International Journal of Engineering &Technology, *7 (3.12) (2018) 427-433*

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Investigating Implications of Metric Based Predictive Data Mining Approaches towards Software Fault Predictions

Pooja Kapoor[1], Deepak Arora[2], Ashwani Kumar[3]

[1, 2]Department of Computer Science & Engineering, Amity University, Lucknow
[3]Area of IT& Systems, IIM Lucknow
Corresponding Author E-mail: [1]author@ boulder.nist.gov, [2]author@lamar.colostate.edu
[3]author@nrim.go.jp

Abstract

Context: Since 1990, various researches have been working in the area of software fault prediction but yet it is difficult to assess the impacts and progressive path of this research field. Objective: In this research work, author's major objective is to investigate the context and dimensions of research studies performed by different researchers in the area of software fault prediction. This work also focuses on presenting a well defined systematic view of their findings and suggestions after a critical examination of all major approaches applied in this key research area. Method: This research work includes 112 total manuscripts published between 2009 and 2014. These studies are gathered from a pool of total 587 manuscripts. The selection criteria for these manuscripts are title, keywords and citation of that paper. Result: The results of this investigation shows that most of the research work related to software fault prediction have been performed on available data set from NASA repository. Most of the research work performed is basically confined to analysis or comparative study of various machine learning techniques based on their classification accuracy. Various research work published doesn't exhibit clearer representation of any specific prediction model. Conclusion: Still after years of development, there is a huge gap between the industry requirement and the research being performed by different researchers in the field of Software fault prediction. A better collaboration between industry academia is still required. This research work represents a critical investigative approach towards finding the exact gaps to be filled and explored more authentic future research areas in this field. All result finding have been critically examined and compared with existing literature work for better understanding and deep insight over identifying the major strengths of chosen research field.

Index Terms: Software fault prediction; Software metric; Systematic reviews.

1. Introduction

The work presented here is a gap analysis in the field of Software fault prediction. The objective of gap analysis is to identify a direction where a contribution could be made. The work evaluates various literature studies of software fault prediction, so as to identify the appropriate approach for investigating the research question(s). This work is a secondary study that reviews the primary systematic review studies since 2003-2015, related to Software Fault Prediction. Systematic literature review is a process that should answer the questions like how to proceed for the review, steps involved in review and reporting the outcome the review. Systematic reviews are conducted with an aim to cumulate the work in an analytical way. Such reviews do not consider the impact of the research done in a specific area but the objective here is to find out further direction for research in the area. The basic motivation for performing this investigation is to identify the extent to which researchers has explored the field of software fault detection.

In this study, author investigated 11 Systematic reviews on Software Fault Prediction papers published between 2003 and 2015. According to the available literature, Software fault prediction is not a new field to explore, as authors could find literature studies from 1990, exploring various perspectives related to this field. However each study ensures a lot of scope for further refinements in the field. This work is a sincere effort to analyze the systematic findings of different researchers in the related field. To identify the depth of systematic study performed by the researchers, the current work focuses on following research attributes like their major research objective in the field of software fault prediction, common literature studies adopted, proposed suggestions and major findings.

Authors have identified 11 systematic reviews from 2003 - 2015 for their study. The inclusion criteria for the selection of these review studies includes the title, keywords, objective of the study under consideration and number of citations. The objective of chosen studies, were to explore findings presented in previous studies. The current work is in-depth analysis of these systematic reviews. The objective of the current study is to analyze gap on the basis of findings and future scope given by individual studies for software fault prediction. The studies chosen were analyzed according to the publication year. Authors have explored them for findings and

Copyright © 2018 Authors. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

suggestion as an outcome of these systematic review studies. Authors have excluded all those manuscripts which does not meet the criteria especially papers without any citations are not considered. A good cited work ensures a good quality work. A work is considered as a quality work if it is easy to understand, with informative content and presents the outcome in a systematic way. Following is the flow of paper: Section 2 explains papers considered for the investigation in this study. Section 3 discuss the Research methodology, section 4 provides the findings and suggestions proposed by the studies in view related to software fault prediction problems and section 5 presents the conclusions and future direction.

2. Related Work

Using inclusion criteria 11 primary studies were selected for the investigation, out of which 112 secondary common studies were gathered. Authors have presented below a detailed study about the objective, method used and findings provided by these selected 11 primary studies:

(Catal&Diri, 2009) [1]

Objective: To provide a systematic review of software fault prediction. Focus of the study is to know about all machine learning methods, type of datasets and software metrics that have been majorly used by the studies in literature before and after year 2005, for software fault prediction.
Method: This is a time line study from year 1990-2008. The main objective of the study was to evaluate the literature work, in the field of software fault prediction.
Findings: The study witnesses an exponential increase in use of public data set and machine learning methods for software fault prediction since 2005. The method-level metrics are found to be the most dominant metrics in the field of fault prediction. According to the study more use of public datasets for on software fault prediction is required in combination to advance machine learning algorithms can present better fault predictor models. Study analyses that there is scope of work to done using class level metrics.

(Catal, 2011) [2]

Objective: This is a classification work. This work classify, papers included in the study. The categorization is on the basis of methodology used by the studies and their publication year **Method:** Software engineering literature is collected for software fault prediction and both machine learning and statistical based approaches used in literature are studied here. The published year of the papers under consideration is between 1990 -2009.
Findings: Author concludes that, most of the studies use method level metrics. Use of machine learning techniques is very high for prediction models. Naïve Bayes is suggested as a robust machine learning algorithm for supervised softwarefault prediction problem.

(Kitchenham, 2010) [3]

Objective: The objective of the study is to identify the most influential research work in literature related to software metrics. The aim of the study is to know the extent to which software metric studies can be aggregated.

Method: The inclusion criteria for selection of work, is SCOPUS highly cited papers from 2000 to 2005.Study critically analyzed the papers under review.

Findings: The work proposed a critical need for better collaboration among the previous work done and new studies in area of software metrics. Authors suggested a more systematic way of study is required.

(Malhotra , 2011) [4]

Objective: Studied the direction and intensity of work done in field of software fault prediction using object oriented metrics. Method: This study considered the journal and conference (ACM, IEEE, Springer) publications from 1998 to 2010 that are concerned with software fault prediction. The selection criteria were title, keywords and abstract of the relevant studies.
Findings: Authors analysed that most of the work in literature is done using CK metric suite. CK metrics are heavily and mostly used by the software developers in software development industry. With the availability of PROMISE repository by NASA, use of the machine learning methods has been increased since 2000. Study suggests that there is a need for more readily and freely available data sets for researchers to work upon and bring better insights to the field.

(Hall, 2012) [5]

Objective: Objective of the study is to analyze prediction models proposed in literature. Major concern of the study is: evaluating the context with which the models were developed. Study also aims to find the effect of independent variables and most prominent modeling techniques from the literature studies selected.
Method: Systematic literature review major fault prediction studies from January 2000 to December 2010.In total of 208 studies were studied.
Findings: Study analyzed that most of the work in literature lack in providing the contextual information. The published work is unable to clearly present the model for real use and even it is difficult for researcher to analyze across the model. Study raised many questions to be answer like how to build effective prediction model, a large set of predictive studies are required to have better quality and cost effective industrial software systems.

(Kumar, 2015) [6]

Objective: Systematic review conducted to collect evidence on software fault prediction techniques.
Method: Out of 577, only 15 papers (year 2002 to 2014) were selected by the study based on the selection criteria. The selection criteria are title abstract and keywords related to various software fault prediction methodology and techniques. The secondary selection criteria used is references and citations of these publications **Findings:** Findings of the study suggest that most of the literature work uses defect data set provided by promise repository of NASA. Most of the work done in literature predicts strong relation between various parameters of software metrics and fault prediction. Defect data set of CK metrics are major source of findings and model generation.

(Radjenovic, 2013) [7]

Objective: This study presents a systematic review on Software metrics their applicability in software fault Prediction.
Method: Total of 106 papers was analyzed published between 1991 and 2011. Inclusion criteria of the study are: it considered the empirical studies, both from academia and industry. It included those studies that empirically compared the software metrics for software fault prediction.

Findings: The major concern here was to classify the work across the metrics and context properties. Most of the studies under consideration properly used metrics to answer their Research Question. Studies lack in proper analysis and description of proposed models for software prediction using software metrics. Finding of the study suggest that literature under consideration were found to be weak in describing the outcomes properly.

(Malhotra, 2015) [8]

Objective: It is a comparative analysis of different machine learning techniques. Bayesian, Decision trees, Ensemble, Neural networks, Support vector machines, Rule based learning, Evolutionary algorithms are few ML techniques considered in the study.
Method: This is a systematic review from 1991 to 2013, including 139 studies. The work analyse 139 papers on the basis of Machine learning methods used and provide a comparative analysis of these methods.
Findings: The work has critically analysed research questions considered for the study. Research findings include that predictive capabilities of various ML techniques, have not considered in literature. Authors also suggested that clear contextual information is required in work so that published work can be successfully repeated by the software community.

(Juan Murillo-Morera, 2015) [9]

Objective: the research objective of the work is to know about the frequency of journal and conference, type of metrics and machine learning techniques used in literature.
Method: study identified a total of 89 literature studies from reputed sources and after study using inclusion criteria only 70 were actually studied here.
Findings: The study provided a statistical data about the software metrics, models and techniques used in literature.
Halstead, McCabe and LOC, software metrics and machine learning techniques are found dominant in literature studies.

(R. S. Wahono, 2015) [10]

Objective: the work identifies the major researcher in the field of software fault prediction. The findings of the work include type of data set used by the studies. And the major methods used in literature.
Method: This is a systematic review using publication from year 2000 to 2013.
Findings: include statistical findings like 77 percentages of the studies use classification methods. Most of the studies work on public dataset.

(Norman E. Fenton, 1991) [11]

Objective: The paper identifies various approaches and metrics and their relation with defect and failures of the software. Study suggested a model for software fault prediction.
Method: It is a study with critical analysis on publication from year 1979 to 1996.
Findings: Study recommends use of Bayesian network model.

3. Research Methodology

Authors included 11 studies (P1,P2,P3,P4,P5,P6,P7,P8,P9,P10,P11) [1-11], based on the inclusion criteria of the work : Title, keyword and no. of citations. The search is done majorly through Google search engine. Table 1 represents the list of all selected papers (P1-P11) for the study along with their title and citations. The publication year of these selected papers (P1-P11) is from year 2009 to 2014. A total of 578 published papers have been collected from reference list, of 11 selected papers (P1- P11). These papers are published between years 1990 to 2014.Literature study show that software quality predictions and software fault is not a new area of research.

Table 1: List of primary papers along with title and citation index

Paper ID	Title	Citations
P1 [1]	A systematic review of software fault prediction studies.	323
P2 [2]	Review: Software fault prediction: A literature review and current trends.	186
P3 [3]	What's up with software metrics? – A preliminary mapping study.	177
P4 [4]	Software Fault prediction for Object Oriented Systems: A Literature Review.	7
P5 [5]	A Systematic Literature Review on Fault Prediction Performance in Software Engineering.	448
P6 [6]	A Critical Analysis on Software Fault Prediction Techniques.	11
P7 [7]	Software Fault Prediction Metrics:A Systematic Literature Review.	167
P8 [8]	A systematic review of machine learning techniques for software fault prediction.	66
P9 [9]	Software Fault Prediction: A Systematic mapping study.	1
P10 [10]	A Systematic Literature Review of Software Defect Prediction: Research Trends, Datasets, Methods and Frameworks.	22
P11 [11]	A Critique of Software Defect Prediction Models.	1070

Figure 1 represents bar graph representation of year wise publications considered in the reference list (total of 578 publications) of 11 selected papers. The publication data of 578 papers is then arranged in sorted order of year of publication. A total combination of 112 secondary studies out of 578 studies is identified as common studies of 11 selected papers [1-112]. These selected 112 publications are the secondary references of the current work shown in Fig 1. the methods/ model used and kind of metrics used in the literature. All most all of the 11 systematic reviews considered here are published in recent years from 2009 to 2015. P11 is only paper that published in year 1999 with highest citation as 1070 (Table 1). Figure 1 represents the year wise 112 publication considered as secondary study. It is visible from curve that number of publication increased from 2004. Curve show highest publication in year 2007. Many studies conclude that software fault prediction studies exponentially increased after 2005 as NASA provided open access to its data set. Figure 2 represents curve for common publication from 112 secondary references. Curve represents an increase in common studies starting 2002. This implies that after 2002 many publications were available. These quality work were considered by most of the systematic reviews considered in current study.

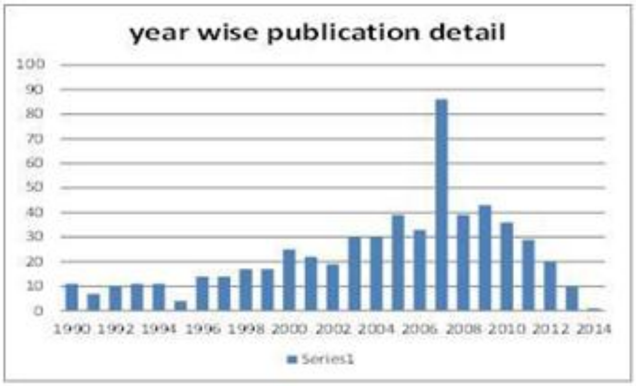

Fig. 1: Bar representation of year wise publication of 112 secondary papers.

4. Result Discussion

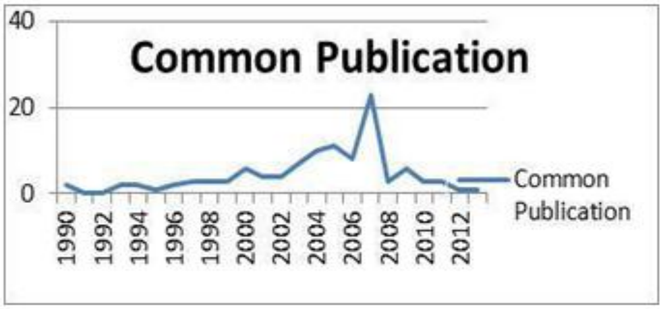

Fig. 2: Represents Curve for common publication in 11 systematic reviews selected for the study

Table 2: Table to represent the context of paper under study

#	Context	Paper ID
1	Journal/Papers	P1,P3,P9,P10
2	Datasets	P1,P4,P7,P10,P11
3	Methods/Modeling	P1,P2,P4,P5,P6,P9,P10,P
		11
4	Kind of metrics	P1,P3,P4,P5,P6,P7,P9,P1
		1
5	Cost and quality	P6

4.1 Major Findings

After thorough analysis of 11 systematic reviews for software fault prediction, it has been analyzed that major context of these studies are to find: journal/Papers in the area of software fault prediction journal, type of datasets used for fault prediction, methods/modeling techniques used for fault prediction, dominant software metrics that are used for fault prediction. Few more contexts on which various other studies conducted includes: feature reduction, fault distribution, performance measures, and frameworks for software fault prediction. Table 2 presents the five main contexts on which majority of the systematic reviews under study focus. It is clear from the Table 2 that majority of the papers like : P1,P2,P3,P4,P5,P6P7,P9,P10,P11 has done their study across On the basis of the findings and suggestion provided by 11 systematic reviews (Table 3) for software fault predations under selected studies, we listed few major findings these are: Use of the public data set has been in use since 2005.Most of the studies related to Software fault prediction or software metrics use Machine learning techniques for various analysis. Most of the studies use method level metrics for software related prediction.Many of the studies in field of software fault prediction and software metrics are unable to clearly specify there context. Findings of many systematics reviews conclude that publication in area of software fault predictions mostly Summarize ML techniques, asses performance accuracy and capabilities of Machine learning techniques and provide comparative studies for Software Fault Prediction models. In table 3, as most of the studies use defect data set provided by NASA, PROMISE repository, so it shows more public datasets is required for better results in this direction. Most Used Methods in Software Defect Prediction are Logistic Regression (LR), Naïve Bayes (NB), K-Nearest Neighbour (k-NN), Neural Network (NN), Decision Tree (DT), Support Vector Machine (SVM), Random Forest (RF)

Table 3: Represents Research questions, findings and suggestion mentioned in Paper P1-P11

Papers	Research	Findings	Suggestions
	questions		
P1	8	Public dataset, method	Yes, public data
		level metrics	sets,
P2	1	Public dataset, method	Yes, practical SFP
		level metrics	
P3	2	Need of collaborative	No
		approach	
P4	1	CK metrics dominant	No
P5	3	Lack of contextual	Lack in current
		information	knowledge
P6	4	Correlation between	No
		metrics and fault	
P7	3	Lack of proper	No
		analysis	
P8	6	Predictive capabilities	Contextual
		of ML	information
			required
P9	6	Strength and weakness	No
		of ML	
P10	9	Active researcher,	No
		number of	
		publication, research focus	
P11	1	Bayesian Network	No

5. Conclusion and Future Scope

The aim of the study was to analyze the systematic reviews done in literature for software fault prediction. With the selection criteria of Title "Systematic review on software fault prediction", key words used as software fault prediction, software metrics, and number of citations authors selected 11 systematic review studies for software fault prediction from year 2009 to 2015.From these 11 systematic review authors collected 587 secondary studies. Out of these secondary study's authors selected 112 studies which are commonly used by selected 11 systematic reviews Most of the published work is from the year 2005 onwards. The probable reason for such an increase in number of publication related to software fault prediction is due to availability of defect data sets by NASA. Most of the studies use available data set from NASA repository. Various studies present a predictive model using machine learning techniques, regression or statistical analysis. Large number of papers presents comparative study of classification accuracy using different classifiers. Most of the work published in literature, were not very clear in presenting their prediction model. There is a huge gap between the industry requirement and the research being done by the academic researchers in field of Software fault prediction. A better collaboration between industry academia is required. There is critical requirement of systematic and collaborative work to be done by researcher of the area. Software industry should provide freer data set for better work.

References

[1] C. Catal and B. Diri, “A systematic review of software fault prediction studies,” Expert Systems with Applications, vol. 36, no. 4, pp. 7346–7354, 2009

[2] C. Catal, Software fault prediction: a literature review and current trends, Expert systems with Applications 38(2011)4626-4636B.

[3] Kitchenham, “What’s Up with Software Metrics?—A Prelimin-ary Mapping Study” J. Systems and Software, vol. 83, no. 1, pp. 37-51, 2010T.

[4] Malhotra R, Jain A (2011) Software fault prediction for object oriented systems: a literature review. SIGSOFT SoftwEng notes 36(5): 1–6

[5] Hall, S. Beecham, D. Bowes, D. Gray, S. Counsell, A systematic literature review on fault prediction performance in software engineering, IEEE Transactions on Software Engineering (2011) 1–31P.

[6] Singh PK, Rajan P, Sangwan OP. A critical analysis on software fault prediction techniques.World applied Science Journal. 2015; 33(3):371-9

[7] Radjenović, D., Heričko, M., Torkar, R., &Živkovič, A. (2013, August). Software fault prediction metrics: A systematic literature review. Information and Software Technology. http://doi.org/10.1016/j.infsof.2013.02.009.

[8] RuchikaMalhotra, A systematic review of machine learning techniques for software fault prediction, Applied Soft Computing, v.27 n.C, p.504-518, February 2015

[9] Juan Murillo-Morera, Christian Quesada-Lopez, and Marcelo Jenkins. Software fault prediction: A systematic mapping study. CIBSE 2015 - XVIII Ibero-American Conference on Software Engineering, pages 446{459, 2015.

[10] R. S. Wahono. A systematic literature review of software defect prediction: Research trends, datasets, methods and frameworks. Journal of Software Engineering, 1(1):1--16, 2015.

[11] Norman E. Fenton , Martin Neil, A Critique of Software Defect Prediction Models, IEEE Transactions on Software Engineering, v.25 n.5, p.675-689, September 1999

Secondary References

[1] Kitchenham, B.A., Pickard, L.M., Linkman, S.J., 1990. An evaluation of some design metrics. Software Engineering Journal 5 (1), 50–58.

[2] Compton, B.T., Withrow, C.: Prediction and control of ada software defects. Journal of Systems and Software 12 (1990) 199–207

[3] Li, W., Henry, S., 1993. Object-oriented metrics that predict maintainability.Journal of Systems and Software 23, 111–122

[4] Briand, L., Basili, V., &Hetmanski, C. (1993).Developing interpretable models with optimized set reduction for identifying high risk software components. IEEE Transactions on Software Engineering, 19(11), 1028–1044.

[5] Chidamber, S.R., Kemerer, C.F., 1994. A metrics suite for object-oriented design. IEEE TSE 20 (6), 476–493.

[6] Lorenz, M. and Kidd, J. 1994. Object-Oriented Software Metrics.Prentice-Hall.

[7] Lanubile, F., Lonigro, A., &Visaggio, G. (1995). Comparing models for identifying fault-prone software components. In Seventh international conference on software engineering and knowledge engineering (pp. 312–319).

[8] Basili, V., Briand, L., Melo, W., 1996.A validation of object-oriented design metrics as quality indicators. IEEE Transactions on Software Engineering 22 (10), 751–761

[9] N. Ohlsson and H. Alberg “Predicting Error-Prone Software Modules in Telephone Switches, IEEE Trans. Software Eng., vol. 22, no. 12, pp. 886-894, 1996.

[10] Briand, L., Devanbu, P. and Melo, W. 1997. An Investigation into Coupling Measures for C++.In Proceedings of the 19th International Conference on Software Engineering.

[11] Khoshgoftaar, T.M., Allen, E.B., Hudepohl, J.P., Aud, S.J.: Application of neural networks to software quality modeling of a very large telecommunications system. Neural Networks, IEEE Transactions on 8 (1997) 902– 909.

[12] Cohen, W. W., &Devanbu, P. T. (1997). A comparative study of inductive logic programming methods for software fault prediction.In Fourteenth international conference on machine learning, Nashville, Tennessee, USA (pp. 66–74).

[13] Evett, M., Khoshgoftar, T., Chien, P., & Allen, E. (1998).GP-based software quality prediction. In Third annual conference on genetic programming (pp. 60–65).

[14] Ohlsson, N., Zhao, M., &Helander, M. (1998). Application of multivariate analysis for software fault prediction. Software Quality Journal, 7(1), 51–66.

[15] Binkley, A. B., &Schach, S. R. (1998). Prediction of run-time failures using static product quality metrics. Software Quality Journal, 7(2), 141–147.

[16] Tang, M.H., Kao, M-H. and Chen, M-H.1999. An Empirical Study on Object Oriented Metrics. In Proceedings of the International Symposium on Software Metrics.

[17] N. Fenton and M. Neil, “A critique of software defect prediction models,” Software Engineering, IEEE Transactions on, vol. 25, no. 5, pp. 675–689, 1999.

[18] Kaszycki, G. (1999). Using process metrics to enhance software fault prediction models.In Tenth international symposium on software reliability engineering, Boca Raton, Florida.

[19] Denaro, G. (2000). Estimating software fault-proneness for tuning testing activities.In Twenty-second international conference on software engineering (pp. 704–706). New York, NY: ACM

[20] Guo, P., &Lyu, M. R. (2000). Software quality prediction using mixture models with EM algorithm. In First Asia-Pacific conference on quality software (pp. 69–80). Hong Kong, China: IEEE Computer Society.

[21] X. Yuan, T. Khoshgoftaar, E. Allen, K. Ganesan, An application of fuzzy clustering to software quality prediction, in: Proceedings 3rd IEEE Symposium on Application-Specific Systems and Software Engineering Technology, 2000, pp. 85–90.

[22] Xu, Z., Khoshgoftaar, T. M., & Allen, E. B. (2000). Prediction of software faults using fuzzy nonlinear regression modeling. In Fifth IEEE international symposium on high assurance systems engineering, Albuquerque, New Mexico (pp. 281–290).

[23] Khoshgoftaar, T. M., Allen, E. B., &Busboom, J. C. (2000). Modeling software quality: The software measurement analysis and reliability toolkit.In Twelfth IEEE international conference on tools with artificial intelligence (pp. 54–61). Vancouver, BC, Canada: IEEE Computer Society.

[24] Cartwright, M. and Shepperd, M. 2000. An Empirical Investigation of an Object-Oriented Software System. IEEE Transactions on Software Engineering, 26(8):786-796.

[25] Emam, K. E., Melo, W., & Machado, J. C. (2001). The prediction of faulty classes using object-oriented design metrics. Journal of Systems and Software, 56(1), 63–75.

[26] Khoshgoftaar, T., Gao, K., &Szabo, R. M. (2001). An application of zero-inflated poisson regression for software fault prediction.In Twelfth international symposium on software reliability engineering (pp. 66– 73). Washington, DC: IEEE Computer Society.

[27] Briand, L.C., Wüst, J. and Lounis, H. 2001. Replicated Case Studies for Investigating Quality Factors in Object-Oriented

[28] Schneidewind, N. F. (2001). Investigation of logistic regression as a discriminant of software quality. In Seventh international symposium on software metrics (328–337). Washington, DC: IEEE Computer Society.

[29] Pizzi, N. J., Summers, R., &Pedrycz, W. (2002). Software quality prediction using median-adjusted class labels. In International joint conference on neural networks (pp. 2405–2409). Honolulu, HI: IEEE Computer Society.

[30] Khoshgoftaar, T. M. (2002). Improving usefulness of software quality classification models based on boolean discriminant functions. In Thirteenth international symposium on software reliability engineering (pp. 221– 230). Annapolis, MD, USA: IEEE Computer Society.

[31] Khoshgoftaar, T. M., &Seliya, N. (2002). Tree-based software quality estimation models for fault prediction. In Eighth IEEE symposium on software metrics, Ottawa, Canada (pp. 203– 215).

[32] Mahaweerawat, A., Sophasathit, P., &Lursinsap, C. (2002). Software fault prediction using fuzzy clustering and radial basis function network. In International conference on intelligent technologies, Vietnam (pp. 304–313).

[33] Denaro, G., Lavazza, L., &Pezzè, M. (2003). An empirical evaluation of object oriented metrics in industrial setting. In The 5th CaberNet plenary workshop, Porto Santo, Madeira Archipelago, Portugal.

[34] Denaro, G., Pezzè, M., &Morasca, S. (2003). Towards industrially relevant faultproneness models. International Journal of Software Engineering and Knowledge Engineering, 13(4), 395–417.

[35] Guo, L., Cukic, B., & Singh, H. (2003). Predicting fault prone modules by the Dempster–Shafer belief networks.In Eighteenth IEEE international conference on automated software engineering (pp. 249–252). Montreal, Canada: IEEE Computer Society.

[36] Khoshgoftaar, T. M., &Seliya, N. (2003). Fault prediction modeling for software quality estimation: Comparing commonly used techniques. Empirical Software Engineering, 8(3), 255–283.

[37] Koru, A. G., &Tian, J. (2003). An empirical comparison and characterization of high defect and high complexity modules. Journal of Systems and Software, 67(3), 153– 163.

[38] Reformat, M. (2003). A fuzzy-based meta-model for reasoning about number of software defects.In Tenth international fuzzy systems association world congress, Istanbul, Turkey (pp.644–651).

[39] Thwin, M. M., &Quah, T. (2003). Application of neural networks for software quality prediction using object-oriented metrics.In Nineteenth international conference on software maintenance (pp. 113–122). Amsterdam, The Netherlands: IEEE Computer Society.

[40] Kaminsky, K., &Boetticher, G. (2004). How to predict more with less, defect prediction using machine learners in an implicitly data starved domain. In The 8th world multiconference on systemics, cybernetics and informatics, Orlando, FL.

[45] Kanmani, S., Uthariaraj, V. R., Sankaranarayanan, V., &Thambidurai, P. (2004). Object oriented software quality prediction using general regression neural networks. ACM SIGSOFT Software Engineering Notes, 29(5), 1. http://doi.org/10.1145/1022494.1022515

[46] Khoshgoftaar, T. M., &Seliya, N. (2004). Comparative assessment of software quality classification techniques: An empirical case study. Empirical Software Engineering, 9(3), 229–257.

[47] Mahaweerawat, A., Sophatsathit, P., Lursinsap, C., &Musilek, P. (2004). Fault prediction in object-oriented software using neural network techniques. In Proceedings of the InTech conference, Houston, TX, USA (pp. 27– 34).

[48] L. Guo, Y. Ma, B. Cukic, H. Singh, Robust prediction of fault-proneness byrandom forests, in: 15th International Symposium on Software ReliabilityEngineering, 2004, pp.417–428.

[49] Mahaweerawat, A., Sophatsathit, P., Lursinsap, C., &Musilek, P. (2004). Fault prediction in object-oriented software using neural network techniques. In Proceedings of the InTech conference, Houston, TX, USA (pp. 27– 34).

[50] Menzies, T., & Di Stefano, J. S. (2004). How good is your blind spot sampling policy? In Eighth IEEE international symposium on high-assurance systems engineering (pp. 129–138). Tampa, FL, USA: IEEE Computer Society.

[51] Menzies, T., DiStefano, J., Orrego, A. S., & Chapman, R. (2004). Assessing predictors of software defects.In Proceedings of the Workshop on Predictive Software Models.

[52] Wang, Q., & Yu, B. (2004). Extract rules from software quality prediction model based on neural network. 16th IEEE International Conference on Tools with Artificial Intelligence, (Ictai), 191–195. http://doi.org/10.1109/ICTAI.2004.62

[53] Zhong, S., Khoshgoftaar, T. M., &Seliya, N. (2004). Unsupervised learning for expertbased software quality estimation.In Eighth IEEE international symposium on high assurance systems engineering (pp. 149–155).

[54] Challagulla, V. U., Bastani, F. B., Yen, I., & Paul, R. A. (2005).Empirical assessment of machine learning based software defect prediction techniques. In Tenth IEEE international workshop on object-oriented real-time dependable systems (pp. 263–270). Sedona, Arizona: IEEE Computer Society.

[55] I. Myrtveit, E. Stensrud, and M. Shepperd, “Reliability and validity in comparative studies of software prediction models,” IEEE Transactions on Software Engineering, pp.380–391, 2005.

[56] Koru, A. G., & Liu, H. (2005). An investigation of the effect of module size on defect prediction using static measures.In Workshop on predictor models in software engineering, St. Louis, Missouri (pp. 1–5).

[57] A.G. Koru, H. Liu, Building effective defect-prediction models in practice, IEEESoftw. 22 (2005) 23–29.

[58] Gyimo´ thy, T., Ferenc, R. and Siket, I. 2005 Empirical Validation of Object-Oriented Metrics on Open Source Software for Fault Prediction. IEEE Transactions on Software Engineering, 31(10):897-910.

[59] Hassan, A. E., & Holt, R. C. (2005). The top ten list: Dynamic fault prediction. In Twenty-first IEEE international conference on software maintenance (pp. 263–272). Budapest, Hungary: IEEE Computer Society.

[60] Khoshgoftaar, T. M., Seliya, N., &Gao, K. (2005). Assessment of a new three-group software quality classification technique: An empirical case study. Empirical Software Engineering, 10(2), 183–218.

[61] Koru, A. G., & Liu, H. (2005). An investigation of the effect of module size on defect prediction using static measures.In Proceedings of the 2005 workshop on Predictor models in software engineering - PROMISE ’05 (Vol. 30, pp. 1–5). New York, New York, USA: ACM Press. http://doi.org/10.1145/1082983.1083172

[62] Ostrand, T. J., Weyuker, E. J., & Bell, R. M. (2005). Predicting the location and number of faults in large software systems. IEEE Transactions on Software Engineering, 31(4), 340–355.

[63] Tomaszewski, P., Lundberg, L., &Grahn, H. (2005). The accuracy of early fault prediction in modified code.In Fifth conference on software engineering research and practice in Sweden, Västerås, Sweden (pp. 57–63).

[64] Xing, F., Guo, P., &Lyu, M. R. (2005). A Novel Method for Early Software Quality Prediction Based on Support Vector Machine. 16th IEEE International Symposium on Software Reliability Engineering (ISSRE’05), 213–222. http://doi.org/10.1109/ISSRE.2005.6

[65] Ma, Y., Guo, L., &Cukic, B. (2006). A statistical framework for the prediction of fault-proneness.Advances in machine learning application in software engineering (pp. 237–265). Idea Group Inc.

[66] Y.M. Zhou, H.T. Leung, Empirical analysis of object-oriented design metrics for predicting high and low severity faults, IEEE Transactions on Software Engineering 32 (2006) 771–789.

[67] Boetticher, G. (2006). Improving credibility of machine learner models in software engineering. Advanced machine learner applications in software engineering.Series on software engineering and knowledge engineering. Hershey, PA, USA: Idea Group Publishing

[68] Challagulla, V. U. B., Bastani, F. B., & Yen, I. L. (2006). A unified framework for defect data analysis using the MBR technique.In Eighteenth IEEE international conference on tools with artificial intelligence, Washington, DC, USA (pp. 39–46).

[69] Khoshgoftaar, T. M., Seliya, N., &Sundaresh, N. (2006). An empirical study of predicting software faults with case based reasoning. Software Quality Journal, 14(2), 85–111.

[70] Mertik, M., Lenic, M., Stiglic, G., &Kokol, P. (2006). Estimating software quality with advanced data mining techniques. In International conference on software engineering advances (p. 19). Papeete, Tahiti, French Polynesia: IEEE Computer Society.

[71] Nikora, A. P., & Munson, J. C. (2006). Building high-quality software fault predictors. Software-Practice and Experience, 36(9), 949–969.

[72] Shepperd, M., Cartwright, M., &Mair, C. (2006). Software defect association mining and defect correction effort prediction. IEEE

Transactions on Software Engineering, 32(2), 69–82. http://doi.org/10.1109/TSE.2006.1599417

[73] B. Kitchenham and S. Charters, “Guidelines for performing systematic literature reviews in software engineering (version 2.3),” Keele University, UK, Tech. Rep. EBSE Technical Report EBSE-2007-01, 2007.

[74] G. Pai, B. Dugan, Empirical analysis of software fault content and fault proneness using Bayesian methods, IEEE Transactions on Software Engineering 33 (2007) 675–686.

[75] H.M. Olague, L.H. Etzkorn, S. Gholston, S. Quattlebaum, Empirical validation of three software metrics suites to predict fault-proneness of object-oriented classes developed using highly iterative or agile software development processes, IEEE Transactions on Software Engineering 33 (2007) 402–419.

[76] Binkley, D., Feild, H., &Lawrie, D. (2007). Software fault prediction using language processing. Testing: Industrial and academic conference. Practice and research techniques (pp. 99–108). Cumberland Lodge, Windsor, UK: IEEE Press.

[77] Kanmani, S., Uthariaraj, V.R., Sankaranarayanan, V. and Thambidurai, P. 2007. Object-oriented software fault prediction using neural networks. Information and Software Technology, 49: 483–492.

[78] Koru, A. G., & Liu, H. (2007). Identifying and characterizing change-prone classes in two large-scale open-source products. Journal of Systems and Software, 80(1), 63–73.

[79] M. Jorgensen and M. Shepperd, “A systematic review of software development cost estimation studies,” Software Engineering, IEEE Transactions on, vol. 33, no. 1, pp. 33–53, 2007.

[80] Gao, K., &Khoshgoftaar, T. M. (2007). A comprehensive empirical study of count models for software fault prediction. IEEE Transactions for Reliability, 56(2), 223– 236.

[81] H. Zhang and X. Zhang, “Comments on "data mining static code attributes to learn defect predictors",” Software Engineering, IEEE Transactions on, vol. 33, no. 9, pp. 635 –637, sept. 2007.

[82] Jiang, Y., Cukic, B., &Menzies, T. (2007). Fault prediction using early lifecycle data. In Eighteenth IEEE international symposium on software reliability (pp. 237– 246).Trollhättan, Sweden: IEEE Computer Society.

[83] Kitchenham B.A., Charters S.M., (2007). Guidelines for Performing Systematic Literature Reviews in Software Engineering, Version 2.3. EBSE Technical Report EBSE-2007-01, Keele University and Durham University.

[84] Li, Z., & Reformat M. (2007). A practical method for the software fault-prediction.In IEEE international conference on (pp. 659–666).

[85] Ma, Y., Guo, L., &Cukic, B. (2007). A Statistical Framework for the Prediction of Fault-Proneness. In Advances in Machine Learning Applications in Software Engineering (pp. 1–26)

[86] Mahaweerawat, A., Sophatsathit, P., &Lursinsap, C. (2007). Adaptive self-organizing map clustering for software fault prediction.In Fourth international joint conference on computer science and software engineering, KhonKaen, Thailand (pp. 35–41).

[87] Menzies, T., Dekhtyar, A., Distefano, J., & Greenwald, J. (2007b). Problems with precision: A response to comments on data mining static code attributes to learn defect predictors. IEEE Transactions on Software Engineering, 33(9), 637–640.

[88] Menzies, T., Greenwald, J., & Frank, A. (2007). Data Mining Static Code Attributes to Learn Defect Predictors. IEEE Transactions on Software Engineering, 33(1), 2– 13.

[89] Pai, G. J., & Dugan, J. B. (2007). Empirical analysis of software fault content and fault proneness using Bayesian methods. IEEE Transactions on Software Engineering, 33(10), 675–686.

[90] Seliya, N., &Khoshgoftaar T. M. (2007). Software quality analysis of unlabeled program modules with semisupervised clustering. IEEE Transactions on Systems, Man, and Cybernetics, 37(2), 201–211.

[91] T. Menzies, J. Greenwald, A. Frank, Data mining static code attributes to learndefect predictors, IEEE Trans. Softw. Eng. 33 (2007) 2–13.

[92] Tomaszewski, P., Håkansson, J., Grahn, H., & Lundberg, L. (2007). Statistical models vs. expert estimation for fault prediction in modified code – An industrial case study. Journal of Systems and Software, 80(8), 1227– 1238.

[93] Wang, Q., Zhu, J., & Yu, B. (2007). Feature selection and clustering in software quality prediction. In Eleventh international conference on evaluation and assessment in software engineering, Keele, England.

[94] Y. Jiang, B. Cukic, T. Menzies, Fault prediction using early lifecycle data, in: The 18th IEEE International Symposium on Software Reliability (ISSRE ’07), 2007, pp. 237–246.

[95] Zhang, H., & Zhang, X. (2007). Comments on data mining static code attributes to learn defect predictors. IEEE Transactions on Software Engineering, 33(9), 635– 637.

[96] A. Marcus, D. Poshyvanyk, R. Ferenc, Using the conceptual cohesion of classes for fault prediction in object-oriented systems, IEEE Transactions on Software Engineering 34 (2008) 287–300.

[97] Bibi, S., Tsoumakas, G., Stamelos, I., &Vlahvas, I. (2008). Regression via classification applied on software defect estimation. Expert Systems with Applications, 34(3), 2091–2101.

[98] R. Shatnawi, W. Li, The effectiveness of software metrics in identifying errorprone classes in post-release software evolution process, Journal of Systems and Software 81 (2008) 1868–1882.

[99] B. Turhan, T. Menzies, A. Bener, and J. Di Stefano, “On the relative value of cross-company and within-company data for defect prediction,” Empirical Software Engineering, vol. 14, no. 5,pp. 540–578, 2009.

[100] C. Catal and B. Diri, “A systematic review of software fault prediction studies,” Expert Systems with Applications, vol. 36, no. 4, pp. 7346–7354, 2009.

[101] Catal, C., &Diri, B. (2009b).Investigating the effect of dataset size, metrics sets, and feature selection techniques on software fault prediction problem. Information Sciences, 179(8), 1040–1058. http://doi.org/10.1016/j.ins.2008.12.001

[102] Chang, C., Chu, C., &Yeh, Y. (2009).Integrating in-process software defect prediction with association mining to discover defect pattern. Information and Software Technology, 51(2), 375–384.

[103] Mende, T., &Koschke, R. (2009).Revisiting the evaluation of defect prediction models. Proceedings of the 5th International Conference on Predictor Models in Software Engineering - PROMISE ’09, 1. http://doi.org/10.1145/1540438.1540448

[104] Turhan, B., Kocak, G., &Bener, A. (2009). Data mining source code for locating software bugs: A case study in telecommunication industry. Expert Systems and Application, 36(6), 9986–9990.

[105] Singh, Y., Kaur, A. and Malhotra, R. 2010.Empirical validation of object-oriented metrics for predicting fault proneness models. Software Quality Journal, 18(1):3–35.

[106] Arisholm, E., Briand, L. C., &Johannessen, E. B. (2010).A systematic and comprehensive investigation of methods to build and evaluate fault prediction models. Journal of Systems and Software, 83(1), 2–17. http://doi.org/10.1016/j.jss.2009.06.055

[107] N. Seliya, T.M. Khoshgoftaar, J. Van Hulse, Predicting faults in high assurancesoftware, in: IEEE 12th International Symposium on High-Assurance SystemsEngineering, 2010, pp. 26–34.

[108] C. Catal, Software fault prediction: a literature review and current trends,Expert Syst. Appl. 38 (2011) 4626– 4636.

[109] D. Gray, D. Bowes, N. Davey, Y. Sun, and B. Christianson, “The misuse of the nasa metrics data program data sets for automated software defect prediction,” in Evaluation and Assessment in Soft-ware Engineering (EASE), 2011.

[110] Song, Q., Jia, Z., Shepperd, M., Ying, S., & Liu, J. (2011).A General Software Defect-Proneness Prediction Framework. IEEE Transactions on Software Engineering, 37(3), 356–370.

[111] Hall, T., Beecham, S., Bowes, D., Gray, D., &Counsell, S. (2012). A Systematic Literature Review on Fault Prediction Performance in Software Engineering. IEEE Transactions on Software Engineering 38 (6), 1276 1304.

[112] Radjenovi´c, D., Heriˇcko, M., Torkar, R., ˇZivkoviˇc, A.: Software fault prediction metrics: A systematic literature review. Information and Software Technology 55 (2013) 1397–1418