A FEATURE RANKING ALGORITHM IN PRAGMATIC QUALITY FACTOR MODEL FOR SOFTWARE QUALITY ASSESSMENT

RUZITA AHMAD

MASTER OF SCIENCE (INFORMATION TECHNOLOGY)
UNIVERSITI UTARA MALAYSIA
2013

Permission to Use

In presenting this thesis in fulfilment of the requirements for a postgraduate degree from Universiti Utara Malaysia, I agree that the Universiti Library may make it freely available for inspection. I further agree that permission for the copying of this thesis in any manner, in whole or in part, for scholarly purpose may be granted by my supervisor(s) or, in their absence, by the Dean of Awang Had Salleh Graduate School of Arts and Sciences. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to Universiti Utara Malaysia for any scholarly use which may be made of any material from my thesis.

Requests for permission to copy or to make other use of materials in this thesis, in whole or in part, should be addressed to:

Dean of Awang Had Salleh Graduate School of Arts and Sciences
UUMCollege of Arts and Sciences
Universiti Utara Malaysia
06010 UUM Sintok

Abstrak

Kualiti perisian adalah satu bidang penyelidikan yang penting dan telah mendapat perhatian dikalangan komuniti kejuruteraan perisian terutama dalam mengenal pasti atribut penting dalam proses pembangunan perisian. Tesis ini menerangkan penyelidikan asli dalam bidang model kualiti perisian dengan memperkenalkan algoritma *Feature Ranking Algorithm* (FRA) untuk model *Pragmatic Quality Factor* (PQF). Algoritma yang dicadangkan mampu memperbaiki kelemahan model sedia ada dalam mengemaskini dan mempelajari kombinasi atribut untuk penaksiran kualiti perisian. Teknik penaksiran sedia ada kurang keupayaan untuk menyenaraikan atribut mengikut keutamaan dan keupayaan pembelajaran data yang boleh meningkatkan proses penaksiran kualiti. Tujuan kajian ini adalah untuk mengenal pasti dan mencadangkan penggunaan teknik dalam bidang Kepintaran Buatan ke arah meningkatkan proses penaksiran kualiti dalam model PQF. Oleh itu, algoritma FRA yang menggunakan *Feature Ranking Technique* (FRT) telah dibina dan prestasi algoritma FRA telah dinilai. Metodologi yang digunakan terdiri daripada kajian teori, reka bentuk rangka kerja formal untuk kualiti perisian pintar, mengenal pasti kesesuaian ciri-ciri FRT untuk penyenaraian atribut, pembangunan dan penilaian algoritma FRA. Penaksiran atribut telah bertambah baik dengan menggunakan algoritma FRA yang mengandungi formula untuk mengira keutamaan atribut dan diikuti oleh adaptasi pembelajaran melalui aplikasi *Java Library for Multi Label Learning* (MULAN). Hasil kajian menunjukkan bahawa prestasi algoritma FRA mempunyai kolerasi yang sangat kuat dengan model pakar iaitu model PQF. Ujian statistik menunjukkan bahawa FRA telah menghasilkan keputusan ketepatan yang lebih baik berbanding algoritma Kolmogorov-Smirnov Correlation Based Filter (KSCBF) iaitu 98% berbanding 83% masing-masing. Ujian statistik juga menghasilkan keputusan bagi algorithm FRA iaitu 0.052 adalah lebih baik berbanding dengan algoritma KSCBF iaitu 0.048. Ini menunjukkan bahawa keputusan FRA adalah lebih signifikan berbanding algoritma yang digunakan. Sumbangan utama kajian ini adalah dalam pelaksanaan teknik FRT yang memperkenalkan pengiraan *Most Priority of Features* (MPF) dalam algoritma FRA untuk teknik penaksiran tersebut. Kesimpulannya, penemuan kajian ini menyumbang kepada usaha penyelidikan baru dalam bidang pemilihan atribut dalam kualiti perisian.

Kata Kunci: Perisian kualiti, Algoritma FRA, Teknik Kepintaran Buatan, dan Mesin Pembelajaran

Abstract

Software quality is an important research area and has gain considerable attention from software engineering community in identification of priority quality attributes in software development process. This thesis describes original research in the field of software quality model by presenting a Feature Ranking Algorithm (FRA) for Pragmatic Quality Factor (PQF) model. The proposed algorithm is able to improve the weaknesses in PQF model in updating and learning the important attributes for software quality assessment. The existing assessment techniques lack of the capability to rank the quality attributes and data learning which can enhance the quality assessment process. The aim of the study is to identify and propose the application of Artificial Intelligence (AI) technique for improving quality assessment technique in PQF model. Therefore, FRA using FRT was constructed and the performance of the FRA was evaluated. The methodology used consists of theoretical study, design of formal framework on intelligent software quality, identification of Feature Ranking Technique (FRT), construction and evaluation of FRA algorithm. The assessment of quality attributes has been improved using FRA algorithm enriched with a formula to calculate the priority of attributes and followed by learning adaptation through Java Library for Multi Label Learning (MULAN) application. The result shows that the performance of FRA correlates strongly to PQF model with 98% correlation compared to the Kolmogorov-Smirnov Correlation Based Filter (KSCBF) algorithm with 83% correlation. Statistical significance test was also performed with score of 0.052 compared to the KSCBF algorithm with score of 0.048. The result shows that the FRA was more significant than KSCBF algorithm. The main contribution of this research is on the implementation of FRT with proposed Most Priority of Features (MPF) calculation in FRA for attributes assessment. Overall, the findings and contributions can be regarded as a novel effort in software quality for attributes selection.

Keywords: Software Quality, FRA Algorithm, Artificial Intelligence (AI) Technique, and Machine Learning

Acknowledgement

In the name of Allah, the Beneficent, the Merciful. Alhamdulillah, grateful to Almighty Allah (SWT) for all the blessing He has bestowed upon me, lastly this thesis is finally completed.

A lot of thanks to Universiti Utara Malaysia for the supports and the facilities provided. My deepest gratitude and appreciation goes to my respected supervisors, Associate Professor Dr. Jamaiah Hj Yahaya and Dr. Siti Sakira Kamaruddin for never ending support, courage and helps along to complete this thesis.

Last but not least, I wish to express my deepest appreciation goes to my supportive husband, my loving mother, and my dear children, for believing in me, their continuous prayers and unconditional support. Lastly I would like to dedicate this work to the memory of my father, Allahyarham Ahmad bin Sulaiman (1938-2010) whom I unexpected lost during this study. Thank you all.

Table of Contents

List of Tables

List of Figures

List of Appendices

List of Abbreviations

AHS	Automatic Hybrid Search
AI	Artificial Intelligence
ANN	Artificial Neural Network
API	Application Programming Interface
ARFF	Attribute Relation File Format
AUC	Area Under the Curve
BNS	Bi-Normal Separation
CBFS	Correlation Based Feature Selection
CBFSS	Consistency Based Feature Subset Selection
CBR	Case-Based Reasoning
CS	Chi-Square
DF	Document Frequency
DFM	Default F-Measure
DGM	Default Geometric Mean
ESD	Airforce Electronic System Division
FAS	Filter Attribute Selection
FCBF	Fast Correlation Based Filter
FRA	Feature Ranking Algorithm
FRT	Feature Ranking Technique
FS	Feature Selection
FSST	Feature Subset Selection Technique
GA	Genetic Algorithm
GR	Gain Ratio
GRNN	Generalized Regression Neural Network
GUI	Graphical User Interface
HFS	Hybrid Feature Selection
IBL	Instance Based Learning
IEEE	International Symposium on Requirement Engineering
IG	Information Gain

ISO	International Organization Standard
JRE	Java Runtime Environment
KNN	K-Nearest Neighbour
KS	Kolmogorov Smirnov
KSCBF	Kolmogorov-Smirnov Correlation Based Filter
K-S TEST	Kolmogorov Smirnov Two Sample Test
LEET	Large Experiment and Evaluation Tool
LOC	Lines of Code
LR	Logistic Regression
MATLAB	Matrix Laboratory
MI	Mutual Information
MLKNN	Multi Label K-Nearest Neighbour
MLOSS	Machine Learning Open Source Software
MLP	Multi Layer Perceptron
MPF	Most Priority of Attribute
MULAN	Java Library for Multi Label Learning
NB	Naïve Bayes
NN	Neural Network
OA	Overal Accuracy
PQF	Pragmatic Quality Factor
PS	Probabilistic Search
QFD	Quality Function Deployment
RADC	Rome Air Development Centre
RAKEL	Random *k*-Labelstes
RS	Rough Sets
SPSS	Statistical Package for the Social Sciences
SQA	Software Quality Assurance
SQuaRE	Software Product Quality Requirement and Evaluation
STS	Spring Source Tool Suite
SU	Symmetrical Uncertainty
SVM	Support Vector Machine

WEKA	Waikato Environment Knowledge Analysis
WLLR	Weighted Log Likelihood Ratio
WWW	World Wide Web
XML	Extensible Markup Language

CHAPTER ONE
INTRODUCTION

1.1 Overview

Chapter One presents the overall study and briefly explains the aims of the research. Several sections have been defined to classify and identify the purpose of this study. These include research background, problem statement of the research, research motivation, research objectives, scope of study and methodology.

1.2 Research Background

Nowadays, rapid development and diffusion of software quality is related to technologies in several industries. Statistics shows on insufficiently understood requirements accounted to 50% of errors. This was followed by design incorrectly understood from requirements, which accounted to 30% of errors. Hence, programming errors of system design contributed to 20% of errors (Humphrey et al., 1989). In fact, the organization has outlined the exactly errors in perfectly before they starts to develop a software product. Thus, Software Quality Assurance (SQA) is a very important domain in software development and its purpose is to find ways to reduce the rate and associated cost of failure from poor product and services (Humphrey et al., 1989).

In order to reduce errors in systems design and to fulfill user needs and requirements, the quality of systems development should be highlighted as an important goal. Normally, the standard level of quality is recommended by the International Organization for Standardization (ISO) and IEEE as well. ISO defines quality as the

The contents of the thesis is for internal user only

REFERENCES

Aamodt, A. (1994). Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches, *7*, 39–59.

Abran, A., Khelifi, A. & Suryn, W. (2003). Usability meanings and interpretations in ISO standards. *Software Quality Journal, 11,* 323-336.

Aguero, M., Madou, F., Esperon, G. & Lopez, D. L. (2010). Artificial intelligence for software quality improvement. *World Academy of Science and Technology, 63*.

Anna, S. C., Garcia, A. P., Chavez, C. V. & Lucena, C. J. (2003). On the reuse and maintenance of aspect-oriented software: An assessment framework. *Computer Science Department,* Retrieved September 27, 2010 from IEEE Computer Society.

Allen, E. B. (2001). Controlling Overfitting in Classification Tree Models of Software Quality. *2001 International Symposium on Empirical Software Engineering, 2001.*, 59–79.

Ashrafi, N. (2003). The impact of software process improvement on quality in theory and practice. *Information & Management*, *40*(7), 677–690.

Azuma, M. (1991). SQuaRE the next generation of the ISO/IEC 9126 and 14598 international standards series on software product quality. *Technical Report, ISO/IEC* JTC1/SC7/WG6.

Bansiya, J., & Davis, C. G. (2002). A hierarchical model for object-oriented design quality assessment. *IEEE Transactions on Software Engineering*, *28*(1), 4–17.

Baumgartner, D., & Serpen, G. (2009). Large Experiment and Evaluation Tool for WEKA Classifiers. *5th International Conference on Data Mining*, 340–346.

Bevan, N. (1984). Quality in use: Incorporating Human Factors into the Software Engineering Lifecycle. *Software Engineering Standards Symposium and Forum, 1997.'Emerging International Standards'. ISESS 97., Third IEEE International*, 169–179.

Bevan, N. (1997). Quality and usability: A new framework. *National Physical Laboratory*, United Kingdom.

Bevan, N. (1999). Quality in use: Meeting user needs for quality. *The Journal of Systems and Software, 49, March 19.* 89-96.

Bevan, N. (1999). Quality in use: Incorporating human factors into the software engineering lifecycle. *National Physical Laboratory*, Retrieved September 27, 2010 from IEEE Computer Society.

Bhatti, S. N. (2005). Why Quality? ISO 9126 Software Quality Metrics (Functionality) Support by UML Suite. *Advances in Engineering Software*, *30*(2), 1–5.

Biesiada, J., & Duch, W. (2005). Feature Selection for High-Dimensional Data: A Kolmogorov-Smirnov Correlation-Based Filter. *Advances in Soft Computing*, *30*, 95–103.

Biesiada, J., & Duch, W. (2007). Feature Selection for High-Dimensional Data: A Pearson Redundancy Based Filter. *Advances in Soft Computing*, *45*, 242–249.

Blachnik, M., Duch, W., Kachel, A., & Biesiada, J. (2009). Feature Selection for Supervised Classification: A Kolmogorov-Smirnov Class Correlation-Based Filter. In *AIMeth, Symposium On Methods Of Artificial Intelligence. Gliwice, Poland (10-19 November 2009)*.

Blum, A. L., & Langley, P. (1997). Artificial Intelligence Selection of relevant features and examples in machine. *Artificial Intelligence*, *97*, 245–271.

Briand, L. et al., (2000). Exploring the relationships between design measures and software quality in object-oriented systems. *Journal of Systems and Software 51*, 245–273.

Buglione, L. & Abran, A. (1999). A quality factor for software. Proceedings from: *QUALITA99, 3rd International Conference on Quality and Reliability,* 335-344.

Burgess, C. J. (2000). Using Artificial Intelligence to solve problems in software quality management. Proceedings from: *The 8th International Conference on Software Quality Management (SQM2000), Software Quality Management VIII*. ISBN 1-902505-25-5, 77–89.

Cheikhi, L., Abran, A. & Suryn, W. (2006). Harmonization of usability measurements in ISO9126 software engineering standards. Proceedings from: *IEEE International Conference on Software Engineering, July 9-12, 2006, Montreal Quebec, Canada,* ISBN: 1-4244-0497-5, 3246-3251.

Dash, M., & Liu, H. (1997). Feature selection for classification. *Intelligent Data Analysis*, *1*(1-4), 131–156.

Dash, M., & Liu, H. (2003). Consistency-based search in feature selection. *Artificial Intelligence*, *151*(1-2), 155–176.

Denning, P. J. (1992). What is Software Quality? *A Commentary from Communications of ACM* (January).

Deraman, A. & Yahaya, J. H. (2010). Measuring the unmeasurable characteristics of software quality using pragmatic quality factor. Proceedings from: *2010 3rd IEEE International Conference on Computer Science and Information Technology, July 7-10, 2010, Chengdu, China,* ISBN:978-1-4244-5539-3, 197-202.

Dromey, R. G., & Popper, K. (1994). A model for software product quality. *Software Quality Institute*, (October), 1–35.

Dromey, G. R. (1995). A model for software product quality. *IEEE Transaction on Software Engineeering, February, 21(2),* 146-162.

Dromey, G. R. (1998). Software product quality: Theory, model and practice. *Software Quality Institute. Griffith University, Brisbane, Technical Report*. Retrieved 23 August, 2010, from http://www.sqi.gu.edu.au.

Dromey, G. R. (1999). Cornering the chimera, *IEEE Software, January*, 33-43.

Duch, W., Winiarski, T., Biesiada, J., & Kachel, A. (2003, June). Feature selection and ranking filters. *International Conference on Artificial Neural Networks (ICANN) and International Conference on Neural Information Processing (ICONIP)*. 251-254.

Durrett, R. (2010). *Probability: theory and examples*. Cambridge University Press.

Engineers, E. (1993). *IEEE Standard for a Software Quality Metrics Methodology* (pp. 1–73).

Forman, G. (2003). An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Research, 3,*1289–1305.

Fitzpatrick, R. (1996). Software quality: Definitions and strategies issues. Retrieved Sept 13, 2010, from http://ieeexplore.ieee.org/xpl/standards.jsp.

Fitzpatrick, R. & Higgins, C. (1998). Usable software and its attributes synthesis of software quality: European community law and human-computer. Retrieved Sept 13, 2010, from http://ieeexplore.ieee.org/xpl/standards.jsp.

Forman, G. (2003). An Extensive Empirical Study of Feature Selection Metrics for Text Classification. *Machine Learning Research*, *3*, 1289–1305.

Frank, E., Hall, M., Trigg, L., Holmes, G., & Witten, I. H. (2004). Data mining in bioinformatics using Weka. *Bioinformatics (Oxford, England)*, *20*(15), 2479–81.

Friedman, N., Geiger, D., & Goldszmidt, M. (1997). Bayesian network classifiers. *Machine learning*, *29*(2), 131-163.

Gao, K., Khoshgoftaar, T. M., & Napolitano, A. (2009). Exploring Software Quality Classification with a Wrapper-Based Feature Ranking Technique. *2009 21st IEEE International Conference on Tools with Artificial Intelligence*, 67–74.

Gao, K., Raton, B., & Wang, H. (2009). An Empirical Investigation of Filter Attribute Selection Techniques for Software Quality Classification. *Information Reuse & Integration, 2009. IRI'09. IEEE International Conference*, 272–277.

Gao, K. (2010). An Evaluation of Sampling on Filter-Based Feature Selection Methods. *Proceedings of the Twenty-Third International Florida Artificial Intelligence Research Society Conference (FLAIRS 2010)*, 416–421.

Garcia, A. F. (2003). On the Reuse and Maintenance of Aspect-Oriented Software: An Assessment Framework. *Proceedings of Brazilian Symposium on Software Engineering*, 19-34.

Goulao, M. & Abreu, F. B. (2007). Towards a components quality model. *Information Systems Group (INESC)*, Retrieved Oct 26, 2010, from http://www.msc.com.my/xtras/fact_figures/msc.asp.

Grunwald, P., & Vitanyi, P. (2008). Shannon Information and Kolmogorov Complexity, 1–54. Retrieved Nov 29, 2012, from http://arxiv.org/abs/cs/0410002

Guyon, I. (2003). An Introduction to Variable and Feature Selection 1 Introduction. *Journal of Machine Learning Research*, *3*, 1157–1182.

Hall, M. A., & Smith, L. A. (1997). Feature Subset Selection: A Correlation Based Filter Approach. Retrieved Aug 8, 2011, from http://scholar.google.com.my/scholar?q=feature+subset+selection%3A+A+correlation+based+filter+approach&hl=en&as_sdt=0%2C5

Hall, M. A., & Smith, L. A. (1998). Practical Feature Subset Selection for Machine Learning. Retrieved Aug 8, 2011, from http://researchcommons.waikato.ac.nz/bitstream/handle/10289/1512/Practical%20feature%20subset?sequence=1

Hamann, D., Jarvinen, J., & Birk, A. (1998). A Product-Process Dependency Definition Method. *IEEE Transactions on Software Engineering*, *15504*(23239), 898–904.

Hall, M. (1999). Correlation-based feature selection for machine learning. Retrieved Aug 8, 2011, from http://www.lri.fr/~pierres/donn%E9es/save/these/articles/lpr-queue/hall99correlationbased.pdf

Hall, M. A. (2000). Benchmarking Attribute Selection Techniques for Data Mining. Retrieved Aug 8, 2011, from http://researchcommons.waikato.ac.nz/bitstream/handle/10289/1026/uow-cs-wp-2000-10.pdf?sequence=1

Hall, M. A., & Holmes, G. (2003). Benchmarking attribute selection techniques for discrete class data mining. *IEEE Transactions on Knowledge and Data Engineering*, *15*(6), 1437–1447.

Hall, M., National, H., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA Data Mining Software: An Update. *ACM SIGKDD Explorations Newsletter*, *11*(1), 10-18.

Hall, M. A., & Witten, I. H. (2010). WEKA Experiences with a Java Open-Source Project. *Journal of Machine Learning Research*, *11*, 2533–2541.

Humphrey, W. S., Kitson, D., Olson, T. G., Humphrey, W. S., & Kitson, D. (1989). *Conducting SEI-Assisted Software Process Assessments Conducting SEI-Assisted*. *Software Engineering Institute*.

Ioannou, M., Sakkas, G., Tsoumakas, G., & Vlahavas, I. (2010). Obtaining Bipartitions from Score Vectors for Multi-Label Classification. *2010 22nd IEEE International Conference on Tools with Artificial Intelligence*, 409-416.

IEEE. (1993). IEEE standard for a software quality metrics methodology. Retrieved August 20, 2010, from http://ieeexplore.ieee.org/xpl/standards.jsp.

ISO/IEC 9126. (1996). Software quality characteristics and metrics-Part2: External metrics. *Technical Report, ISO/IEC* JTC1/SC7/WG6.

Jain, A., & Zongker, D. (1997). Feature Selection: Evaluation, Application, and Small Sample Performance. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *19*(2), 153-158.

Jenner, M. G. (1995). *Software Quality Management and ISO 9001*. New York: A Wiley/QED publication.

Jiang, B., Ma, L., & Xie, W. (2008). A Hybrid Feature Selection Algorithm: Combination of Symmetrical Uncertainty and Genetic Algorithms. *The Second Internatiional Symposium on Optimization and System Biology (OSB'08)*, 152–157.

John, G. H., Kohavi, R., & Karl, P. (1994). Irrelevant Features and the Subset Selection Problem. *Proceedings of the Eleventh International Conference* (pp. 121–129).

Jorgensen, M. (1999). Software quality measurement. *Advances in Engineering Software*, *30*(12), 907–912.

Jung, H., & Kim, S. (2004). Product Quality: A Survey. *IEEE Computer Society*, 88–92.

Khoshgoftaar, T. M., Munson, J. C., Bhattacharya, B. B., & Richardson, G. D. (1995). Predictive modeling techniques of software quality from software measures. *IEEE Transactions on Software Engineering*, *18*(11), 979-987.

Khoshgoftaar, T. M., Allen, E. B., Hudepohl, J. P., & Aud, S. J. (1997). Application of neural networks to software quality modeling of a very large telecommunications system. *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council*, *8*(4), 902–9.

Khoshgoftar, T., Chien, P. D., & Allen, E. (1998). GP-based software quality prediction. *Proceedings of the Third Annual Conference Genetic Programming, volume* (pp. 60-65).

Khoshgoftaar, T. M., Allen, E. B., Halstead, R., Trio, G. P., & Flass, R. M. (1999). Using process history to predict software quality. *Computer* (Vol. 31, pp. 66-72).

Khoshgoftaar, T. M., & Allen, E. B. (2000). A practical classification rule for software quality models. *Reliability, IEEE Transactions on*, *49*(2), 209-216.

Khoshgoftaar, T. M., Yuan, X., & Allen, E. B. (2000). Balancing misclassification rates in classification tree models of software quality. *Empirical Software Engineering*, *5*(4), 313-330.

Khoshgoftaar, T. M., Nguyen, L., Gao, K., & Rajeevalochanam, J. (2003). Application of an attribute selection method to CBR-based software quality classification. *Proceedings of 15th IEEE International Conference on Tools with Artificial Intelligence*, 47–52.

Khoshgoftaar, T. M., & Su, X. (2009). A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*, *2009*, Retrieved Aug 8, 2011, from http://dl.acm.org/citation.cfm?id=1722966

Khoshgoftaar, T. M., Wang, H., & Van Hulse, J. (2010). A comparative study of threshold-based feature selection techniques. In *Granular Computing (GrC), 2010 IEEE International Conference.* 499-504.

Khosravi, K. (2004). A Quality Model for Design Patterns. Retrieved Aug 8, 2011, from http://www-etud.iro.umontreal.ca/~ptidej/yann-gael/Work/Publications/Documents/041021+Kashayar+Khosravi+Technical+Report.doc.pdf

Kilidar, H., Cox, K., & Kitchenham, B. (2005). The use and usefulness of the ISO/IEC 9126 quality standard. *2005 International Symposium on Empirical Software Engineering, 2005.*, 122–128.

Kitchenham, B., & Pfleeger, S. L. (1996). Introduction Software quality: The Elusive Target. *IEEE Software*, *13*(1), 12–21.

Kira, K., & Rendell, L. A. (1992). The feature selection problem: Traditional methods and a new algorithm. In *Proceedings of the National Conference on Artificial Intelligence*. 129-129.

Kohavi, R., & John, G. H. (1996). Wrappers for Feature Subset Selection. *Artificial Intelligence*, *97*(1), 273–324.

Kolodner, J. L. (1992). An Introduction to Case-Based Reasoning. *Artificial Intelligence Review*, *6*(1), 3–34.

Kolodner, J. (1993). *Case-Based Reasoning, Morgan Kaufmann.*

Kolodner, J. L., Simpson, R. L, & Cyrans, K. S. (1993). A process model of cased-based reasoning in problem solving. *Georgia Institute of Technology, Atlanta, Technical Report.* Retrieved 27 August 2010, from http://www.sqi.gu.edu.au.

Kononenko, I. (1994). Estimating Attributes: Analysis and Extensions of RELIEF. *Machine Learning: ECML-94*, 171–182.

Kumar, R., Rai, S. & Trahen, J. L. (1998). Neural network techniques for software quality evaluation. *Proceedings of the Annual Reliability and Maintainability Symposium*, 155-161.

Laboratorio, M. O. (2002). A Systemic Quality Model for Evaluating Software Products. Retrieved Sept10, 2011, from http://www.lisi.usb.ve/publicaciones/02 calidad sistemica/calidad_24.pdf

Langley, P., & Flamingo, L. (1994). Selection of Relevant Features in Machine Learning. *AAAI Technical Report FS-94-02*, 127–131.

Langley, P., & Blum, A. L. (1997). Selection of Relevant Features and Examples in Machine Learning. *Artificial Intelligence*, *97*, 245–271.

Lee, Y. W., Strong, D. M., Kahn, B. K., & Wang, R. Y. (2002). AIMQ: A Methodology for Information Quality Assessment. *Information & Management*, *40*(2), 133–146.

Lees, B., Hamza, M., & Irgens, C. (1996). Applying Case-Based Reasoning to Software Quality Management. *Burkhard & Lenz (1996)*, 162-169.

Li, S., Xia, R., Zong, C., & Huang, C. R. (2009). A Framework of Feature Selection Methods for Text Categorization. *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, *2*(August), 692–700.

Liu, H., Li, J., & Wong, L. (2002). A comparative study on feature selection and classification methods using gene expression profiles and proteomic patterns. *Genome informatics. International Conference on Genome Informatics*, *13*, 51–60.

Moreira, A., Araújo, J., & Brito, I. (2002). Crosscutting quality attributes for requirements engineering. *Proceedings of the 14th international conference on Software engineering and knowledge engineering - SEKE '02*, 167-174.

Norman, F. (2002). Body of Knowledge for Software Quality. *IEEE Computer Society*, (February), 77-82.

Olivier, P. (2001). Diagrammatic reasoning: An artificial intelligence perspective. *Artificial Intelligence Review*, *15*(1), 63-78.

Ortega, M., Pérez, M., & Rojas, T. (2000). A model for software Product Quality with a Systemic Focus. *4th World Multiconference on Systemics, Cybernetics and Informatics SCI 2000 and The 6th International Conference on Information Systems, Analysis and Synthesis ISAS 2000* (pp. 395-401).

Ortega, M., & Rojas, T. (2003). Construction of a systemic quality model for evaluating a software product. *Software Quality Jurnal*, *11*(July), 219–242.

Ortega, M., Perez, M. & Rojas, T. (2003). A model for software product quality with a systemic focus. Retrieved October 28, 2010 from IEEE Computer Society.

Pfleeger, S. L. (2001). *Software Engineering: Theory and Practice", 2nd ed*. Upper Saddle River, N.J: Prentice Hall.

Pomerol, J. C. (1997). Artificial intelligence and human decision making. *European Journal of Operational Research*, *99*(1), 3-25.

Punch, W. F., Goodman, E. D., Pei, M., Chia-shun, L., Hovland, P., & Enbody, R. (1993). Further Research on Feature Selection and Classification Using Genetic Algorithms. *Proceedings of the 5th International Conference on Genetic Algorithms*, (Jun), 557–564.

Qutaish, R. E. (2009). Measuring the software product quality during the software development life-cycle: An international organization for standardization standard perspective. *Journal of Computer Science, 5 (5),* 392-397.

Raton, B. (2003). Fault Prediction Modeling for Software Quality Estimation: Comparing Commonly Used Techniques. *Empirical Software Engineering*, *8*, 255–283.

Rawashdeh, A., & Matalkah, B. (2006). A new software quality model for evaluating COTS components. *The Journal of Computer Science 2 (4).* 373-381.

Reformat, M., Pedrycz, W., & Pizzi, N. J. (2003). Software Quality Analysis with the Use of Computational Intelligence. *Information and Software Technology*, *45*(7), 405–417.

Rieser, V., & Lemon, O. (2006). Using Machine Learning to Explore Human Multimodal Clarification Strategies. *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, (July), 659–666.

Robu, R., Stoicu., & Tivadar, V. (2010). Arff Convertor Tool for WEKA Data Mining Software. *2010 International Joint Conference on Computational Cybernetics and Technical Informatics*, 247–251.

Ross, S., Fang, L., & Hipel, K. W. (2002). A case-based reasoning system for conflict resolution: design and implementation. *Engineering Applications of Artificial Intelligence*, *15*(3), 369-383.

Saeys, Y., Inza, I., & Larrañaga, P. (2007). A Review of Feature Selection Techniques in Bioinformatics. *Bioinformatics (Oxford, England)*, *23*(19), 2507-2517.

Sajnani, H., Javanmardi, S., Mcdonald, D. W., & Lopes, C. V. (2010). Multi-Label Classification of Short Text: A Study on Wikipedia Barnstars. *The AAAI-11 Workshop on Analyzing Microtext.* Retrieved Nov 23, 2011, from http://pensivepuffin.com/dwmcphd/papers/Sajani.et.al-MultiLabelBarnstars-AAAIShortTextWorkshop.pdf

Science, F., & Box, P. O. (2006). A New Software Quality Model for Evaluating COTS Components Adnan Rawashdeh and Bassem Matalkah. *Journal of Computer Science*, *2*(4), 373–381.

Software, E., & Raton, B. (2004). Comparative Assessment of Software Quality Classification Techniques: An Empirical Case Study. *Empirical Software Engineering*, *9*, 229–257.

Sonnenburg, S., Braun, M. L., Ong, C. S., Bengio, S., Bottou, L., Holmes, G., & Williamson, R. C. (2007). The need for open source software in machine learning. Retrieved Nov 26, 2012, from http://researchcommons.waikato.ac.nz/handle/10289/3928

Stefani, A., & Xenos, M. (2008). E-Commerce System Quality Assessment using a Model based on ISO 9126 and Belief Networks. *Software Quality Journal*, *16*(March), 107–129.

Suryn, W., Abran, A. & April, A. (2003). ISO/IEC SQuaRE: The second generation of standards for software product quality. Retrieved December 20, 2010, from http://www.lrgl.uqam.ca/publications/pdf/799.pdf.

Swiniarski, R. W., & Skowron, A. (2003). Rough Set Methods in Feature Selection and Recognition. *Pattern Recognition Letters*, *24*, 833–849.

Szabo, R. M., & Guasti, P. J. (1995). Exploring the Behaviour of Neural Network Sofware Quality Models. *Software Engineering Journal*, (May), 89–96.

Tadeuchi, Y., Oshima, R., Nishida, K., Yamauchi, K., & Omori, T. (2007). Quick Online Feature Selection Method for Regression-A Feature Selection Method Inspired by Human Behavior. *Systems, Man and Cybernetics, 2007. ISIC. IEEE International Conference.* pp. 1895-1900.

Tadeuchi, Y., Oshima, R., Nishida, K. & Yamauchi, K. (2007). A feature selection method inspired by human behavior. *University Research Institute, Tamagawa.* 1895-1900.

Tahir, M. A., Kittler, J., Mikolajczyk, K., & Yan, F. (2010). Improving Multilabel Classification Performance by Using Ensemble of Multi-label Classifiers. *Multiple Classifier Systems*, 11–21.

Tervonen, I. (1996). Support for quality-based design and inspection. *IEEE Software (January),* 44-54.

Thwin, M. M. T., & Quah, T. S. (2005). Application of Neural Networks for Software Quality Prediction using Object-Oriented Metrics. *Journal of Systems and Software*, *76*(2), 147–156.

Tomar, A. B. (2011). A Systematic Study of Software. *International Journal of Software Engineering & Application (IJSEA)*, *2*(4), 61–70.

Trohidis, K., & Kalliris, G. (2008). Multi Label Classification of Music into Emotion. *ISMIR 2008: Proceedings of the 9th International Conference of Music Information Retrieval*, 325–330.

Tsoumakas, G., & Vlahavas, I. (2010). Random k-labelsets: An ensemble method for multilabel classification. *Machine Learning: ECML 2007*, 406-417.

Tsoumakas, G., & Vilcek, J. (2011). MULAN: A Java Library for Multi-Label Learning. *Journal of Machine Learning Research*, *12*, 2411–2414.

Vivanco, R. (2007). Improving Predictive Models of Software Quality Using an Evolutionary Computational Approach. *2007 IEEE International Conference on Software Maintenance*, 503–504.

Wang, Q. (2009). Feature Selection and Clustering in Software Quality Prediction. *Evaluation and Assessment in Software Engineering 2007*, 1–12.

Wang, H., Khoshgoftaar, T. M., Gao, K., & Seliya, N. (2009). High-Dimensional Software Engineering Data and Feature Selection. *2009 21st IEEE International Conference on Tools with Artificial Intelligence*, 83–90.

Wang, H., Khoshgoftaar, T. M., & Seliya, N. (2011). How many software metrics should be selected for defect prediction? In *Proceedings of the Twenty-Fourth International Florida Artificial Intelligence Research Society Conference* (pp. 69-74).

Weiss, N.A. (2008). *Introductory Statistics*. Pearson International Edition.

Wenger, E. (2004). Artificial intelligence and tutoring systems. *International Journal of Artificial Intelligence in Education*, *14*, 39-65.

Whittaker, J. A., & Voas, J. M. (2002). 50 years of software: key principles for quality. *IT professional*, *4*(6), 28-35.

Witten, I. H., Frank, E., Trigg, L., Hall, M., Holmes, G., & Cunningham, S. J. (1999). Weka: Practical Machine Learning Tools and Techniques with Java Implementations. Retrieved January 10, 2012, from http://scholar.google.com.my/scholar?q=Weka+%3A+Practical+Machine+Learning+Tools+and+Techniques+with+Java+Implementations&hl=en&as_sdt=0%2C5

Wolf, L., & Shashua, A. (2003). Feature selection for unsupervised and supervised inference: the emergence of sparsity in a weighted-based approach. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference,* 378-384.

Xenos, M., & Christodoulakis, D. (1997). Measuring perceived software quality. *Information and Software Technology*, *39*(6), 417–424.

Yahaya, J. H., Deraman, A. & Hamdan, A. R. (2007). A case study in applying software certification model by product quality approach. *The International*

Conference on electrical Engineering and Informatics, June 17-19, Bandung, Indonesia, 706-709.

Yahaya, J. H, Deraman, A. & Hamdan, A. R. (2008). Software quality from behavioural and human perspectives. *IJCSNS International Journal of Computer Science and Network Security, 8(8), August 30*, 53-63.

Yahaya, J. H, Deraman, A. & Hamdan, A. R. (2008). Software certification implementation: Case study analysis and findings. *The 3rd International Symposium on Information Technology 2008 (ITSIM 2008), August 26-29, Kuala Lumpur,* 1541-1548.

Yahaya, J. H., Deraman, A., Hamdan, A. R. & Baharom, F. (2008). Software product certification: A collaborative perspective approach. *The 9th Asia Pacific Industrial Engineering & Management Systems Conference, December 3-5, Bali, Indonesia,* 760-768.

Yahaya, J. H., Deraman, A. & Hamdan, A. R. (2010). Continuously ensuring quality through software certification: A case study. *The International Conference on Information Society (i-Society 2010), June 28-30, London, UK.*

Yang, J., & Honavar, V. (1997). Feature Subset Selection Using A Genetic Algorithm Feature Subset Selection Using 1 Introduction. *Intelligent Systems and Their Applications*, *13*(2), 44–49.

Yu, L., & Liu, H. (2003). Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution. *Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003)*, *2*(2), 856–864.

Yu, L., & Liu, H. (2004). Efficient Feature Selection via Analysis of Relevance and Redundancy. *Journal of Machine Learning Research*, *5*, 1205–1224.