSO and Bagging) 174

6.3 AUC Comparison of the Five Different Methods on Nine Datasets 179

6.4 Weight Values of Selected Relevant Features on Ten Classifiers 182

6.5 Weight Values of Selected Relevant Features in Ten Classifiers in Value Order 183

7.1 Flowchart of the Hybrid Genetic Algorithm based Neural Network Parameter Optimization and Bagging Method (NN-GAPO+BB) 189

7.2 AUC of NN Model on Nine Datasets 192

© Universiti Teknikal Malaysia Melaka

7.3 AUC of NN-GAPO+B Model on 9 Datasets 193

7.4 AUC Comparisons of the NN Model and NN-GAPO+B Model 194

© Universiti Teknikal Malaysia Melaka

LIST OF APPENDICES

APPENDIX	TITLE	PAGE
A	NASA MDP Datasets	219
A1	MW1 Dataset	219
A2	KC3 Dataset	225
B	Generated Software Defect Prediction Models	231
B1	C4.5 Model of MW1 Dataset	231
B2	Support Vector Machine Model of MW1 Dataset	232
B3	Neural Network Model of MW1 Dataset	234

© Universiti Teknikal Malaysia Melaka

LIST OF ABBREVIATIONS AND GLOSSARY

1R	*Inferring Rudimentary Rules*. A simple, cheap method that often comes up with quite good rules for characterizing the structure in data. It generates a one-level decision tree expressed in the form of a set of rules that all test one particular attribute (Witten et al. 2011).
ACM	*Association for Computing Machinery*. The world's largest and most prestigious scientific and educational computing society. It was founded in 1947 and its headquarters are in New York City.
ACO	*Ant Colony Optimization*. Ant colony optimization is derived from the foraging behavior of real ants in nature, based strongly on the ant system metaheuristic developed by Dorigo, Maniezzo and Colorni,. The main idea of ACO is to model the problem as the search for a minimum cost path in a graph. Artificial ants walk through this graph, looking for good paths. Each ant has a rather simple behavior so that it will typically only find rather poor-quality paths on its own. Better paths are found as the emergent result of the global cooperation among ants in the colony (Floudaos & Pardalos 2009).
AIS	*Artificial Immune System*. Artificial immune system is an intelligent problem-solving technique that has been used in scheduling problems for about ten years. AISs are computational systems inspired by theoretical immunology, observed immune functions, principles and mechanisms in order to solve problems. Nature and in particular biological systems have always been fascinating to the human expert owing to their complexity, flexibility and sophistication. The nervous system inspired the evolution of an artificial neural network, in the very similar manner immune system motivated the emergence of the AIS. The AIS can be defined as an abstract or metamorphic computational

© Universiti Teknikal Malaysia Melaka

	system using ideas gleaned from the theories and component of immunology (Floudaos & Pardalos 2009).
AUC	*Area under Curve*. An empirical measure of classification performance based on the area under an ROC curve. It evaluates the performance of a scoring classifier on a test set, but ignores the magnitude of the scores and only takes their rank order into account. AUC is expressed on a scale of 0 to 1, where 0 means that all negatives are ranked before all positives, and 1 means that all positives are ranked before all negatives (Sammut & Webb 2011).
BE	*Backward Elimination*. Backward elimination is one of several computer-based iterative variable-selection procedures. It begins with a model containing all the independent variables of interest. Then, at each step the variable with smallest F-statistic is deleted (if the F is not higher than the chosen cutoff level).
BP	*Back Propagation*. An abbreviation for "backward propagation of errors", is a common method of training neural networks. From a desired output, the network learns from many inputs, similar to the way a child learns to identify a dog from examples of dogs. It is a supervised learning method, and is a generalization of the delta rule. It requires a dataset of the desired output for many inputs, making up the training set. It is most useful for feed-forward networks. Back propagation requires that the activation function used by the artificial neurons be differentiable.
CART	*Classification and Regression Tree*. A machine learning method for constructing prediction models from data. The models are obtained by recursively partitioning the data space and fitting a simple prediction model within each partition. As a result, the partitioning can be represented graphically as a decision tree. Classification trees are designed for dependent variables that take a finite number of unordered values, with prediction error measured in terms of misclassification cost. Regression trees are for dependent variables that take continuous or ordered discrete values, with prediction error typically measured by the

© Universiti Teknikal Malaysia Melaka

	squared difference between the observed and predicted values (Loh 2011).
CBO	*Coupling between Object Classes*. The coupling between object classes is a count of the number of other classes to which it is coupled. It has been introduced by Chidamber and Kemerer. CBO relates to the notion that an object is coupled to another object if one of them acts on the other, i.e. methods of one use methods or instance variables of another. As stated earlier, since objects of the same class have the same properties, two classes are coupled when methods declared in one class use methods or instance variables defined by the other class.
CBR	*Case Based Reasoning*. It solves problems by retrieving similar, previously solved problems and reusing their solutions. Experiences are memorized as cases in a case base. Each experience is learned as a problem or situation together with its corresponding solution or action (Sammut & Webb 2011)
CF-SDP	A comparison framework of classification models for software defect prediction
CPU	*Central Processing Unit*. A group of circuits that performs the basic functions of a computer. The CPU is made up of three parts, the control unit, the arithmetic and logic unit and the input/output unit (Collin 2004)
DIT	*Depth of Inheritance Tree*. Depth of Inheritance Tree is the maximum length of a path from a class to a root class in the inheritance structure of a system. DIT measures how many super-classes can affect a class. DIT is only applicable to object-oriented systems.
DT	*Decision Tree*. A tree-structured classification model, which is easy to understand, even by non-expert users, and can be efficiently induced from data. The induction of decision trees is one of the oldest and most popular techniques for learning discriminatory models, which has been developed independently in the statistical and machine learning communities (Sammut & Webb 2011).
EM	*Expectation-Maximization*. An iterative method for finding maximum likelihood or maximum a posteriori (MAP) estimates of parameters in

© Universiti Teknikal Malaysia Melaka

	statistical models, where the model depends on unobserved latent variables. The EM iteration alternates between performing an expectation (E) step, which creates a function for the expectation of the log-likelihood evaluated using the current estimate for the parameters, and a maximization (M) step, which computes parameters maximizing the expected log-likelihood found on the E step. These parameter-estimates are then used to determine the distribution of the latent variables in the next E step (Sammut & Webb 2011).
FNR	*False Negative Rate*. Percentage of incorrect results that are, in fact, positive. See FPR.
FPR	*False Positive Rate*. The false positive rate is FP / (FP + TN). Where FP is number of false positives, and TN is number of true negatives. In statistics, when performing multiple comparisons, the term false positive ratio, also known as the false alarm ratio, usually refers to the probability of falsely rejecting the null hypothesis for a particular test. The false positive rate (false alarm rate) usually refers to the expectancy of the false positive ratio.
FS	*Forward Selection*. Forward selection is one of several computer-based iterative variable-selection procedures. It resembles step-wise regression except that a variable added to the model is not permitted to be removed in the subsequent steps.
GA	*Genetic Algorithm*. Genetic algorithms are search procedures based on the mechanics of natural selection and natural genetics. The first GA was developed by John H. Holland in the 1960s to allow computers to evolve solutions to difficult search and combinatorial problems, such as function optimization and machine learning. GAs are based on an imitation of the biological process in which new and better populations among different species are developed during evolution. Thus, GAs use information about a population of solutions, called individuals, when they search for better solutions. A GA is a stochastic iterative procedure that maintains the population size constant in each iteration, called a generation. The basic operation is the mating of two solutions in order

© Universiti Teknikal Malaysia Melaka

	to form a new solution. To form a new population, a binary operator, called crossover, and a unary operator, called mutation, are applied. Crossover takes two individuals, called parents, and produces two new individuals, called offspring, by swapping parts of the parents (Floudaos & Pardalos 2009).
GAFS+B	A hybrid genetic algorithm based feature selection and bagging technique for software defect prediction (Wahono & Herman 2014).
GRNN	*General Regression Neural Network*. Generalized Regression Neural Networks is a special case of Radial Basis Networks. Compared with its competitor, e.g. standard feedforward neural network, GRNN has several advantages. First of all, the structure of a GRNN is relatively simple and static with 2 layers, namely pattern and summation layers. Once the input goes through each unit in the pattern layer, the relationship between the input and the response would be memorized and stored in the unit.
IBL	*Instance Based Learning*. Instance-based learning refers to a family of techniques for classification and regression, which produce a class label and predication based on the similarity of the query to its nearest neighbor in the training set. In explicit contrast to other methods such as decision trees and neural networks, instance-based learning algorithms do not create an abstraction from specific instances. Rather, they simply store all the data, and at query time derive an answer from an examination of the query's nearest neighbor (Sammut & Webb 2011).
IDE	*Integrated Development Environment*. A software application that provides comprehensive facilities to computer programmers for software development. An IDE normally consists of a source code editor, build automation tools and a debugger. Most modern IDEs offer Intelligent code completion features. Some IDEs contain a compiler, interpreter, or both, such as Net Beans and Eclipse; others do not, such as SharpDevelop and Lazarus.
IEEE	*Institute of Electrical and Electronics Engineers*. A professional association with its Corporate Office in New York City and its

© Universiti Teknikal Malaysia Melaka

	Operations Center in Piscataway, New Jersey and is dedicated to advancing technological innovation and excellence. It has about 425,000 members in about 160 countries, slightly less than half of whom reside in the United States.
ISI	*Institute for Scientific Information*. It was founded by Eugene Garfield in 1960 and then acquired by Thomson Scientific & Healthcare in 1992, became known as Thomson ISI. ISI offered bibliographic database services which its specialty on citation indexing and analysis. It maintains citation databases covering thousands of academic journals. This database allows a researcher to identify which articles have been cited most frequently, and who has cited them. The database not only provides an objective measure of the academic impact of the papers indexed in it, but also increases their impact by making them more visible and providing them with a quality label. The ISI also publishes the annual Journal Citation Reports which list an impact factor for each of the journals that it tracks.
IV&V	*Independent Verification and Validation*. Independent verification and validation involves verification and validation done by a third party organization not involved in the development of the product. Thus, the product, such as a software, gets examined by third party. The main check performed is whether user requirements are met alongside ensuring that the product is structurally sound and built to the required specifications.
k-NN	*k-Nearest Neighbor*. k-Nearest Neighbor algorithm represents that classification method, in which a new object is labeled based on its closest (k) neighboring objects. In principle, given a training dataset (left) and a new object to be classified (right), the distance (referring to some kind of similarity) between the new object and the training objects is first computed, and the nearest (most similar) k objects are then chosen (Gorunescu 2011).
LDA	*Linear Discriminant Analysis*. A methods used in statistics, pattern recognition and machine learning to find a linear combination of

© Universiti Teknikal Malaysia Melaka

	features which characterizes or separates two or more classes of objects or events. The resulting combination may be used as a linear classifier, or for dimensionality reduction before later classification. LDA is closely related to ANOVA (analysis of variance) and regression analysis, which also attempt to express one dependent variable as a linear combination of other features or measurements. However, ANOVA uses categorical independent variables and a continuous dependent variable, whereas discriminant analysis has continuous independent variables and a categorical dependent variable.
LOC	*Line of Code*. Also known as source lines of code (SLOC), is a software metric used to measure the size of a computer program by counting the number of lines in the text of the program's source code. LOC is typically used to predict the amount of effort that will be required to develop a program, as well as to estimate programming productivity or maintainability once the software is produced.
LR	*Logistic Regression*. Logistic regression provides a mechanism for applying the techniques of linear regression to classification problems (Sammut & Webb 2011). It measures the relationship between a categorical dependent variable and one or more independent variables, which are usually continuous, by using probability scores as the predicted values of the dependent variable.
M	*Mean*. In probability and statistics, mean and expected value are used synonymously to refer to one measure of the central tendency either of a probability distribution or of the random variable characterized by that distribution. For a data set, the terms arithmetic mean, mathematical expectation, and sometimes average are used synonymously to refer to a central value of a discrete set of numbers: specifically, the sum of the values divided by the number of values.
MBR	*Memory Based Reasoning*. Memory based reasoning technique is one such quantitative method that predicts a new case by retrieving similar cases from the past. It uses the past cases to predict the solution to the

© Universiti Teknikal Malaysia Melaka