

Predicting Open Source Forked Pattern Survability

by Bee Bee Chua

Thesis submitted in fulfilment of the requirements for
the degree of

Doctor of Philosophy

under the supervision of Professor Ying Zhang & Associate
Professor Lu Qin

University of Technology Sydney
Faculty of Engineering and Information Technology

October 2021

CERTIFICATE OF ORIGINAL AUTHORSHIP

I, Bee Bee Chua, declare that this thesis, is submitted in fulfilment of the requirements for the award of Doctor of Philosophy in Information Technology, in the School of Computer Science at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

Signature: Production Note:
Signature removed prior to publication.

Date: 21st October 2021

Acknowledgements

Really, I have many people I wish to say thank you but to the very special and honourable ones are below:

Firstly, my research supervisory team, Professor Ying Zhang and Associate Professor Lu Qin, for their kind guidance and support throughout my PhD journey. Especially to Professor Zhang, a truly amazing and highly intellectual supervisor who I find to be extremely hardworking, critical, humble and knowledgeable. I am deeply indebted and wish to acknowledge his high-quality supervisory efforts, support, encouragement, belief and trust in me that I can produce good quality research.

Secondly, my family. I am grateful to my parents and my family members, including five adorable nieces and a nephew, for their continued love and support. To my special god daughter Maris Stella who I fondly miss.

To the team of medical specialists for their amazing expertise in healing my neck and right arm injury. My neurosurgeon Dr. Prakash Damodaran, my pain specialist Dr. Hasher Kadvil, my general practitioners Dr. Pramod Singh and Dr. Deep Kumar for a successful medical procedure and to my most respectful hand therapist Miss Tamara Carter for her hard work on healing my neck and hand. To all nurses and front desk receptionist staff for cheering me up during that most difficult and challenging time. Without them, my thesis writing is impossible.

To my research collaborators and mentors at UTS for advice and teaching collaboration, including Mr. Ravindra Bagia, Professor Roger Hadgraft, Dr. Danilo Valeros Bernardo, Dr. Jane Brennan. Dr. Laurel Dyson, Dr. Yulei Sui and Dr. Wentao. Li.

To many other researchers who I know sincerely to thank them for their esteemed support and encouragement: Professor June Verner, Professor Fethi Rabhi, Professor Paul Rowland, Professor Aileen Cartel-Steel, Associate Professor Shannon Kennedy-Clark, Professor Mehregan Mahdavi and Associate Professor Andrew Levula.

My heartfelt gratitude to Dr. Amy Nisselle for her expertise on guiding me how to write for an academic audience, copyediting and proofing my conference papers and thesis.

Finally, my deepest gratitude to Dr. Danilo Valeros Bernardo for his years of unwavering endearment and patience with me. Without his words of kind encouragement and motivation, I would not be able to achieve this degree.

List of Publications

This thesis comprised of a series of published and to-be-published articles together with an exegesis. The list of the publications is as follows:

1. B. B. Chua and Y. Zhang, “Applying a systematic literature review and content analysis method to analyse open source developers’ forking motivation interpretation, categories and consequences.” *Journal of Australasian Information Systems*, 2020. Vol. 24 No.1, pp. 1–19.
2. B. B. Chua, and Y. Zhang, “Predicting open source programming language repository file survivability from forking data.” *OpenSym’19: Proceedings of the 15th International Symposium on Open Collaboration*. 2019. Skövde Sweden
3. B. B. Chua, “A survey paper on open source forking motivation reasons and challenges.” *Conference Proceedings of the Pacific Asia Conference of Information Systems (PACIS)*, 2017, Langkawi, Malaysia
4. B. B. Chua. Detecting sustainable programming languages through forking on open source projects for survivability. *Proceedings of the IEEE International Symposium on Software Reliability Engineering (ISSRE)* in conjunction with a WOSAR workshop, IEEE, 2015, Gaithersburg, USA. 120–124
5. B. B. Chua and Y. Zhang. “Healthy Fork File Repository (HFFR) Performance Prediction”. *Journal of Systems and Information Technology (JST)* Elsevier, Under review.

Other relevant publications developed but not included in this thesis include:

1. B. B. Chua. “Analysing Version Control Open-Source Software Survivability”. Proceedings of the 19th International Conference on Distributed Multimedia Systems, DMS 2013, August 8-10 2013, Brighton, UK. Knowledge Systems Institute.
2. B. B. Chua and D.V. Bernardo. Open-Source Developer Download Tiers: A Survival Framework. 13th IEEE International Conference on IT Convergence and Security, ICITCS, 2013, Macau, China.

Contents

.....	i
CERTIFICATE OF ORIGINAL AUTHORSHIP	ii
Acknowledgements	iv
List of Publications	vi
Contents	viii
List of Figures	xi
List of Tables	xii
List of Acronyms.....	xiv
Abstract.....	1
Chapter 1	3
1.0 Introduction	3
1.1 Background.....	5
1.2 Research Motivation.....	6
1.3 Research Contributions.....	7
Chapter 2: Forking Literature Survey.....	8
2.1 Overview	8
2.2 Motivation	8
2.3 Approaches	8
2.4 Introduction	9
2.5 Research Study Motivation and Research Questions	12
2.5.1 Research study motivation	12
2.5.2 Research questions	13
2.6 Methodology: Systematic Literature Review and Content Analysis Method	15
2.6.1 Systematic literature review search criteria	17
2.6.2 Search strategy	18
2.6.3 Methodological framework	20
2.6.4 Content analysis method	21
2.7 Forking Motivation Interpretations.....	22
2.7.1 How do researchers interpret developer forking and categorise forking motivational behaviour?	23
2.7.2 What were the most popular methodologies used by forking researchers from 1990 to 2017?.....	32

2.7.3 What aspects of open source forking have been researched and reported?	32
2.7.4 Newcomers or new developers forking motivation from 2020 to 2021	34
2.7.5 Shifting motivation through time and journey	35
2.7.6 Shifting forking motivation.....	36
2.8 Summary from the literature survey	37
Chapter 3: Literature Survey Research Methodology	40
3.1 Overview	40
3.2 Motivation	40
3.3 Introduction	40
3.4 Literature Survey Selection Criteria and Categorisation	41
3.5 Category I: Survey-based Research Methodology	42
3.6 Category II: Data Mining Algorithm-based Research Methodology.....	50
3.7 Category III: Machine Learning Algorithm-based Research Methodology	58
3.8 Machine Learning: A K Nearest Neighbour Method.....	65
3.8.1 Euclidean distance metric	66
3.8.2 Adopting Euclidean distance: characteristics identification and rationale	66
3.8.3 Identifying Euclidean distance characteristics	67
3.8.4 Our research dataset characteristics	67
Chapter 4: Models	69
4.1 Overview	69
4.2 Literature Survey Road Map Model	69
4.3 Chua and Zhang Open Source Software Forking Pattern Prediction Model	70
Chapter 5: A Pilot Study	72
5.1 Overview	72
5.2 Motivation	72
5.3 Background.....	73
5.4 Forking Patterns.....	74
5.5 Software Survival and Programming Language Survival Importance.....	75
5.6 Survivability Prediction on the K Nearest Neighbour Method	77
5.7 Programming Language Repository File Categorisation and Fork Pattern Classifiers.....	81
5.8 Classifier Results	82
5.9 K Nearest Neighbour Results.....	83

5.9.1 Case One	84
5.9.2 Case Two	85
5.9.3 Case Three	85
5.9.4 Case Four	85
5.10 Evaluation.....	86
5.11 Conclusions and Future Work	88
Chapter 6: A Longitudinal Study	90
6.1 Overview	90
6. 2 Motivation	90
6.3 Background.....	91
6.4 Fork Pattern Identification and Data Collection	92
6.5 Normalisation and Euclidean Distance.....	95
6.6 Results	100
6.7 Evaluative Test Results	102
6.8 Discussion.....	104
Chapter 7: Conclusion.....	106
7.1 Overview	106
7.2 Contributions	107
7.3 Recommendations	108
7.4 Future Work.....	109
Bibliography	110

List of Figures

Figure 2. 1: Combined approaches: systematic literature review and content analysis method	16
Figure 2. 2: The systematic literature review search strategy for research papers.....	19
Figure 2. 3: Data collection methods in the 21 papers.....	32
Figure 2. 4: Units of analysis in the 21 papers.....	33
Figure 2. 5: Forking lessons learnt across the 21 papers.....	34
Figure 2. 6: The open source developers' motivation movement.....	36
Figure 3. 1: Paper selection criteria	42
Figure 4. 1: Literature survey mapping model.....	70
Figure 4. 2: The Chua and Zhang OSS forking pattern prediction model	71
Figure 5. 1: Categorising programming language repository file forks as short- or long-lived.....	82
Figure 5. 2: Evaluative results comparison of the dataset.....	88
Figure 6. 1: Euclidean distance ranking.....	102
Figure 6. 2: Evaluative results	104

List of Tables

Table 2. 1: The systematic literature review identified 21 relevant and suitable papers	19
Table 2. 2 A forking motivation methodological framework	20
Table 2. 3: Forking interpretation types.....	21
Table 2. 4: Fork categorisation, sustainability and lessons learnt.....	28
Table 3. 1: Literature Survey Research Methodology in OSS	46
Table 3. 2: Data Mining algorithm-based type research methodology	55
Table 3. 3: Machine learning research-based methodology in OSS	61
Table 3. 4: Four widely-adopted KNN distance metrics.....	66
Table 5. 1: Fork patterns	75
Table 5. 2: Variables defined for programming language survivability	79
Table 5. 3: Forking patterns	81
Table 5. 4: Categorising programming language repository files forks as short- or long-lived.....	82
Table 5.5: Categorising programming language repository files sorted by Euclidean distance.....	84
Table 5. 6: Environment compliance	86
Table 6. 1: Examples of file repository monthly forking.....	93
Table 6. 2: Big query statement	93
Table 6. 3: Forking data of selected file repositories, 2015–2020	94
Table 6. 4: Variables defined for a healthy fork file repository	96
Table 6. 5: Forking in 5 years (2015-2020) after normalisation	97

Table 6. 6: Healthy fork file repository types and counts	99
Table 6. 7: Healthy fork file repository types ranked by Euclidean distance	100
Table 6. 8: Healthy fork file repository types ranked by Euclidean distance	101
Table 6. 9: Definition and formula for accuracy, precision, sensitivity and specificity	103

List of Acronyms

CAM	Content Analysis Method
CVS	Control Version System
FN	False Negative
FP	False Positive
HFFR	Healthy File Fork Repository
KNN	K Nearest Neighbour Method
OS	Open Source
OSS	Open Source Software
SLR	Systematic Literature Review
SPF	Specific Repository File
SRFHF	Specific Repository File that did not meet the full environment licence but has Healthy Fork
SRFMSPL	Specific Repository File that met official licence compliance and adopted a Modern Sustainable Programming Language
SRFOL	Specific Repository File that met Official Licence compliance
SRFOLHF	Specific Repository File that met Official Licence compliance that has Healthy Fork
SRFOLMSPLHF	Specific Repository File that met Official Licence compliance and adopted a Modern Sustainable Programming Language that has Healthy Fork
SRFOLTSPL	Specific Repository File that met Official Licence and adopted a Traditional Sustainable Programming Language
SRFTSPL	Specific Repository File that adopted a Traditional Sustainable Programming Language
SRFTSPLHF	Specific Repository File that adopted a Traditional Sustainable Programming Language that has Healthy Fork
TN	True Negative
TP	True Positive
VT	Virus Total

Abstract

The motivational behaviour of open source (OS) developers has always been an active focus of research. With the introduction of the forking technique a related research area of developer forking motivational behaviour has gained significance, partly due to the problem of forking scarcity and low fork visibility performance.

The objective of forking is to improve and innovate source code quality from voluntary developers. Unfortunately, the forking technique is not very sustainable in improving fork efficiency and efficacy. Further, developers may spend time forking source codes that may become inactive and consequently prove to be a waste of time and effort. From the perspective of project owners, if their repositories do not receive a good fork response from developers, their repositories will not grow.

This doctoral research study aimed to address these problems by avoiding forking scarcity, increasing high fork visibility performance, and promoting positive developer forking motivation. We also needed to investigate OS environment compliance to determine whether it contributes to improved fork visibility, reduced fork deficiency and/or is viewed positively by developers.

The research approach was to apply a model to predict high fork visibility. The model is based on the K Nearest Neighbour machine learning algorithm, using the Euclidean distance metric to predict high fork visibility performance. We piloted it using nine repository classifiers and then conducted a longitudinal study of five select repository classifiers to determine accuracy and distance approximation. Our work adds a new body of knowledge to OS forking theory and provides a deeper understanding of developer forking motivational behaviour.

In the first phase of this study, we conducted a literature review of forking motivation and research methods used in OSS. We then developed and tested our model. In the last phase, we identified OSS patterns and detected fork longevity to determine whether environmental compliance was fully, partially or not at all satisfied. Most importantly, we showed that high fork visibility environmental compliance distance approximation can positively predict developer forking interest.