INFORMATION TECHNOLOGY, MANAGEMENT AND OPERATIONS RESEARCH PRACTICES

MULTI-CRITERIA DECISION MODELS IN SOFTWARE RELIABILITY

Methods and Applications

Edited by

Ashish Mishra

Nguyen Thi Dieu Linh

Manish Bhardwaj

Carla M. A. Pinto

CRC Press
Taylor & Francis Group

Multi-Criteria Decision Models in Software Reliability

This book provides insights into contemporary issues and challenges in multi-criteria decision models. It is a useful guide for identifying, understanding and categorising multi-criteria decision models and ultimately for implementing the analysis for effective decision-making.

The use of multi-criteria decision models in software reliability engineering is a relatively new field of study, and this book collects all the latest methodologies, tools and techniques in one single volume. It covers model selection, assessment, resource allocation, release management, upgrade planning, open-source systems, bug tracking system management and defect prediction.

Multi-Criteria Decision Models in Software Reliability: Methods and Applications will cater to researchers, academicians, postgraduate students, software developers, software reliability engineers and IT managers.

Information Technology, Management and Operations Research Practices

Series Editors:
Vijender Kumar Solanki, Sandhya Makkar, and Shivani Agarwal

This new book series will encompass theoretical and applied books and will be aimed at researchers, doctoral students and industry practitioners to help in solving real-world problems. The books will help in various paradigms of management and operations. The books will discuss the concepts and emerging trends on society and businesses. The focus is to collate the recent advances in the field and take the readers on a journey that begins with understanding the buzz words such as employee engagement, employer branding, mathematics, operations and technology and how they can be applied in various aspects. It walks readers through engaging with policy formulation, business management and sustainable development through technological advances. It will provide a comprehensive discussion on the challenges, limitations and solutions to everyday problems such as how to use operations, management and technology to understand the value-based education system, health and global warming and real-time business challenges. The book series will bring together some of the top experts in the field throughout the world who will contribute their knowledge regarding different formulations and models. The aim is to provide the concepts of related technologies and novel findings to an audience that incorporates specialists, researchers, graduate students, designers, experts and engineers who are occupied with research in technology-, operations- and management-related issues.

Sustainability, Big Data, and Corporate Social Responsibility
Evidence from the Tourism Industry
Edited by Mohammed El Amine Abdelli, Nadia Mansour, Atilla Akbaba, and Enric Serradell-Lopez

Entrepreneurial Innovations, Models, and Implementation Strategies for Industry 4.0
Edited by Ravindra Sharma, Geeta Rana, and Shivani Agarwal

Multi-Criteria Decision Models in Software Reliability
Methods and Applications
Edited by Ashish Mishra, Nguyen Thi Dieu Linh, Manish Bhardwaj, and Carla M. A. Pinto

For more information about this series, please visit: https://www.routledge.com/Information-Technology-Management-and-Operations-Research-Practices/book-series/CRCITMORP

Multi-Criteria Decision Models in Software Reliability

Methods and Applications

Edited by
Ashish Mishra, Nguyen Thi Dieu Linh,
Manish Bhardwaj, and Carla M. A. Pinto

CRC Press
Taylor & Francis Group
Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an **informa** business

First edition published 2023
by CRC Press
6000 Broken Sound Parkway NW, Suite 300, Boca Raton, FL 33487-2742

and by CRC Press
4 Park Square, Milton Park, Abingdon, Oxon, OX14 4RN

CRC Press is an imprint of Taylor & Francis Group, LLC

Library of Congress Cataloging-in-Publication Data
Names: Mishra, Ashish (Ashish Kumar), editor. | Dieu Linh, Nguyen Thi, editor. | Bhardwaj, Manish (Professor of computer science and engineering), editor. | Pinto, Carla M. A. (Computer scientist), editor.
Title: Multi-criteria decision models in software reliability : methods and applications / edited by Ashish Mishra, Nguyen Thi Dieu Linh, Manish Bhardwaj, and Carla M.A. Pinto.
Description: First edition. | Boca Raton : CRC Press, 2023. | Includes bibliographical references.
Identifiers: LCCN 2022022574 (print) | LCCN 2022022575 (ebook) | ISBN 9780367408824 (hardback) | ISBN 9781032342542 (paperback) | ISBN 9780367816414 (ebook)
Subjects: LCSH: Computer software—Reliability—Mathematical models. | Computer software—Development—Decision making. | Multiple criteria decision making.
Classification: LCC QA76.76.R44 M85 2023 (print) | LCC QA76.76.R44 (ebook) | DDC 005—dc23/eng/20220915
LC record available at https://lccn.loc.gov/2022022574
LC ebook record available at https://lccn.loc.gov/2022022575

ISBN: 978-0-367-40882-4 (hbk)
ISBN: 978-1-032-34254-2 (pbk)
ISBN: 978-0-367-81641-4 (ebk)

DOI: 10.1201/9780367816414

Typeset in Times
by codeMantra

Contents

Preface

It is a matter of pleasure for us to put forth the book titled, *Multi-Criteria Decision Models in Software Reliability: Methods and Applications*. In the present era, software reliability plays a vital role in solving different kinds of problems and providing promising solutions in digital world. Because of the increase in digitalisation in today's lifestyle and each and every service to make the life easier, good software interfaces are required. Due to the increase in the usability and dependency on software, one important feature matters a lot, that is software reliability. The success of incorporation of the heavy software in the system works only with reliability feature. Such reliability depends upon different criteria and the deployed environment. It does not always relate to one or two factors, but it depends upon various factors such as physical or virtual.

This book explores various factors and criteria within different chapters related to reliability and decision-making steps. These aspects make decision-making approaches more powerful, reliable and efficient. The above-mentioned characteristics make the software reliability approaches more suitable and competent for decision-making systems. Nowadays, machine learning is incorporated in each and every field of engineering to make the automated system for better decision-making solutions. This kind of system provides the efficient decision in less time. Medical science and engineering have been using various medical systems such as medical imaging devices, medical testing devices and medical information systems. In order to analyse such big data efficiency, image processing, signal processing and data mining play important roles for computer-aided diagnosis and monitoring.

Decision-making in the medical field is a very important part because it is directly related to human life, so monitoring and diagnosis software should be reliable enough to provide the correct reports. This book will enable the reader to appreciate the applications of multi-criteria decision models in software reliability and their different methods used in various fields according the field criteria.

CHAPTER 1

This chapter focuses on building an item-item recommender system using collaborative filtering. The proposed model uses the well-known MovieLens dataset and also uses the concept of Bayesian average for evaluating movie popularity. In order to deal with the problem of sparsity, our proposed model builds compressed sparse row (CSR) matrix. This chapter uses machine learning approach using K-nearest neighbours for recommending movies based on similarity.

CHAPTER 2

This chapter focuses on the examination of relevant literature and provides a conceptual framework that explains the role of machine learning and profound learning in the development of intelligent (artificial) beings.

CHAPTER 3

This chapter reviews the various classifications used to predict software defects using software measurements in the literature. In this chapter, a detailed analysis of application of data mining and machine learning approaches used for software quality, defect and quality analysis is presented.

CHAPTER 4

This chapter analyses the types of ambiguities that arise due to poor management of requirement engineering and how it affects software quality and customer satisfaction. Moreover, it discusses the challenges an enterprise faces when, in prototype model, new feature are added continuously based on business requirements.

CHAPTER 5

This chapter describes the integration of multi-criteria decision making (MCDM)-based fuzzy analytic hierarchy process (FAHP) and fuzzy Technique for Order Preference by Similarity to Ideal Solution (FTOPSIS) methods that are applied for the formation or selection of best group of programmers.

CHAPTER 6

This chapter intends to use one of the unknown yet powerful machine learning algorithms, MCDM, to foresee the presence of heart disease in a person more accurately in order to save more lives by detecting and treating the patient before any major issue.

CHAPTER 7

In this chapter, the classification of software reliability models (SRMs) is studied on the basis of effective and efficient quality of SR models and obtains software faults with categorisation of vast variety of available software.

CHAPTER 8

This chapter provides a detailed study of different types of reliability models, which are responsible for the software reliability measurements. As every model has different criteria, so no single model is perfect. It also provides information about software quality improvement.

CHAPTER 9

This chapter shows the comparison of different techniques to resolve vulnerabilities using different multi-criteria decision analysis (MCDA) methods. The MCDM saves and sorts the list of criteria affecting the environments.

CHAPTER 10

This chapter describes and gives possible approaches for the safety assessment of AI systems. The AI system to integrate safety level needs and used for probabilistic failure behaviour for the dangerous part of the random budget for failure relevant in AI system.

CHAPTER 11

In this chapter, a step-by-step model for the FDP and FCP is proposed based on the ANN. The test initiative is taken into account as it has a strong impact on the error detection and correction process.

CHAPTER 12

In this chapter, various MCDM methodologies are studied with different performance parameters along with the new methodology FMCDM and its applications. The new methodology is compared with the traditional methodologies.

CHAPTER 13

In this chapter, to extend the capabilities of large-scale application and fix any faults detected during operation, software systems with optimisation help in selecting new techniques constantly for improving the next release sequence of plan, which is a huge challenge for firms developing or managing such vast and sophisticated systems.

CHAPTER 14

In this chapter, modelling data are evaluated with a deep neural network algorithm that is created expressly to predict the amount of faults, and the fault-free software system is finalised.

CHAPTER 15

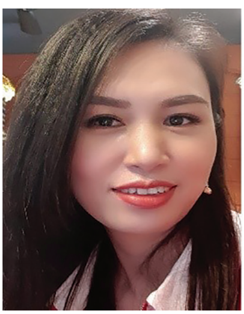
This chapter reviews the recent technologies and uses deep learning mechanisms to detect vulnerabilities. It shows how they apply state-to-state neural techniques that are helpful for capturing probable vulnerable codes and patterns. It also provides complete reviews of the visions, concepts and ideas of the game modifiers for their field of interest.

We sincerely thank Ms. Erin Harris, Senior Editorial Assistant, CRC Press/Taylor & Francis Group, for giving us an opportunity to convene this book in her esteemed publishing house and for their kind cooperation in completion of this book, and Dr. Vijender Kr. Solanki, Sandhya Makkar and Shivani Agarwal, Series Editors in IT, Management and Operation Research. We thank our esteemed authors for having shown confidence in this book and considering it as a platform to showcase and share their original research work. We would also wish to thank the authors whose papers were not published in this book, probably because of minor shortcomings.

Editors

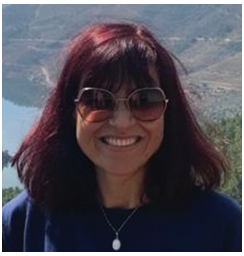
Dr. Ashish Mishra is currently working as a Professor in the Department of Computer Science and Engineering, Gyan Ganga Institute of Technology and Sciences, Jabalpur [M.P], India.

He is a qualified individual with around 19 years of expertise in teaching and R&D with specialisation in Computer Science Engineering. He completed B.E., M.Tech. and MBA. He received his Ph.D. degree from AISECT University, Bhopal, India. He has been a part of various seminars, webinars, paper presentations, research paper reviews and conferences as co-convener, Member of Organizing Committee, Member of Advisory Committee and Member of Technical Committee, and he has contributed to organising INSPIRE Science Internship Camp. He is a Senior Member of IEEE, Life Member of CSI and Secretary CSI Jabalpur Chapter. He has published many research papers in reputed journals and conferences. He also has papers in Springer and IEEE conferences. He is also a reviewer and Session Chair, Keynote Speaker of IEEE, Springer international conferences, CSNT-2015, CICN-2016, CICN2017, INDIACom-2019, ICICC-CONF 2019, ICICC-CONF 2020 and ICICC-CONF 2021. His research interests include IoT, data mining, cloud computing, image processing and knowledge-based systems. He published 30 patents in Intellectual Property India. He has published 8 books in the area of data mining, image processing and artificial intelligence.



Dr. Nguyen Thi Dieu Linh is currently working as a Dy. Head of Science and Technology Department, Hanoi University of Industry, Vietnam (HaUI). She received her Ph.D. in Information and Communication Engineering from Harbin Institute of Technology, Harbin, China. She has more than 19 years of academic experience in electronics, IoT, telecommunication, big data and artificial intelligence. She has published more than 30 research articles in national and international journals, books and conference proceedings. She is a reviewer for Information Technology Journal, Mobile Networks and Applications Journal and some international conferences. Now, she is an editor for some books such as *Artificial Intelligence Trends for Data Analytics Using Machine Learning and Deep Learning Approaches*; *Distributed Artificial Intelligence: A Modern Approach* published by Taylor & Francis Group, LLC; and *Data Science and Medical Informatics in Healthcare Technologies* published by Springer. Otherwise, she is an editor of International Journal of Hyperconnectivity and the Internet of Things (IJHIoT) IGI-Global, the USA.

Dr. Manish Bhardwaj is currently working as a Research Assistant Professor in the Department of Computer Science and Engineering, KIET Group of Institutions, Muradnagar, Ghaziabad, India. He is a qualified individual with around 11 years of expertise in teaching and R&D with specialisation in Computer Science Engineering. He received his Ph.D. degree from Dr. Abdul Kalam Technical University (AKTU), Lucknow, India. He completed M.Tech. (Computer Science & Engineering) from SRM University, Chennai (Gold Medalist, received award from former central health minister Mr. Gulam Nabi Azad). He is contributing to the scientific community by his enormous academics and research works in the areas of computer science, simulations, mobile ad hoc network protocols and wireless sensor networks. He has published nearly 60 Research Papers in various international journals/conferences. He has also taken part in nearly 150 international conferences and journals as General Chair, International Scientific Committee Members/Reviewer (SCOPUS index journals and conferences) and Editorial Board Member/Reviewer in reputed journals such as IEEE and Springer. He has contributed 1 book as an Editor and 8 book chapters in various renowned publications such as CRC Press and IGI Global. He has nearly 14 patents (ten national + four international).



Dr. Carla M.A. Pinto is a Coordinating Professor in the School of Engineering at Polytechnic of Porto, Portugal. Her main research topic is epidemiology, in particular Mathematical Epidemiology. She is interested in mathematical challenges and their role in providing advice on public health policies. Mrs. Pinto is trained in Nonlinear Dynamics, Bifurcation Theory. Previous research included the analysis of Central Pattern Generators for Animal and Robot Locomotion, coupled cell networks, and neuron-like equations (Hodgkin-Huxley equations, Fitz-Hugh Nagumo, and Morris-Lecar). She is an Associate Editor of international journals with a high impact factor. She is the Guest Editor of several books. She has published more than 100 articles. Her h-index is 20 and she has over 1700 citations.

Contributors

Aakriti
Bharati Vidyapeeth's College of Engineering
New Delhi, India

D. Akila
Department of Computer Applications
Saveetha College of Liberal Arts and Sciences
SIMATS deemed to be University
Chennai, India

Isha Bansal
Bharati Vidyapeeth's College of Engineering
New Delhi, India

L. Bhagyalakshmi
Rajalakshmi Engineering College
Chennai, India

S. K. Bharadwaj
Madhav Institute of Technology & Science
Gwalior, India

Manish Bhardwaj
KIET Group of Institutions, Delhi-NCR
Ghaziabad, India

Korhan Cengiz
University of Fujairah
Fujairah, UAE

Deviprtiya
Vels Institute of Science, Technology and Advanced Studies
Chennai, India

V.R. Elangovan
Agurchand Manmull Jain College
Chennai, India

Kartik Gupta
Bharati Vidyapeeth's College of Engineering
New Delhi, India

Sardar M. N. Islam
ISILC
Victoria University
Melbourne, Australia

Rachna Jain
Bhagwan Parshuram Institute of Technology
Delhi, India

S. Jeyalaksshmi
Vels Institute of Science, Technology and Advanced Studies
Chennai, India

Padmaja Joshi
CDAC
Mumbai, India

Rachana Kamble
Technocrats Institute of Technology
Bhopal, India

Aarti M. Karande
Sardar Patel Institute of Technology
Mumbai, India

Devansh Kashyap
Kalinga Institute of Industrial Technology
Bhubaneswar, India

Shreyansh Keshri
Kalinga School of Management
Bhubaneswar, Odisha, India

Hassan Raza Mahmood
FAST NUCES Chiniot-Faisalabad Campus
Chiniot-Faisalabad, Pakistan

Ashish Mishra
Gyan Ganga Institute of Technology and Sciences
Jabalpur, India

D. K. Mishra
Madhav Institute of Technology & Science
Gwalior, India

Jyoti Mishra
Gyan Ganga Institute of Technology and Sciences
Jabalpur, India

Nishchol Mishra
RGPV
Bhopal, India

Neelu Nihalani
RGPV
Bhopal, India

Samad Noeiaghdam
Irkutsk National Research Technical University
Irkutsk, Russia
and
South Ural State University
Chelyabinsk, Russia

D. Padmapriya
Vels Institute of Science, Technology and Advanced Studies
Chennai, India

Vishal Paranjape
RGPV
Bhopal, India

Piramu Prithika
Vels Institute of Science, Technology and Advanced Studies
Chennai, India

Prashant Richhariya
Technocrats Group of Institutions
Bhopal, India

Harish K. Shakya
Amity University
Gwalior, India

Aditi Sharma
Parul University
Vadodara, Gujarat, India

Saurabh Sharma
Amity University
Gwalior, India

Vineet Sharma
KIET Group of Institutions, Delhi-NCR
Ghaziabad, India

Himanshu Shekhar
Hindustan Institute of Technology and Science
Chennai, India

Vikas Shinde
Madhav Institute of Technology & Science
Gwalior, India

Rajeev Shrivastava
Princeton Institute of Engineering & Technology for Women
Hyderabad, India

Ragini Shukla
Dr. C. V. Raman University
Chhattisgarh, India

Shweta Singh
KIET Group of Institutions, Delhi-NCR
Ghaziabad, India

Shubham Singh
Galgotias University
Greater Noida, India

Anurag Sinha
Department of Computer Science
IGNOU
New Delhi, India

Anita Soni
IES University
Bhopal, India

Sanjay Kumar Suman
St. Martin's Engineering College
Hyderabad, India

Kshitij Tandon
Jaypee University of Engineering and Technology
Guna, India

Narina Thakur
Bhagwan Parshuram Institute of Technology
Delhi, India

1 Enhancing Software Reliability by Evaluating Prediction Accuracy of CBF Algorithm Using Machine Learning

Vishal Paranjape, Neelu Nihalani and Nishchol Mishra
RGPV

CONTENTS

DOI: 10.1201/9780367816414-1

1.1 INTRODUCTION

A vital factor affecting system reliability is software reliability. Alternatively, it is described as the likelihood of software being successfully executed for a particular instant of time. Several techniques were proposed for determining the software's reliability. A particular task is fulfilled by a software system in a particular environment for predefined number of input cases is termed as software reliability. A very important connection to software reliability is software quality, comprising functionality, usability, performance, etc. Software quality hinders the growth of software reliability. It is difficult to reach certain level of reliability with any system with a complexity. The machine learning approach guarantees to predict accurate solution to a given problem and therefore is a promising approach for ensuring software reliability. Today, machine learning approaches are used in a number of applications; one of the most used approaches is recommender systems where a user is being recommended items on the basis of his/her purchasing history of buying habits. A number of applications such as e-commerce, movies recommendation and social networking such as Facebook make use of recommender systems.

The entire chapter is divided into the following sections: Section 1.2 deals with the background details. Section 1.3 presents the ML techniques and methodology used for reliability assessment in our proposed work. The experimental set-up is discussed in Section 1.4. Results are represented in Section 1.5. Section 1.6 concludes the chapter.

1.2 BACKGROUND DETAILS & RELATED WORK

1.2.1 Software Reliability

An important feature for enhancing software quality is ensuring software reliability dealing with the bugs present in the system [1]. Fault in code is the major reason for failure in the system. Analytical models are used to measure the reliability of software termed as software reliability growth models (SRGMs) [2,3].

1.2.2 Criterion to Measure Performance of SGRM

Past research presented several techniques to acquire software reliability, but to access it and estimate mean time to failure (MTTF), we use a mathematical model called SRGM. There are two categories of SGRMs on the basis of nature of process:

1. Times between failures models
2. Fault count models.

Some well-known SRGMs are Goel-Okumoto, Musa-Okumoto, Jelinski-Moranda, etc. For deciding reliability level and to stop testing, we use these models [4].

For evaluating the performance of various models, we use several criteria such as root-mean-square error (RMSE), mean absolute error (MAE), average error (AE), and normalised root-mean-square error (NRMSE). Our proposed model uses only RMSE and MAE approach for evaluating the performance. The mathematical equations for the above-mentioned techniques are given below.

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{N}\left(x_i - \hat{x}_i\right)^2}{N}} \tag{1.1}$$

where

i = Variable
N = Number of non-missing data points
x_i = Actual rating
$\hat{x}_i$ = Predicted rating.

$$\text{MAE} = \frac{\sum_{i=1}^{N}\left|\left(p_i(f) - a_i(f)\right)\right|}{N} \tag{1.2}$$

$$\text{NRMSE} = \frac{\sqrt{\sum_{i=1}^{k}\left(p_i(f) - a_i(f)\right)^2}}{\sum_{i=1}^{k} p_i(f)^2} \tag{1.3}$$

where

k = Number of failures
$a_i(f)$ = Number of actual failures
$p_i(f)$ = Number of predicted failures.

1.3 MACHINE LEARNING: A BRIEF OVERVIEW

A technique that is capable of learning from training data and predicting results is called machine learning. Broadly, we classify machine learning into four categories, which are discussed in the next section. Further, subcategorisation of the different types of ML is depicted in Figure 1.1 below. Under uncertainty, this technique plays a vital role in prediction and decision-making. On the basis of type of data and questionnaire being asked, different taxonomies of ML are available, which classifies machine learning. The classification of ML is given in Figure 1.1.

1.3.1 Supervised Learning

In this method, we use labelled data with the help of which we train our model. In other words, we can say the learning that takes place in the presence of a supervisor is called supervised learning. The major part of this type of learning includes mapping function, which maps I/P variable (X) with the O/P variable (Y).

$$Y = f(X)$$

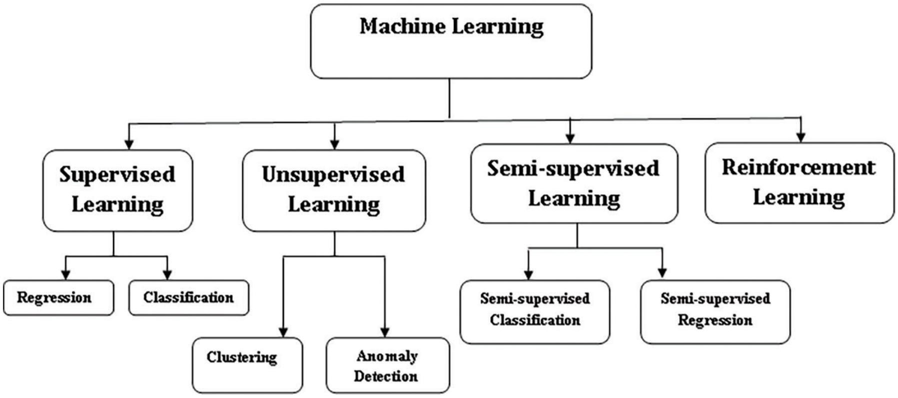


FIGURE 1.1 Categories of machine learning.

Supervision for model training is the main step involved in supervised learning. It can be simulated with the fact that proper learning takes place in the presence of a teacher or mentor in school. Two problems come in this category: **classification** and **regression**.

1. **Classification Models**: The problems in which output variables can be classified as "Yes" or "No", or "Pass" or "Fail" are categorised as classification models. In order to predict data category, we use these models. These can be binary classification or multiclass classification models. Some well-known examples for classification models that are deployed are spam filtering in emails, churn prediction, etc.
2. **Regression Models**: Whenever the output is predicted based on the previous data, we use the concept of regression models, for example house rent prediction. Linear, polynomial, ridge and logistic regression are some of the more familiar regression algorithms.

Regression problems are all about predicting *f%* for a quantitative response, such as blood pressure and temperature. For prediction, many ML algorithms are available, ranging from simple linear regression (LR) [5] and polynomial response surface (PRS) [6] to more complex support vector regression (SVR) [7], decision tree regression (DTR) [8], and random forest regression (RFR) [9]. By accurately quantifying uncertainty in regression problems, we use some machine learning (ML) models [10,11]. DNNs are more reliable than conventional ML equivalents and are effective in controlling the overfitting issue [12] (Figure 1.2).

1.3.2 Unsupervised Learning

The learning that takes place in the absence of a supervisor is called unsupervised learning; in this type of learning, we do not have labelled data. This technique does not provide any training data. A large volume of data is fed to the machine for

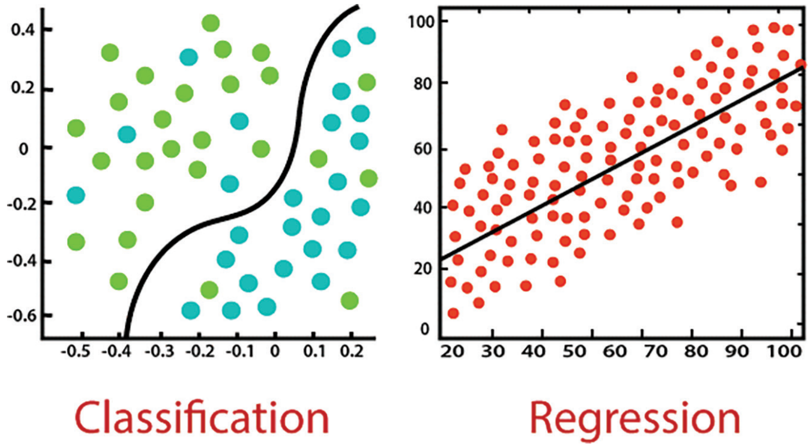


FIGURE 1.2 Classification and regression model.

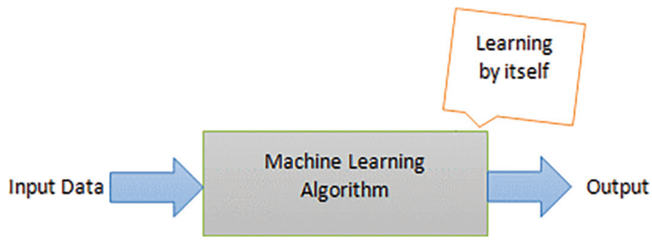


FIGURE 1.3 Unsupervised learning.

developing model and patterns, and on the basis of this learning, the model is fed with the testing data so as to provide efficient predictions. In unsupervised learning, there are no defined outcomes; moreover, it determines whatever different or interesting patterns exist in a given data set. Recommender system is basically based on the concept of unsupervised learning where we use several algorithms such as *k*-means clustering and *k*-nearest neighbours (Figure 1.3).

1.3.2.1 Categorisation of Unsupervised Machine Learning

1. Of all the learning methods, clustering is an important unsupervised learning method. Organising unlabelled data into similar groups is the main task of clustering technique. Therefore, collection of similar data items is called clustering. Grouping of similar data points into cluster and finding similar data points is the main goal of clustering.
2. The technique of identification of rare items or events differing from majority of data is called anomaly detection. Since anomalies or outliers are suspicious, generally we look for them. Bank fraud and medical error detection generally uses anomaly detection techniques.

1.3.3 Semi-Supervised Learning

A technique comprising of mix up of labelled data and unlabelled data during the phase of training is called semi-supervised learning. In this technique, first, the model is trained with the training data and then it is fed with the testing data to get the predictions.

To produce improvement and accuracy in learning, we use unlabelled data. A skilled human agent is required for acquiring labelled data for a learning problem or a physical experiment. It is relatively inexpensive to acquire unlabelled data.

A text document classifier is an example of this type of learning. It is so because it is not time efficient to have a person read the entire document. So, with the help of labelled text it becomes easy to classify labelled text with unlabelled (Figure 1.4).

1.3.4 Reinforcement Learning

An interactive environment using hit and trial is learning which comes under the category of reinforcement learning (RL) and is an ML technique. Mapping between input and output is provided by both supervised and reinforcement learning where we give feedback to the agent. These feedbacks are of two types: Whenever there is a

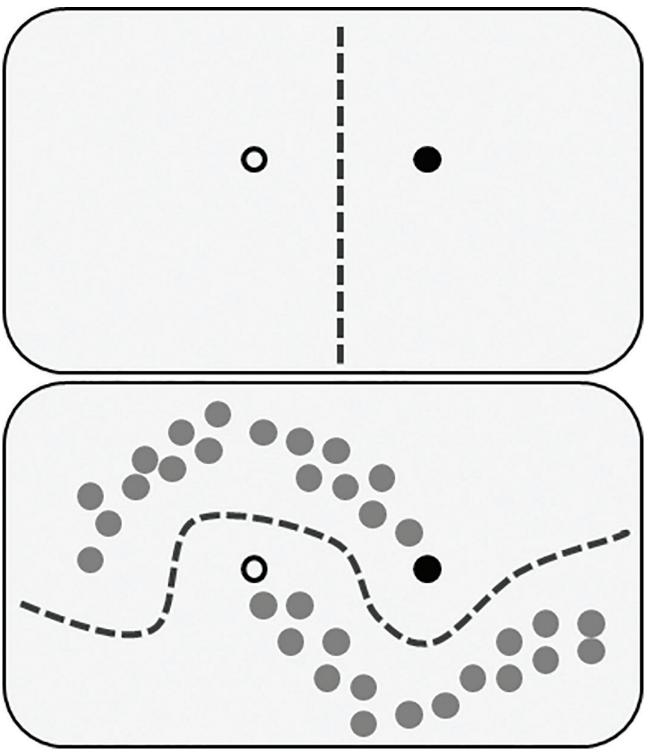


FIGURE 1.4 Semi-supervised learning.

FIGURE 1.5 Reinforcement learning.

positive reward, then that type of performance is repeated, while if there is negative impact of a work, then it is avoided (Figure 1.5).

1.3.4.1 Algorithms Used in Machine Learning

Some commonly used machine learning algorithms are discussed below:

1. Linear Regression

   This technique estimates the exact values, for example total sales prediction and cost of houses, on the basis of continuous variables. The best line is fitted to depict the relationship between two variables. The line is also called regression line shown by the linear equation

   $$Z = m * X + c$$

   where Z is dependent on the values of X and c, and m is the slope.

   For example, if we give an assignment to a student studying in fifth class to separate people according to their weight, then he on the basis of his skills will arrange people and separate them on the basis of their height and weight to classify them just by visualisation. This is a real-life application seen for linear regression. Figure 1.6 given below depicts a simple linear regression.
2. Logistic Regression

   As many a time we get confused by the name regression, whereas in real, it is a classification algorithm. Discrete values comprising values such as 0/1, yes/no and true/false are estimated by logistic regression. The probability of occurrence of event is predicted by fitting data. As this method is basically based on probability, its value generally lies between 0 and 1 (Figure 1.7).
3. Decision Tree

   A well-known algorithm used for classification problems is decision tree. Here, the entire population is split into two or more homogenous sets. In the diagram depicted below, we can see how a decision tree works. For

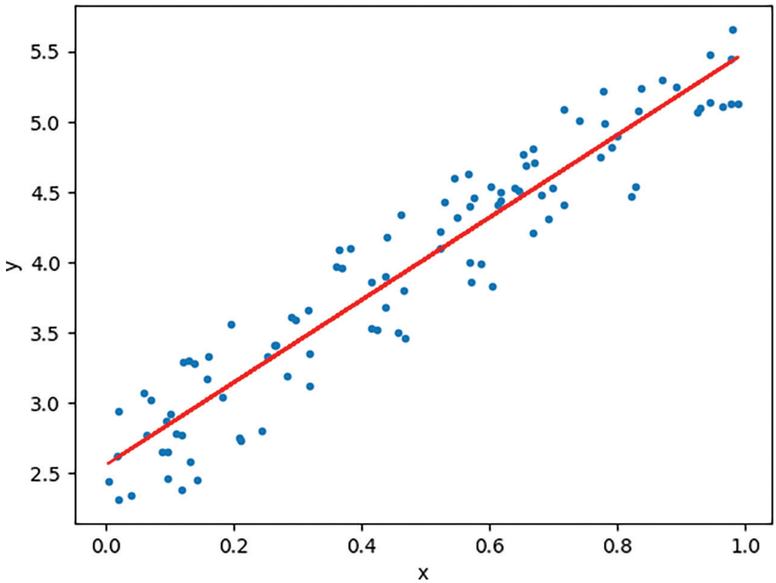


FIGURE 1.6 Linear regression.

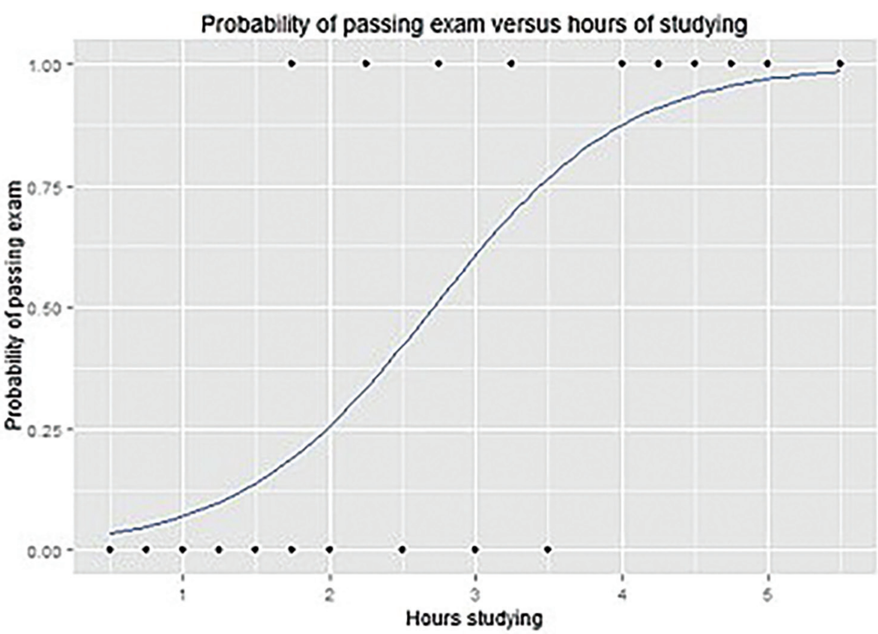


FIGURE 1.7 Logistic regression.

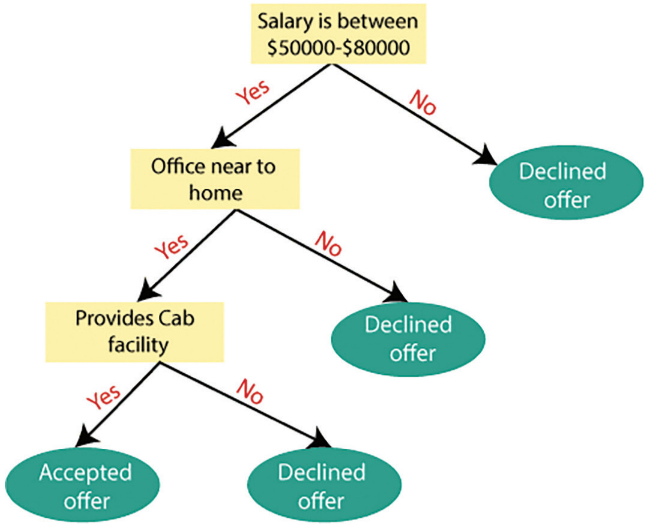


FIGURE 1.8 Decision tree.

example, if an employee is offered a salary between $50000 and $8000 and if his office is near to his home and if office provides cab facility, then the probability of that employee for taking offer letter is more, whereas if the salary is not in that range, he would have not accepted the offer; moreover, if his office was also far from his home, he would have declined the offer and if cab was not provided, still he would have declined offer (Figure 1.8).

4. SVM (Support Vector Machine)

   It divides two items on the basis of their best line or decision boundary called hyperplane. In *n*-dimensional space, there can be several lines/decision boundaries to separate the groups, but we need to find the best decision boundary to help define the data points. The hyperplane of SVM refers to the best boundary (Figure 1.9).

5. Naive Bayes

   A method of classification based on Bayes' theorem is called naive Bayes. This technique assumes that a particular feature in a class is not related to another. For calculating posterior probability, we use Bayes' theorem. It is given below in the form of equation:

$$T(m|n) = \frac{P(n \mid m)P(m)}{P(n)}$$

Here, $P(n|m)$ = Posterior probability
$P(m)$ = Prior probability of class
$P(n|m)$ = Likelihood which is probability of predictor
$P(n)$ = Prior probability of predictor.

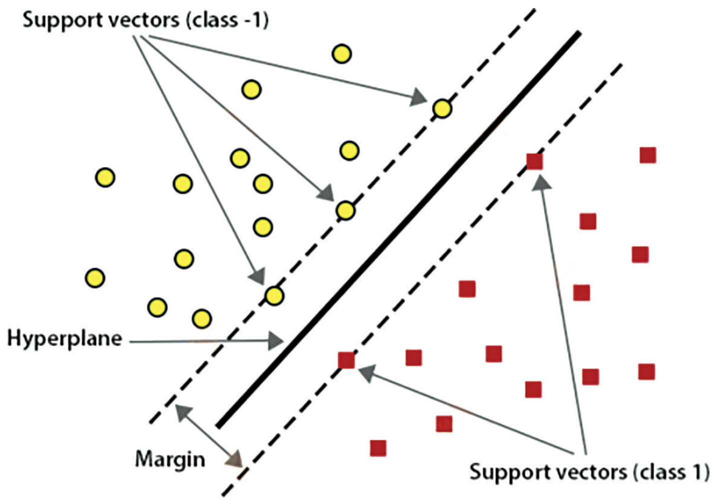


FIGURE 1.9 Support vector machine.

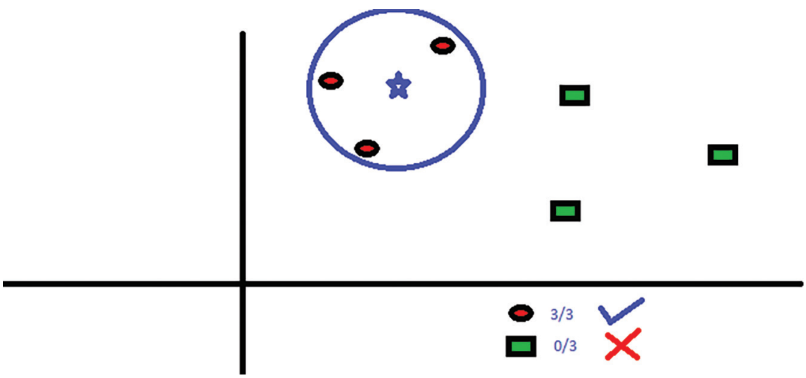


FIGURE 1.10 *k*-Nearest neighbours.

6. *k*NN (*k*-Nearest Neighbours)

   It is a classification problem using classification and regression problems. *k*-Nearest neighbours algorithm involves finding the distance from the data points, and for that, we use Euclidean, Manhattan and Hamming distances. For the sake of convenience, we take an odd value of *k* such as 3 or 5 to distinguish between two different types of items (Figure 1.10).

7. *k*-Means

   For solving clustering problem, we use this type of unsupervised algorithm. With the help of certain number of clusters, we can classify the data set using this technique assuming *k* number of clusters; therefore, its name became *k*-means algorithm. Figure 1.11 below depicts three prominent clusters where each cluster is shown by same coloured data points.

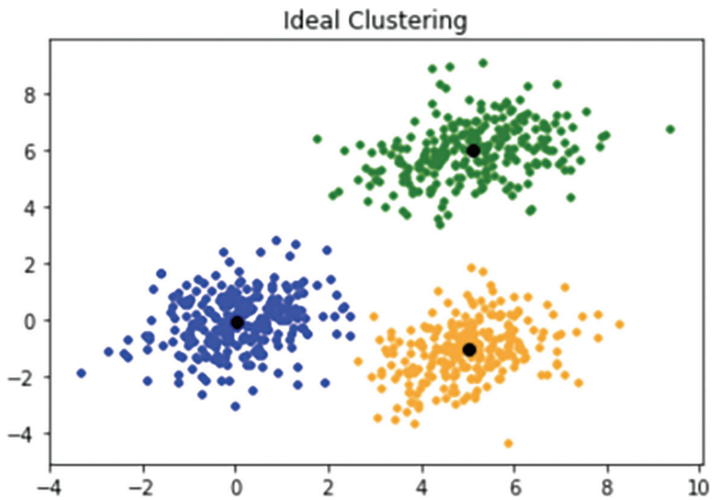


FIGURE 1.11 *k*-Means clustering.

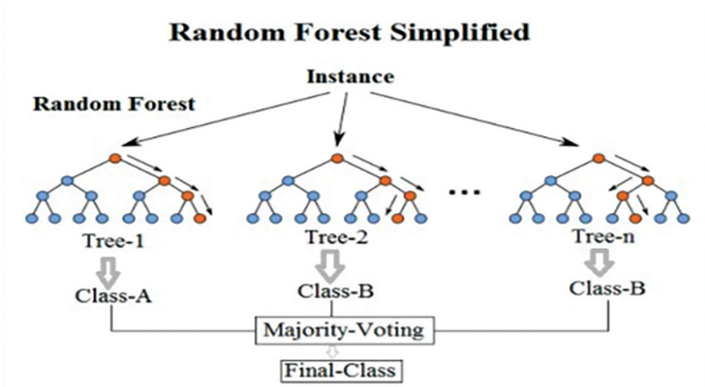


FIGURE 1.12 Random forest.

8. Random Forest

   When we talk about ensembling, then random forest is the most widely used algorithm in supervised machine learning. A collection of decision trees is called a random forest. Classification is given in tree for classifying new object, and we say tree "votes" for that class. These have much more accuracy with respect to decision trees, but lower than gradient boosted trees (Figure 1.12).

1.4 RELATED WORK

There are several works done by several researchers in the field of collaborative filtering-based recommender system. Most of the work based on movie recommendation is based on the concept of personalisation, which suggests movies to users on the basis of their interest and likings.

A k-means clustering-based hybrid recommender system was proposed by Katarya Rahul [13] and was applied to the MovieLens data set with optimisation technique of bio-inspired artificial bee colony.

Ponnam et al. [14] suggested a collective filtering technique based on an item that examines the user's item rating matrix and determines the relationship between different objects in order to calculate the user's recommendations.

A content-based movie recommender framework was proposed by Bagher Rahimpour Cami et al. [15] capturing user choices in temporary mode in user modelling and predicting favourite movies.

Reddy et al. [16] used a genre correlation technique by using the method of content-based filtering.

A weighted hybridisation-based hybrid recommender system was proposed by Hong-Quan Do et al. [17], which didn't use fixed weight and aimed to provide a simple way to dynamically weight the combination of Collaborative Filtering and Content Based Filtering.

An effective GCN (graph convolutional network) algorithm was suggested by Rex Ying et al. [18]. The developed algorithm was effective for data that combine graph convolutions and efficient random walks to produce embeddings incorporations.

A method for tweets recommendation was proposed by Arisara Pornwattanavichai et al. [19], which was based on hybrid recommendation with LDA for unsupervised topic modelling and GMF for supervised learning.

For gaining feedback on movies and movie genres in Rohan Nayak et al. [20] hybrid's framework, and based on their responses, the user will be classified and given a collection of recommendations.

Collaborative filtering, as previously discussed, is a well-known technique for making powerful recommendations based on ratings results. In order to enhance the technique's ability and achieve results by k-means clustering algorithm in movie recommendation framework, we continue our research.

1.5 MACHINE LEARNING TECHNIQUES & METHODOLOGY USED FOR RELIABILITY ASSESSMENT

The entire machine learning process is divided into several tasks. The first and foremost task is data set identification, and we have chosen MovieLens data set for our experimentation. From the well-known GroupLens Research Project at the University of Minnesota, we took MovieLens data [21]. Our goal with using this data set is to generate recommendations of movies to users on the basis of their interest and likings. This data set comprises 264505 ratings (1–5 scale) from 862 users on 2500 movies, and age, occupation, zip code, gender, etc., act as important demographic features taken from user data set. Next, data preprocessing is done to remove any sort of noise from the data set.

For our experimentation work, we are splitting the data set into two parts by 80:20, where the training part (80%) is used to train our model and then 20% is used

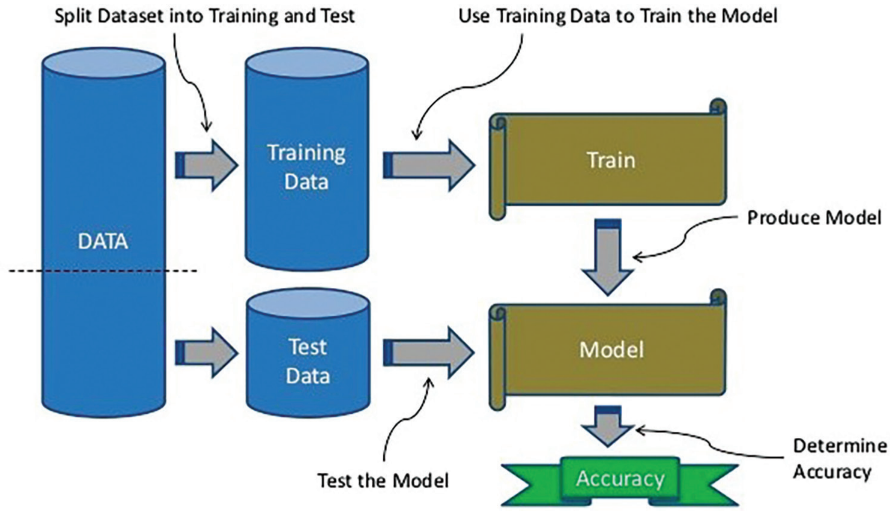


FIGURE 1.13 Machine learning process.

TABLE 1.1
Details of MovieLens Data set

Data set Name	Number of Unique Data
Movies.CSV	2500 Movies
Ratings.CSV	264505 Ratings
Users.CSV	862 Users

for testing. Finally, we also evaluate our model by calculating RMSE and MAE of our proposed model (Figure 1.13).

1.5.1 Data Set

We have taken MovieLens data set for our experimental work. This data set has been taken from (http://www.movieLens.org) for evaluating our proposed recommender system. Our experiments are performed on Google Colab where Google provides with the support of hardware on cloud to do our machine learning task. Here ratings by users are given on a scale from 1 to 5. Our data set is comprised of those users who have given at least 20 ratings. Our data set comprises 1,000,209 ratings given by users for different movies (Table 1.1).

1.5.2 Collaborative Filtering Technique

This approach is based on a user's suggestion of an object based on reactions from similar users. This works by selecting a smaller collection of users from a wide

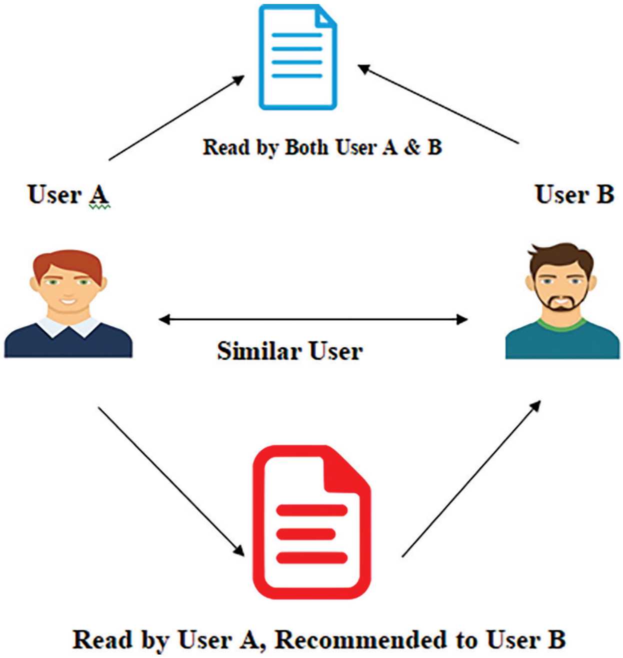


FIGURE 1.14 Collaborative filtering technique.

community of individuals with tastes close to a single user. In this, the main recommendation principle is that other users offer ratings to a specific object (Figure 1.14).

Measuring user similarity in collaborative filtering technique:

i. Pearson Correlation:

$$\sin(a,b) = \frac{\Sigma_{P\in P}(ra.p - ra)(rb.p - rb)}{\sqrt{\Sigma_{p\in P}(ra.p - ra)^2}\sqrt{\Sigma_{p\in P}(rb.p - rb)^2}} \tag{1.4}$$

where a and b are users, while $r_{a.p}$ is rating and P is set of items read by both users.

ii. **Cosine Similarity Measure**: It is measured by the angle between the vectors

$$\sin(\vec{p},\vec{q}) = \frac{\vec{p}\,.\vec{q}}{|\vec{p}|*|\vec{q}|} \tag{1.5}$$

U represents users having rated both items p and q.

1.6 EXPERIMENTAL SET-UP

The idea behind recommending movies to users based on item-item collaborative filtering comprises the steps discussed below:

Step 1. Create an adjusted rating for all movies by users. This adjusted rating is calculated by subtracting the movie's average rating from all users (for movie *j*) from each rating for that movie.

Step 2. Calculate similarity scores between all movies based on their adjusted movie ratings from each user (use cosine similarity). For recommendation purpose, we will only consider top similar movies to a target movie (top *n* nearest neighbours).

Step 3. For recommending a movie to a target user, we will score each movie, using the top *n* nearest neighbours for that movie. The score is basically a weighted rating based on the target user's rating for all movies they have rated and the similarity scores as the weight. Once we score all the movies, pick the top scoring movies from this scoring as recommendations.

The adjusted rating is nothing but the average rating for the movie from all users (*uj*) subtracted from all of the individual movie ratings (*ru*, *j*):

$$Ru, j = ru, j - uj$$

This adjusted rating is now comparable across all movies. This adjusted score basically compares the variation of ratings by a user from the movie's mean rating (Figures 1.15 and 1.16).

Now we create similarity score for each movie with every other movie; for this, we use the concept of cosine similarity (Table 1.2).

For creating recommendation to the target user, we find a score for each movie in the data set and movies with the highest score will be recommended to the user.

Steps involved in scoring are as follows:

1. Get the list of movies the target user has rated (seen movies). These seen movies will be used to create the score for all other movies (unseen movies) based on how the unseen movies are similar to these seen movies. These

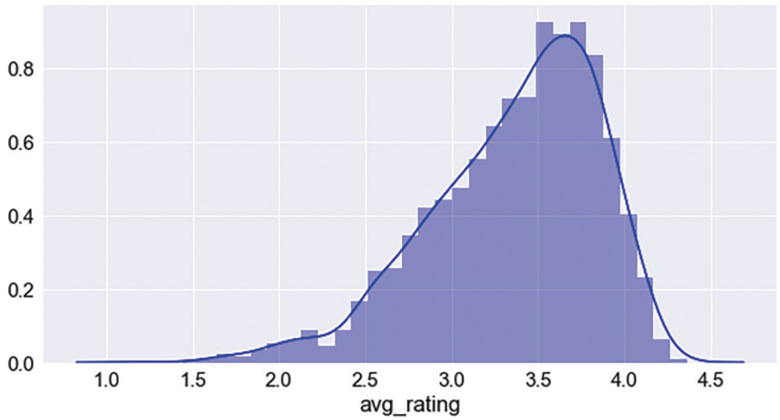


FIGURE 1.15 Potting average ratings across all users.

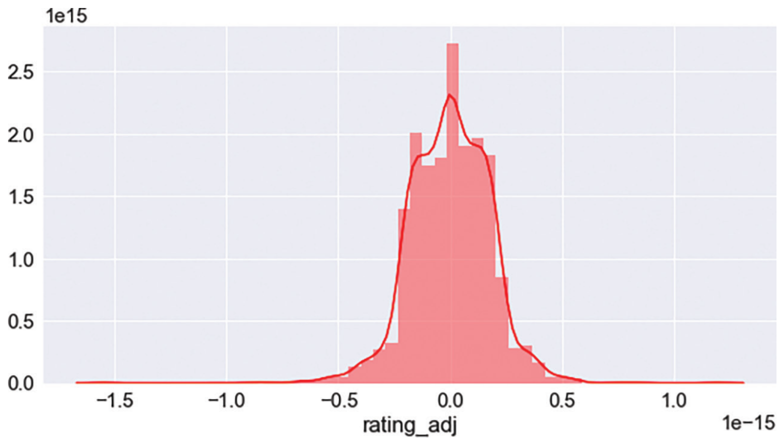


FIGURE 1.16 Potting average-adjusted ratings across all users.

seen movies tell us about the taste of the target user. If they have rated some movies high, we will try to find similar unseen movies to these high rated seen movies and recommend them to the user and vice versa for low rated movies.

2. For all the unseen movies in the data set, get the similarity scores between them and the seen movies. Here we can use all the seen movies or the top N neighbours out of the seen movies to get the similarity scores. We will use $N=30$ for our calculation. In case the number of seen movies is less than 30, we will use all the seen movies.
3. Using the similarity scores between each of the unseen movies and the seen movies, calculate a score for the unseen movies. The formula for the score is given below.
4. Once we get the score, sort the unseen movies based on the score and recommend the top n movies for the user.

We use the following formula to calculate score:

$$S_{u,i} = m_u + \frac{\sum_j \cos(i,j).(r_{uj} - m_j)}{\sum_j \cos(i,j)}$$

where

S is the score for the unseen movie i

m_u is the average rating for all seen movies by the target user U

$\cos(i, j)$ is the cosine similarity (based on adjusted rating) between the unseen movie i and the seen movie j

r_{uj} is the rating of the seen movie j by the target user U

m_j is the average rating from all users for the seen movie j

$r_{uj} - m_j$ is the same as the adjusted rating calculated above.

TABLE 1.2
Similarity Score of Movie vs Movie

MovieId	1	2	3	4	5	6	7	8	9
1	1.000000	0.213859	0.141760	−0.008966	0.097387	0.142986	0.098391	−0.002693	0.249048
2	0.213859	1.000000	0.218855	0.038701	0.125331	0.088945	0.154515	0.087974	0.231964
3	0.141760	0.218855	1.000000	0.056912	0.194855	0.067841	0.215001	0.084497	0.238945
4	−0.008966	0.038701	0.056912	1.000000	0.130774	0.014619	0.165135	0.008468	0.002328
5	0.097387	0.125331	0.194855	0.130774	1.000000	0.014217	0.135021	0.033035	0.070476

TABLE 1.3
Recommendations for User 76630

	MovieId	Title	Genres	Score
0	2906	Random Hearts (1999)	Drama|Romance	3.086117
1	1099	Christmas Carol, A (1938)	Children|Drama|Fantasy	3.060448
2	828	Adventures of Pinocchio, The (1996)	Adventure|Children	3.040377
3	611	Hellraiser, Bloodline (1996)	Action|Horror|Sci-Fi	3.018605
4	1015	Homeward Bound: The Incredible Journey (1993)	Adventure|Children|Drama	3.005596
5	334	Vanya on 42nd Street (1994)	Drama	2.985227
6	3684	Fabulous Baker Boys, The (1989)	Drama|Romance	2.978881
7	1014	Pollyanna (1960)	Children|Comedy|Drama	2.976269
8	1218	Killer, The (Die xue shuang xiong) (1989)	Action|Crime|Drama|Thriller	2.974656
9	2859	Stop Making Sense (1984)	Documentary|Musical	2.970456

TABLE 1.4
Splitting Data set into Training and Testing

Number of Users, Ratings and Movies	Training Data	Testing Data
Number of unique users in RATINGS data	681	181
Number of ratings in RATINGS data	209235	55270
Number of movies	2500	2496

Both the test and training data sets show similar distribution for the number of users per movie and average rating per movie (Tables 1.3 and 1.4). This shows that the test and training data sets are not that different and should be good enough for our evaluation. There is difference in the distribution of the average movie rating per user in test and training data sets, but these should be OK as we will use adjusted movie ratings for our recommendations (Figures 1.17–1.20).

1.6.1 Test Data Set – QUERY vs PROBE

Even from the given test data set, while trying to get the prediction for one user, we will only keep some movie ratings away from the model (QUERY movies), while we will pass on the remaining movies from that user to the model to be used as history (PROBE movies).

This division can be done randomly or on a temporal basis. We will do this based on time (temporal) – keep most recent ratings from a user as query and the older ones as probe. We can do this based on the timestamps available in the ratings data set.

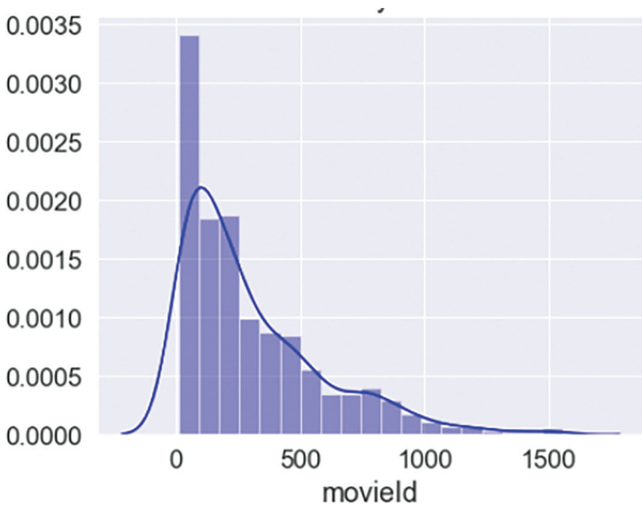


FIGURE 1.17 Movies rated by user in training data set.

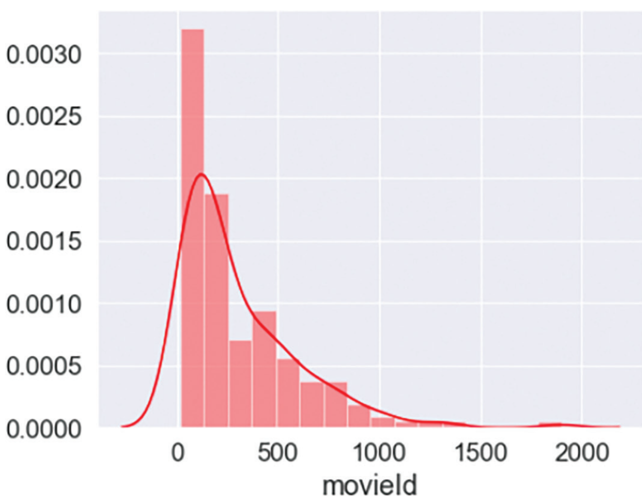


FIGURE 1.18 Movies rated by user in testing data set.

Algorithm 1. User-User Collaborative Filtering

The complete algorithm for user-user CBF will be explained in the following defined function. The steps for this algorithm are the following:

1. Create adjusted user movie rating.
2. Create similarity score for each user with every other user.
3. Create recommendation for the target user based on the similarity score.

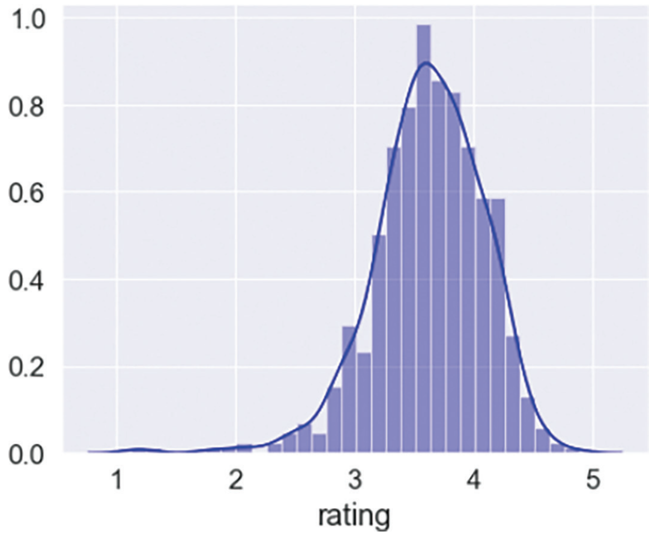


FIGURE 1.19 Average movie rating in training data set.

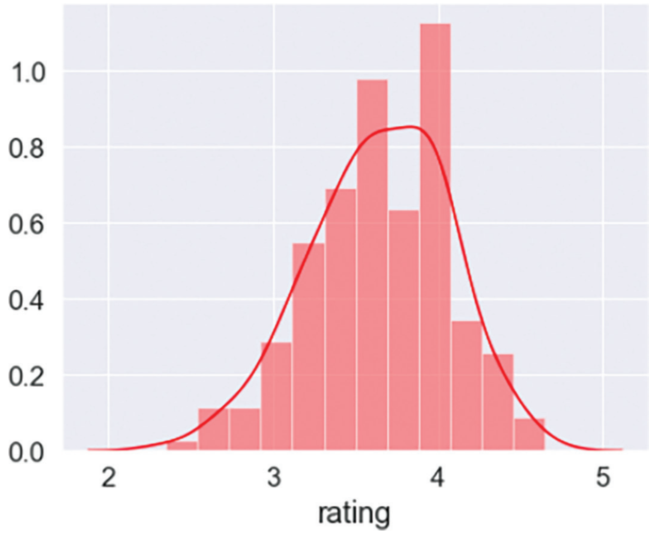


FIGURE 1.20 Average movie rating in testing data set.

Algorithm 2: Item-Item Collaborative Filtering

The steps for the item-item CBF will be as follows:

1. Create adjusted rating for every movie.
2. Get similarity scores between every movie.

3. Rank each movie for a given target user based on a score created using similarity scores between the movie and the top neighbours of the movies (which target user has rated).

1.7 RESULTS EVALUATION

1.7.1 Evaluate the Recommendation from Both Algorithms – RMSE and MAE

In our test query ratings data set, we loop through all the users and get the recommendation from both the algorithms. We will then use the predicted ratings for their movies and compare them with their actual rated movies to calculate the RMSE (root-mean-square error) and MAE (mean absolute error) metrics. The algorithm with the least RMSE or MAE will be considered better performing.

The graph below depicts a comparison between item-item CBF and user-user CBF with the number of neighbours with respect to RMSE (Figure 1.21).

The graph below depicts a comparison between item-item CBF and user-user CBF with the number of neighbours with respect to MAE (Figure 1.22).

From the above graph, it's pretty clear that the user-user algorithm gives much better prediction than the item-item algorithm. It also looks like that the neighbourhood size of ~20 is good enough in our case for user-user algorithm.

We are not choosing the neighbourhood size of 5 as it basically gives out very less number of recommendations and is not good enough.

The table below depicts the RMSE and MAE comparison table the two algorithms item-item CBF and user-user CBF (Table 1.5).

The graph given below depicts comparison between RMSE and MAE with respect to the two algorithms item-item CBF and user-user CBF (Figure 1.23).

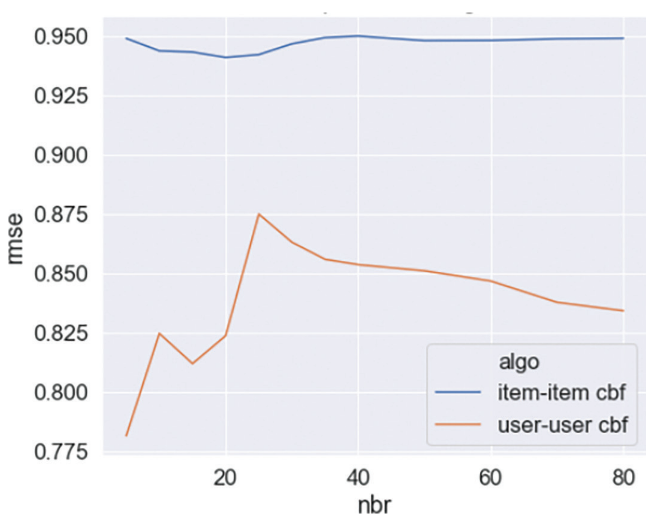


FIGURE 1.21 RMSE plot for algorithms.

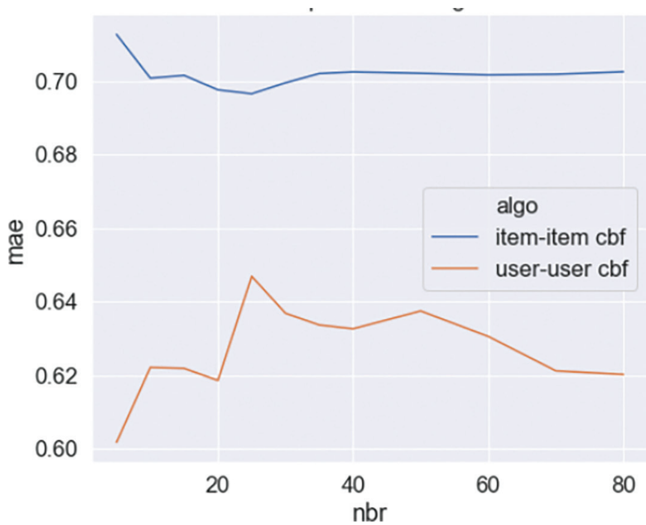


FIGURE 1.22 MAE plot for algorithms.

TABLE 1.5
RMSE and MAE Comparison Table

	Algorithm	NBR	Error_sq	MovieId	Error_abs	RMSE	MAE
0	Item-item CBF	5	489.039404	543	386.951160	0.949013	0.712617
1	Item-item CBF	10	483.644746	543	380.510503	0.943764	0.700756
2	Item-item CBF	15	483.149094	543	380.910301	0.943280	0.701492
3	Item-item CBF	20	480.815570	543	378.782391	0.940999	0.697573
4	Item-item CBF	25	482.042223	543	378.205743	0.942199	0.696511
12	User-user CBF	5	81.272629	133	80.028456	0.781711	0.601718
13	User-user CBF	10	188.447988	277	172.318323	0.824814	0.622088
14	User-user CBF	15	235.440337	357	221.988758	0.812094	0.621817
15	User-user CBF	20	274.206791	404	249.881260	0.823851	0.618518
16	User-user CBF	25	340.749089	445	287.846705	0.875059	0.646847

1.8 CONCLUSIONS

In the present chapter, techniques for establishing software reliability using machine learning have been used. On the basis of our experimental results, it is revealed that machine learning approach proves to be a better approach for predicting accurate software reliability. For analysing our model efficiency, we use the concept of RMSE, NRMSE and MAE criteria. On the basis of the experiment conducted on the well-known MovieLens data set, the ML approach gives better results and it is revealed that our technique provides more accurate results. The results obtained from our experimentation work reveals that the ML-based approach decreases testing cost by estimating the reliability of software and is much more feasible.

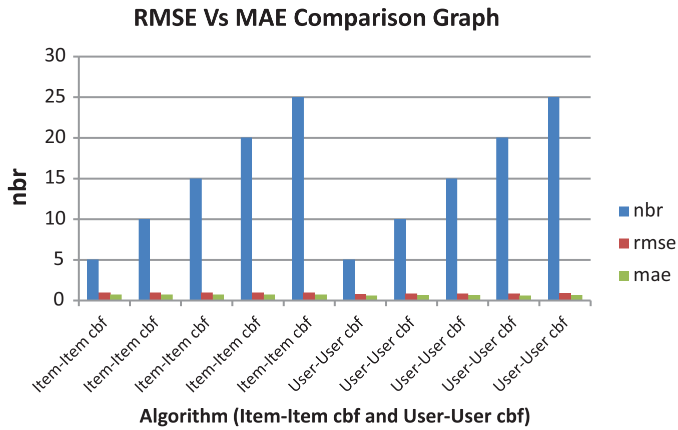


FIGURE 1.23 RMSE and MAE comparison graph for item-item CBF and user-user CBF w.r.t NBR.

REFERENCES

[1] Schick, G.J., Wolverton, R.W., 1978. An analysis of competing software reliability models, *IEEE Trans. Softw. Eng.* SE-4(2), 104–120.

[2] Yamada, S., 2014. *Software Reliability Modeling: Fundamentals and Applications*, Springer, Heidelberg. https://doi.org/10.1007/978-4-431-54565-1.

[3] Almering, V., van Genuchten, M., Cloudt, G., 2007. Using software reliability growth models in practice, *IEEE Comput. Soc.* 24, 82–88.

[4] Quadri, S.M., Ahmad, N., Farooq, S.U., 2011. Software reliability growth modeling with generalized exponential testing effort and optimal software release policy, *Glob. J. Comput. Sci. Technol.* 11(2), 27–42.

[5] Freedman, D.A., 2009. *Statistical Models: Theory and Practice*, Cambridge University Press.

[6] Edwards, J.R., 2007. Polynomial regression and response surface methodology. *Perspect. Organ. Fit*, pp. 361–372.

[7] Smola, A.J. and Schölkopf, B., 2004. A tutorial on support vector regression. *Stat. Comput.*, 14(3), 199–222.

[8] Lewis, R.J., 2000. An introduction to classification and regression tree (CART) analysis. In *Annual Meeting of the Society for Academic Emergency Medicine in San Francisco, California*, Vol. 14. Department of Emergency Medicine Harbor-UCLA Medical Center Torrance, California.

[9] Liaw, A. and Wiener, M., 2002. Classification and regression by random Forest, *R News*, 2(3), 18–22.

[10] Minh, L.Q., Duong, P.L.T. and Lee, M., 2018. Global sensitivity analysis and uncertainty quantification of crude distillation unit using surrogate model based on Gaussian process regression, *Ind. Eng. Chem. Res.*, 57(14), 5035–5044.

[11] Kabir, H.D., Khosravi, A., Hosen, M.A. and Nahavandi, S., 2018. Neural network-based uncertainty quantification: A survey of methodologies and applications, *IEEE Access*, 6, 36218–36234.

[12] Qiu, X., Zhang, L., Ren, Y., Suganthan, P.N. and Amaratunga, G., 2014. Ensemble deep learning for regression and time series forecasting. In *2014 IEEE Symposium on Computational Intelligence in Ensemble Learning (CIEL)* (pp. 1–6). IEEE.

[13] Katarya, R., 2018. Movie recommender system with metaheuristic artificial bee, *Neural Computing and Applications* 30(6), 1983–1990.

[14] Ponnam, L. T., Deepak Punyasamudram, S., Nallagulla, S. N., Yellamati, S., 2016. Movie recommender system using item based collaborative filtering technique. In *2016 International Conference on Emerging Trends in Engineering, Technology and Science (ICETETS)*. IEEE, Pudukkottai, India, pp. 1–5.

[15] Cami, B. R., Hassanpour, H., Mashayekhi, H., 2017. A content-based movie recommender system based on temporal user preferences. In *2017 3rd Iranian Conference on Intelligent Systems and Signal Processing (ICSPIS)*. IEEE, Shahrud, pp. 121–125.

[16] Reddy, S., Nalluri, S., Kunisetti, S., Ashok, S., Venkatesh, B., 2019. Content-based movie recommendation system using genre correlation. In *Smart Intelligent Computing and Applications. Smart Innovation, Systems and Technologies*. Springer, Singapore, pp. 391–397.

[17] Do, H.-Q., Le, T.-H., Yoon, B., Dynamic weighted hybrid recommender systems. In *2020 22nd International Conference on Advanced Communication Technology (ICACT)*. pp. 644–650.

[18] Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W. L., Leskovec, J., 2018. Graph convolutional neural networks FORWEB-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. KDD '18*. Association for Computing Machinery, London, UK, pp. 974–983.

[19] Pornwattanavichai, A., Sakolnagara, P. B. N., Jirachanchaisiri, P., Kitsupapaisan, J., Maneeroj, S. Enhanced tweet hybrid recommender system using unsupervised topic modeling and matrix factorization-based neural network. In *Supervised and Unsupervised Learning for Data Science*. Springer International Publishing, pp. 121–143.

[20] Nayak, R., Mirajkar, A., Rokade, J., Wadhwa, G., 2018. Hybrid recommendation system for movies, *Int. Res. J. Eng. Tech.* 05(03), 4.

[21] Cabannes, V., Rudi, A., Bach, F., 2021. Disambiguation of weak supervision with exponential convergence rates. *Proceedings of the 38th International Conference on Machine Learning*, ICML, arXiv:2102.02789.

Enhancing Software Reliability by Evaluating Prediction Accuracy of CBF Algorithm Using Machine Learning

Schick, G.J., Wolverton, R.W. , 1978. An analysis of competing software reliability models, IEEE Trans. Softw. Eng. SE-4(2), 104–120.

Yamada, S. , 2014. Software Reliability Modeling: Fundamentals and Applications, Springer, Heidelberg. https://doi.org/10.1007/978-4-431-54565-1.

Almering, V., van Genuchten, M., Cloudt, G. , 2007. Using software reliability growth models in practice, IEEE Comput. Soc. 24, 82–88.

Quadri, S.M. , Ahmad, N. , Farooq, S.U. , 2011. Software reliability growth modeling with generalized exponential testing effort and optimal software release policy, Glob. J. Comput. Sci. Technol. 11(2), 27–42.

Freedman, D.A. , 2009. Statistical Models: Theory and Practice, Cambridge University Press.

Edwards, J.R. , 2007. Polynomial regression and response surface methodology. Perspect. Organ. Fit, pp. 361–372.

Smola, A.J. and Schölkopf, B. , 2004. A tutorial on support vector regression. Stat. Comput., 14(3), 199–222.

Lewis, R.J. , 2000. An introduction to classification and regression tree (CART) analysis. In Annual Meeting of the Society for Academic Emergency Medicine in San Francisco, California, Vol. 14. Department of Emergency Medicine Harbor-UCLA Medical Center Torrance, California.

Liaw, A. and Wiener, M. , 2002. Classification and regression by random Forest, R News, 2(3), 18–22.

Minh, L.Q., Duong, P.L.T. and Lee, M. , 2018. Global sensitivity analysis and uncertainty quantification of crude distillation unit using surrogate model based on Gaussian process regression, Ind. Eng. Chem. Res., 57(14), 5035–5044.

Kabir, H.D., Khosravi, A., Hosen, M.A. and Nahavandi, S. , 2018. Neural network-based uncertainty quantification: A survey of methodologies and applications, IEEE Access, 6, 36218–36234.

Qiu, X., Zhang, L., Ren, Y., Suganthan, P.N. and Amaratunga, G. , 2014. Ensemble deep learning for regression and time series forecasting. In 2014 IEEE Symposium on Computational Intelligence in Ensemble Learning (CIEL) (pp. 1–6). IEEE.

Katarya, R. , 2018. Movie recommender system with metaheuristic artificial bee, Neural Computing and Applications 30(6), 1983–1990.

Ponnam, L. T., Deepak Punyasamudram, S., Nallagulla, S. N., Yellamati, S. , 2016. Movie recommender system using item based collaborative filtering technique. In 2016 International Conference on Emerging Trends in Engineering, Technology and Science (ICETETS). IEEE, Pudukkottai, India, pp. 1–5.

Cami, B. R., Hassanpour, H., Mashayekhi, H. , 2017. A content-based movie recommender system based on temporal user preferences. In 2017 3rd Iranian Conference on Intelligent Systems and Signal Processing (ICSPIS). IEEE, Shahrud, pp. 121–125.

Reddy, S., Nalluri, S., Kunisetti, S., Ashok, S., Venkatesh, B. , 2019. Content-based movie recommendation system using genre correlation. In Smart Intelligent Computing and Applications. Smart Innovation, Systems and Technologies. Springer, Singapore, pp. 391–397.

Do, H.-Q., Le, T.-H., Yoon, B. , Dynamic weighted hybrid recommender systems. In 2020 22nd International Conference on Advanced Communication Technology (ICACT). pp. 644–650.

Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W. L., Leskovec, J. , 2018. Graph convolutional neural networks FORWEB-scale recommender systems. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. KDD ’18. Association for Computing Machinery, London, UK, pp. 974–983.

Pornwattanavichai, A., Sakolnagara, P. B. N. , Jirachanchaisiri, P. , Kitsupapaisan, J. , Maneeroj, S. Enhanced tweet hybrid recommender system using unsupervised topic modeling and matrix factorization-based neural network. In Supervised and Unsupervised Learning for Data Science. Springer International Publishing, pp. 121–143.

Nayak, R., Mirajkar, A., Rokade, J., Wadhwa, G. , 2018. Hybrid recommendation system for movies, Int. Res. J. Eng. Tech. 05(03), 4.

Cabannes, V., Rudi, A., Bach, F. , 2021. Disambiguation of weak supervision with exponential convergence rates. Proceedings of the 38th International Conference on Machine Learning, ICML, arXiv:2102.02789.

Significance of Machine Learning and Deep Learning in Development of Artificial Intelligence

Dargan, S., Kumar, M., Ayyagari, M. R. , & Kumar, G. (2019). A survey of deep learning and its applications: A new paradigm to machine learning. Archives of Computational Methods in Engineering, 1–22.

Wlodarczak, P., Soar, J. , & Ally, M. (2015). Multimedia data mining using deep learning. In 2015 Fifth International Conference on Digital Information Processing and Communications (ICDIPC) (pp. 190–196). IEEE.

Nguyen, G. , Dlugolinsky, S., Bobák, M. , Tran, V. , García, Á. L. , Heredia, I. , et al. (2019). Machine learning and deep learning frameworks and libraries for large-scale data mining: a survey. Artificial Intelligence Review, 52(1), 77–124.

Fok, W. W., He, Y. S., Yeung, H. A., Law, K. Y., Cheung, K. H., Ai, Y. Y. , & Ho, P. (2018). Prediction model for students' future development by deep learning and tensorflow artificial intelligence engine. In 2018 4th International Conference on Information Management (ICIM) (pp. 103–106). IEEE.

Lin, Y. L., Chen, T. Y. , & Yu, L. C. (2017). Using machine learning to assist crime prevention. In 2017 6th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI) (pp. 1029–1030). IEEE.

Wawrzyniak, Z. M., Jankowski, S., Szczechla, E., Szymański, Z., Pytlak, R., Michalak, P. , & Borowik, G. (2018). Data-driven models in machine learning for crime prediction. In 2018 26th International Conference on Systems Engineering (ICSEng) (pp. 1–8). IEEE.

Navalgund, U. V. , & Priyadharshini, K. (2018). Crime intention detection system using deep learning. In 2018 International Conference on Circuits and Systems in Digital Enterprise Technology (ICCSDET) (pp. 1–6). IEEE.

Zhong, W., Yu, N. , & Ai, C. (2020). Applying big data based deep learning system to intrusion detection. Big Data Mining and Analytics, 3(3), 181–195.

Aksu, D. , & Aydin, M. A. (2018). Detecting port scan attempts with comparative analysis of deep learning and support vector machine algorithms. In 2018 International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT) (pp. 77–80). IEEE.

Mohamad, M., Selamat, A. , & Salleh, K. A. (2019). An analysis on deep learning approach performance in classifying big data set. In 2019 1st International Conference on Artificial Intelligence and Data Sciences (AiDAS) (pp. 35–39). IEEE.

Aristodemou, L. , & Tietze, F. (2018). The state-of-the-art on Intellectual Property Analytics (IPA): A literature review on artificial intelligence, machine learning and deep learning methods for analysing intellectual property (IP) data. World Patent Information, 55, 37–51.

Gupta, S., Mohanta, S., Chakraborty, M. , & Ghosh, S. (2017). Quantum machine learning-using quantum computation in artificial intelligence and deep neural networks: Quantum computation and machine learning in artificial intelligence. In 2017 8th Annual Industrial Automation and Electromechanical Engineering Conference (IEMECON) (pp. 268–274). IEEE.

Rani, K. S., Kumari, M., Singh, V. B. , & Sharma, M. (2019). Deep learning with big data: An emerging trend. In 2019 19th International Conference on Computational Science and Its Applications (ICCSA) (pp. 93–101). IEEE.

Sari, C. T. , & Gunduz-Demir, C. (2018). Unsupervised feature extraction via deep learning for histopathological classification of colon tissue images. IEEE Transactions on Medical Imaging, 38(5), 1139–1149.

wei Tan, J. , Chang, S. W. , Abdul-Kareem, S. , Yap, H. J. , & Yong, K. T. (2018). Deep learning for plant species classification using leaf vein morphometric. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 17(1), 82–90.

Chandran, P. S., Byju, N. B., Deepak, R. U., Nishakumari, K. N., Devanand, P. , & Sasi, P. M. (2018). Missing child identification system using deep learning and multiclass SVM. In 2018 IEEE Recent Advances in Intelligent Computational Systems (RAICS) (pp. 113–116). IEEE.

Dong, Y. N. , & Liang, G. S. (2019). Research and discussion on image recognition and classification algorithm based on deep learning. In 2019 International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI) (pp. 274–278). IEEE.

Sufri, N. A. J., Rahmad, N. A., Ghazali, N. F., Shahar, N. , & As'Ari, M. A. (2019). Vision based system for banknote recognition using different machine learning and deep learning approach. In 2019 IEEE 10th Control and System Graduate Research Colloquium (ICSGRC) (pp. 5–8). IEEE.

Dong, Y., Zhang, Q., Qiao, Z. , & Yang, J. J. (2017). Classification of cataract fundus image based on deep learning. In 2017 IEEE International Conference on Imaging Systems and Techniques (IST) (pp. 1–5). IEEE.

Huixian, J. (2020). The analysis of plants image recognition based on deep learning and artificial neural network. IEEE Access, 8, 68828–68841.

Mete, B. R. , & Ensari, T. (2019, October). Flower classification with deep CNN and machine learning algorithms. In 2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT) (pp. 1–5). IEEE.

Zawbaa, H. M., Hazman, M., Abbass, M. , & Hassanien, A. E. (2014). Automatic fruit classification using random forest algorithm. In 2014 14th International Conference on Hybrid Intelligent Systems (pp. 164–168). IEEE.

Sewak, M., Sahay, S. K. , & Rathore, H. (2018). Comparison of deep learning and the classical machine learning algorithm for the malware detection. In 2018 19th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD) (pp. 293–296). IEEE.

Badriyah, T., Sakinah, N., Syarif, I. , & Syarif, D. R. (2020). Machine learning algorithm for stroke disease classification. In 2020 International Conference on Electrical, Communication, and Computer Engineering (ICECCE) (pp. 1–5). IEEE.

Lee, K. B. , & Shin, H. S. (2019). An application of a deep learning algorithm for automatic detection of unexpected accidents under bad CCTV monitoring conditions in tunnels. In 2019 International Conference on Deep Learning and Machine Learning in Emerging Applications (Deep-ML) (pp. 7–11). IEEE.

Kavitha, D., Hebbar, R., Vinod, P. V., Harsheetha, M. P., Jyothi, L. , & Madhu, S. H. (2018). CNN based technique for systematic classification of field photographs. In 2018 International Conference on Design Innovations for 3Cs Compute Communicate Control (ICDI3C) (pp. 59–63). IEEE.

Li, Q., Cai, W., Wang, X. , Zhou, Y., Feng, D. D. , & Chen, M. (2014). Medical image classification with convolutional neural network. In 2014 13th International Conference on Control Automation Robotics & Vision (ICARCV) (pp. 844–848). IEEE.

Akila, D. , Pal, S., Jayakarthik, R., Chattopadhyay, S. , & Obaid, A. J. (2021). Deep learning enhancing performance using support vector machine HTM cortical learning algorithm. Journal of Physics: Conference Series, 1963, 1, 012144.

Suseendran, G. , et al. (2021). Deep learning frequent pattern mining on static semi structured data streams for improving fast speed and complex data streams. In 2021 7th International Conference on Optimization and Applications (ICOA). IEEE.

S. Lavanya ; D. Akila , Prediction performance of crime against women using rule based decision tree J48 classification algorithms in various states of India. IEEE Xplore, 139–144.

Implication of Soft Computing and Machine Learning Method for Software Quality, Defect and Model Prediction

Ahmed, M. R. , Zamal, M. F. Bin Ali, M.A. , Shamrat, F. M. J. M. , & Ahmed, N. (2020). The impact of software fault prediction in real-world application: An automated approach for software engineering. ACM International Conference Proceeding Series, 247–251. https://doi.org/10.1145/3379247.3379278.

Amershi, S., Begel, A., Bird, C., DeLine, R., Gall, H., Kamar, E., Nagappan, N., Nushi, B. , & Zimmermann, T. (2019). Software engineering for Machine Learning applications. ICSE, 2020, 1–10. https://fontysblogt.nl/software-engineering-for-machine-learning-applications/.

Arora, I., Tetarwal, V. , & Saha, A. (2015). Open issues in software defect prediction. Procedia Computer Science, 46 (ICICT 2014), 906–912. https://doi.org/10.1016/j.procs.2015.02.161.

Azar, D. , & Vybihal, J. (2011). An ant colony optimization algorithm to improve software quality prediction models: Case of class stability. April. https://doi.org/10.1016/j.infsof.2010.11.013.

Bhavsar, K., Shah, V. , & Gopalan, S. (2020). Scrumbanfall: An agile integration of scrum and kanban with waterfall in software engineering. International Journal of Innovative Technology and Exploring Engineering, 9(4), 2075–2084.https://doi.org/10.35940/ijitee.d1437.029420.

Challagulla, V. U. B., Bastani, F. B., Yen, I. L. , & Paul, R. A. (2005). Empirical assessment of Machine Learning based software defect prediction techniques. Proceedings – International Workshop on Object-Oriented Real-Time Dependable Systems, 263–270.https://doi.org/10.1109/WORDS.2005.32.

Chigurupati, T. R. M., Basha, S. K. M. , & Sunil, E. (2020). Constructive heart disease prophecy with hybrid Machine Learning strategy. International Journal of Scientific Research in Computer Science Applications and Management Studies, 9(3).

Chug, A. , & Malhotra, R. (2016). Benchmarking framework for maintainability prediction of open source software using object oriented metrics. International Journal of Innovative Computing, Information and Control, 12(2), 615–634.

Dam, H. K., Pham, T., Ng, S. W., Tran, T., Grundy, J., Ghose, A., Kim, T. , & Kim, C.-J. (2018). A deep tree-based model for software defect prediction. Software Engineering. http://arxiv.org/abs/1802.00921.

Feldt, R., De Oliveira Neto, F. G. , & Torkar, R. (2018). Ways of applying artificial intelligence in software engineering. Proceedings – International Conference on Software Engineering, 35–41. https://doi.org/10.1145/3194104.3194109.

Fenton, N. E., Society, I. C., Neil, M. , & Society, I. C. (1999). Defects_Prediction_Preprint105579, 25(3), 1–15. http://www.eecs.qmul.ac.uk/~norman/papers/defects_prediction_preprint105579.pdf.

Gray, D., Bowes, D., Davey, N., Sun, Y. , & Christianson, B. (2011). The misuse of the NASA Metrics Data Program data sets for automated software defect prediction. IET Seminar Digest, 2011(1), 96–103. https://doi.org/10.1049/ic.2011.0012.

Guo, P. , & Lyu, M. R. (2000). Software quality prediction using mixture models with EM algorithm. Proceedings – 1st Asia-Pacific Conference on Quality Software, APAQS, 2000, 69–78. https://doi.org/10.1109/APAQ.2000.883780.

Hammouri, A., Hammad, M., Alnabhan, M. , & Alsarayrah, F. (2018). Software Bug Prediction using Machine Learning approach. International Journal of Advanced Computer Science and Applications, 9(2), 78–83. https://doi.org/10.14569/IJACSA.2018.090212.

Hanselmann, G. , & Sarishvili, A. (2007). Heterogeneous redundancy in software quality prediction using a hybrid Bayesian approach. Berichte des Fraunhofer ITWM 125.

Harman, M. (2007). The current state and future of search based software engineering. FoSE 2007: Future of Software Engineering, 342–357. https://doi.org/10.1109/FOSE.2007.29

He, P., Li, B., Liu, X., Chen, J. , & Ma, Y. (2015). An empirical study on software defect prediction with a simplified metric set. Information and Software Technology, 59, 170–190. https://doi.org/10.1016/j.infsof.2014.11.006.

Kapur, R. , & Sodhi, B. (2019). Towards a knowledge warehouse and expert system for the automation of SDLC tasks. Proceedings – 2019 IEEE/ACM International Conference on Software and System Processes, ICSSP 2019, May, 5–8. https://doi.org/10.1109/ICSSP.2019.00011.

Kim, M. (2020). Software engineering for data analytics. IEEE Software, 37(4), 36–42. https://doi.org/10.1109/MS.2020.2985775.

Laradji, I. H., Alshayeb, M. , & Ghouti, L. (2015). Software defect prediction using ensemble learning on selected features. Information and Software Technology, 58(July), 388–402. https://doi.org/10.1016/j.infsof.2014.07.005.

Li, M., Zhang, H., Wu, R. , & Zhou, Z. H. (2012). Sample-based software defect prediction with active and semi-supervised learning. Automated Software Engineering, 19(2), 201–230. https://doi.org/10.1007/s10515-011-0092-1.

Lincke, R., Gutzmann, T. , & Löwe, W. (2010). Software quality prediction models compared. Proceedings – International Conference on Quality Software, 82–91.https://doi.org/10.1109/QSIC.2010.9.

Ma, Y., Luo, G., Zeng, X. , & Chen, A. (2012). Transfer learning for cross-company software defect prediction. Information and Software Technology, 54(3), 248–256. https://doi.org/10.1016/j.infsof.2011.09.007.

Nam, J. (2014). Survey on Software Defect Prediction.Master Thesis. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.722.3147.

Nascimento, E., Nguyen-Duc, A., Sundbø, I. , & Conte, T. (2020). Software engineering for artificial intelligence and Machine Learning software: a systematic literature review. Software Engineering. http://arxiv.org/abs/2011.03751.

Okutan, A. , & Yıldız, O. T. (2014). Software defect prediction using Bayesian networks. Empirical Software Engineering, 19(1), 154–181. https://doi.org/10.1007/s10664-012-9218-8.

Pelayo, L. , & Dick, S. (2007). Applying novel resampling strategies to software defect prediction. Annual Conference of the North American Fuzzy Information Processing Society – NAFIPS, 69–72. https://doi.org/10.1109/NAFIPS.2007.383813.

Pradhan, S., Nanniyur, V. , & Vissapragada, P. K. (2020). On the defect prediction for large scale software systems-from defect density to Machine Learning. Proceedings – 2020 IEEE 20th International Conference on Software Quality, Reliability, and Security, QRS, 2020, 374–381. https://doi.org/10.1109/QRS51102.2020.00056.

Rahman, M. S., Rivera, E., Khomh, F., Guéhéneuc, Y.-G. , & Lehnert, B. (2019). Machine Learning software engineering in practice: an industrial case study. Software Engineering, 1–21. http://arxiv.org/abs/1906.07154.

Rana, R. , & Staron, M. (n.d.). Machine Learning approach for quality assessment and prediction in large software organizations. 2015 6th IEEE International Conference on Software Engineering and Service Science (ICSESS). https://doi.org/10.1109/ICSESS.2015.7339243.

Rashid, E. (2012). A survey in the area of Machine Learning and its application for software quality prediction. ACM SIGSOFT Software Engineering Notes, 37(5), 1–7. https://doi.org/10.1145/2347696.2347709.

Rawat, M. S. , & Dubey, S. K. (2012). Software defect prediction models for quality improvement: A literature study. International Journal of Computer Science Issues, 9(5), 288–296.

Rodriguez, D., Herraiz, I., Harrison, R., Dolado, J. , & Riquelme, J. C. (2014). Preliminary comparison of techniques for dealing with imbalance in software defect prediction. ACM International Conference Proceeding Series. https://doi.org/10.1145/2601248.2601294.

Serban, A., Van Der Blom, K., Hoos, H. , & Visser, J. (2020). Adoption and effects of software engineering best practices in Machine Learning. International Symposium on Empirical Software Engineering and Measurement.https://doi.org/10.1145/3382494.3410681.

Sinha, A., Singh, S. , & Kashyap, D. (2020). Implication of data mining and Machine Learning in software engineering domain for software model, quality and defect prediction. Journal of Emerging Technologies and Innovative Research (JETIR), 7(10).

Sun, Z., Song, Q. , & Zhu, X. (2012). Using coding-based ensemble learning to improve software defect prediction. IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews, 42(6), 1806–1817. https://doi.org/10.1109/TSMCC.2012.2226152.

Tantithamthavorn, C., McIntosh, S., Hassan, A. E. , & Matsumoto, K. (2016). Comments on researcher bias: The use of Machine Learning in software defect prediction. IEEE Transactions on Software Engineering, 42(11), 1092–1094. https://doi.org/10.1109/TSE.2016.2553030.

Tejaswini, P. L. S., Varsha, K. S., Yasaswini, P. , & Yalamanchili, S. (2019). Software defect prediction using Machine Learning techniques.2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184), 2, 1053–1057.

Wahono, R. S. (2007). A systematic literature review of software defect prediction: Research trends, datasets, methods and frameworks.Journal of Software Engineering, 1(1), 1–16. https://doi.org/10.3923/jse.2007.1.12.

Wang, Q., Zhu, J. , & Yu, B. (2007). Feature selection and clustering in software quality prediction. EASE'07 Proceedings of the 11th International Conference on Evaluation and Assessment in Software Engineering, 21–32. http://www.bcs.org/upload/pdf/ewic_ea07_paper3.pdf.

Wang, S. , & Yao, X. (2013). Using class imbalance learning for software defect prediction.IEEE Transactions on Reliability, 62(2), 434–443. https://doi.org/10.1109/TR.2013.2259203.

Washizaki, H., Uchida, H., Khomh, F. , & Guéhéneuc, Y. G. (2019). Studying software engineering patterns for designing Machine Learning systems. Proceedings – 2019 10th International Workshop on Empirical Software Engineering in Practice, IWESEP 2019, 49–54.https://doi.org/10.1109/IWESEP49350.2019.00017.

Xing, F., Guo, P. , & Lyu, M. R. (n.d.). A novel method for early software quality prediction based on support vector machine. 16th IEEE International Symposium on Software Reliability Engineering (ISSRE'05). https://doi.org/10.1109/ISSRE.2005.6.

Zanutto, D., Lorenzini, E. C., Mantellato, R., Colombatti, G. , & Sanchez-Torres, A. (2012). Orbital debris mitigation through deorbiting with passive electrodynamic drag. Proceedings of the International Astronautical Congress, IAC, 4, 2577–2585.

Zheng, J. (2010). Cost-sensitive boosting neural networks for software defect prediction. Expert Systems with Applications, 37(6), 4537–4543. https://doi.org/10.1016/j.eswa.2009.12.056.

Ambiguity Based on Working and Functionality in Deployed Software from Client Side in Prototype SDLC Model Scenario

Abduljalil, S. , & Kang, D.-K. (2011). Effective Model and Methods for Analysing Human Factors in Software Design for Efficient User Experience. Journal of Information and Communication Convergence Engineering, 9(1), 100–104. https://doi.org/10.6109/jicce.2011.9.1.100.

Alferoff, C. , & Knights, D. (2008). Customer relationship management in call centers: The uneasy process of re(form)ing the subject through the "people-by-numbers" approach. Information and Organization, 18(1), 29–50.https://doi.org/10.1016/j.infoandorg.2007.10.002.

Alokla, M., Alkhateeb, M., Abbad, M. , & Jaber, F. (2019). Customer relationship management: A review and classification. Transnational Marketing Journal, 7(2), 187–210. https://doi.org/10.33182/tmj.v7i2.734.

Alshazly, A. A., Elfatatry, A. M. , & Abougabal, M. S. (2014). Detecting defects in software requirements specification. Alexandria Engineering Journal, 53(3), 513–527. https://doi.org/10.1016/j.aej.2014.06.001.

Baars, W. (2006). Project management handbook DANS (Data Archiving and Networked Services). Proceedings – 2014 Agile Conference, AGILE 2014, July, 84. http://localhost:8080/ccsdsdocs/LIFECYCLE-ProjectManagement-DANS-en.pdf.

Bano, M. (2016). Addressing the challenges of requirements ambiguity: A review of empirical literature. 5th International Workshop on Empirical Requirements Engineering, EmpiRE 2015 – Proceedings, July, 21–24. https://doi.org/10.1109/EmpiRE.2015.7431303.

Bäumer, F. S. , & Geierhos, M. (2018). Flexible ambiguity resolution and incompleteness detection in requirements descriptions via an indicator-based configuration of text analysis pipelines. Proceedings of the 51st Hawaii International Conference on System Sciences, 9, 5746–5755. https://doi.org/10.24251/hicss.2018.720.

Chen, I. J. , & Popovich, K. (2003). Understanding customer relationship management (CRM): People, process and technology. Business Process Management Journal, 9(5), 672–688. https://doi.org/10.1108/14637150310496758.

Dalpiaz, F., van der Schalk, I., Brinkkemper, S., Aydemir, F. B. , & Lucassen, G. (2019). Detecting terminological ambiguity in user stories: Tool and experimentation. Information and Software Technology, 110(December), 3–16.https://doi.org/10.1016/j.infsof.2018.12.007.

Ezzini, S., Abualhaija, S., Arora, C., Sabetzadeh, M. , & Briand, L. C. (2021). Using domain-specific corpora for improved handling of ambiguity in requirements. 2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE). 1485–1497. https://doi.org/10.1109/icse43902.2021.00133.

Fantechi, A., Ferrari, A., Gnesi, S. , & Semini, L. (2018). Hacking an ambiguity detection tool to extract variation points: An experience report. ACM International Conference Proceeding Series, 43–50. https://doi.org/10.1145/3168365.3168381.

Ferrari, A., Lipari, G., Gnesi, S. , & Spagnolo, G. O. (2014). Pragmatic ambiguity detection in natural language requirements. 2014 IEEE 1st International Workshop on Artificial Intelligence for Requirements Engineering, AIRE 2014 – Proceedings, August, 1–8. https://doi.org/10.1109/AIRE.2014.6894849.

Ganpatrao Sabale, R. (2012). Comparative study of prototype model for software engineering with system development life cycle. IOSR Journal of Engineering, 02(07), 21–24. https://doi.org/10.9790/3021-02722124.

Gervasi, V., Ferrari, A., Zowghi, D. , & Spoletini, P. (2019). Ambiguity in requirements engineering: Towards a unifying framework. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 11865, 191–210. https://doi.org/10.1007/978-3-030-30985-5_12.

Grieskamp, W., Heisel, M. , & Dörr, H. (1998). Specifying embedded systems with statecharts and Z: An agenda for cyclic software components. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 1382, 88–106. https://doi.org/10.1007/bfb0053585.

Gupta, A. K., Deraman, A. , & Siddiqui, S. T. (2019). A survey of software requirements specification ambiguity. ARPN Journal of Engineering and Applied Sciences, 14(17), 3046–3061.
Hammer, J. H. , & Vogel, D. L. (2013). Assessing the utility of the willingness/prototype model in predicting help-seeking decisions. Journal of Counseling Psychology, 60(1), 83–97. https://doi.org/10.1037/a0030449.
Haron, H. , & Ghani, A. A. A. (2015). A survey on ambiguity awareness towards malay system requirement specification (SRS) among industrial IT practitioners. Procedia Computer Science, 72, 261–268. https://doi.org/10.1016/j.procs.2015.12.139.
Hayman Oo, K., Nordin, A., Ritahani Ismail, A. , & Sulaiman, S. (2018). An analysis of ambiguity detection techniques for software requirements specification (SRS). International Journal of Engineering & Technology, 7(2.29), 501. https://doi.org/10.14419/ijet.v7i2.29.13808.
Husain, M. S. , & Beg, M. R. (2015). Advances in ambiguity less NL SRS : A review. 2015 IEEE International Conference on Engineering and Technology (ICETECH) March 2015, 221–225.
Kamsties, E., Berry, D. M. , & Paech, B. (2001). Detecting ambiguities in requirements documents using inspections. Engineering, July, 1–12. http://publica.fraunhofer.de/documents/N-6838.html%5Cnhttp://www.cas.mcmaster.ca/wise/wise01/KamstiesBerryPaech.pdf%5Cnhttp://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.145.6497&rep=rep1&type=pdf.
Kim, J., Suh, E. , & Hwang, H. (2003). A model for evaluating the effectiveness of CRM using the balanced scorecard. Journal of Interactive Marketing, 17(2), 5–19. https://doi.org/10.1002/dir.10051.
Massey, A. K., Rutledge, R. L., Antón, A. I. , & Swire, P. P. (2014). Identifying and classifying ambiguity for regulatory requirements. 2014 IEEE 22nd International Requirements Engineering Conference, RE 2014 – Proceedings, 115(111), 83–92. https://doi.org/10.1109/RE.2014.6912250.
Matsumoto, Y., Shirai, S. , & Ohnishi, A. (2017). A method for verifying non-functional requirements. Procedia Computer Science, 112, 157–166.https://doi.org/10.1016/j.procs.2017.08.006.
Mazza, C. (1989). Software project management. Computer Physics Communications, 57(1–3), 23–28. https://doi.org/10.1016/0010-4655(89)90188-4.
Mich, L. , & Garigliano, R. (2000). Ambiguity measures in requirements engineering. ICS2000 International Conference on Software: Theory and Practice. 16th IFIP World Computer Congress, 39–48.
Nacheva, R. (2017). Prototyping approach in user interface. 2nd Conference on Innovative Teaching Methods, June, 80–87. https://www.researchgate.net/publication/317414969.
Nigam, A., Arya, N., Nigam, B. , & Jain, D. (2012). Tool for automatic discovery of ambiguity in requirements. International Journal of Computer Science Issues (IJCSI), 9(5), 350–356. www.IJCSI.org.
Osama, S. , & Aref, M. (2018). Detecting and Resolving Ambiguity Approach in Requirement Specification: Implementation, Results and Evaluation. International Journal of Intelligent Computing and Information Sciences, 18(1), 27–36. https://doi.org/10.21608/ijicis.2018.15909
Osman, M. H. , & Zaharin, M. F. (2018). Ambiguous software requirement specification detection: An automated approach. Proceedings - International Conference on Software Engineering, 33–40. https://doi.org/10.1145/3195538.3195545
Pittke, F., Leopold, H. , & Mendling, J. (2016). Automatic detection and resolution of lexical ambiguity in process models (extended abstract). Lecture Notes in Informatics (LNI), Proceedings - Series of the Gesellschaft Fur Informatik (GI), P252, 75–76.
Ragunath, P., Velmourougan, S., Davachelvan, P., Kayalvizhi, S. , & Ravimohan, R. (2010). Evolving a new model (SDLC Model-2010) for software development life cycle (SDLC). International Journal of Computer Science and Network Security, 10(1), 112–119.
Rauterberg, M., Strohm, O. , & Kirsch, C. (1995). Benefits of user-oriented software development based on an iterative cyclic process model for simultaneous engineering. International Journal of Industrial Ergonomics, 16(4–6), 391–409. https://doi.org/10.1016/0169-8141(95)00021-8.
Sabriye, A. O. J. , & Wan Zainon, W. M. N. (2018). An approach for detecting syntax and syntactic ambiguity in software requirement specification. Journal of Theoretical and Applied Information Technology, 96(8), 2275–2284.

Sabriye, A. O. J. , & Zainon, W. M. N. W. (2017). A framework for detecting ambiguity in software requirement specification. ICIT 2017–8th International Conference on Information Technology, Proceedings, May 2017, 209–213. https://doi.org/10.1109/ICITECH.2017.8080002.

Yang, H., Willis, A., De Roeck, A. , & Nuseibeh, B. (2010). Automatic detection of nocuous coordination ambiguities in natural language requirements. ASE '10: Proceedings of the IEEE/ACM international conference on Automated Software Engineering. 53. https://doi.org/10.1145/1858996.1859007.

Selection of Software Programmer Using Fuzzy MCDM Technique in Software Engineering Scenario

Pogarcic, I., Francic, M. and Davidovic, V. (2008), Application of AHP Method in Traffic Planning, ISEP.

Mishra, S. K. and Ray, A. (2012), Software developer selection: a Holistic approach for an eclectic decision, International Journal of Computer Application 47, 1, 12–18.

Yuen, K. K. F. and Lau, H. C.W. (2008), Software vendor selection using fuzzy analytic hierarchy process with ISO/IEC 9126, IAENG International Journal of Computer Science 35, 3.

Gungor, Z., Serhadlıoglu, G. and Kesen, S. E. (2009), A fuzzy AHP approach to personnel selection problem, Applied Soft Computing 9, 641–646.

Cebeci, U. (2009), Fuzzy AHP-based decision support system for selecting ERP systems in textile industry by using balanced scorecard, Expert Systems with Applications 36, 8900–8909.

Buyukozkan, G., Cifci, G. and Guleryuz, S. (2011), Strategic analysis of healthcare service quality using fuzzy AHP methodology, Expert Systems with Applications 38, 9407–9424.

Catak, F.O., Karabas, S. and Yildirim, S. (2012), Fuzzy analytic hierarchy based DBMS selection in Turkish national identity card management project, International Journal of Information Sciences and Techniques (IJIST) 2, 4, 29–38.

Javanbarg, M. B., Scawthorn, C., Kiyono, J. and Shahbodaghkhan, B. (2012), Fuzzy AHP-based multi-criteria decision making systems using particle swarm optimization, Expert Systems with Applications 39, 960–966.

Kabir, G. and Hasin, M. A. A. (2011), Comparative analysis of AHP and fuzzy AHP models for multicriteria inventory classification, International Journal of Fuzzy Logic Systems 1, 1, 1–16.

Sehra, S. K., Brar, Y. S. and Kaur, N. (2012), Multi criteria decision making approach for selecting effort estimation model, International Journal of Computer Applications (0975–8887) 39, 1, 10–17.

Bondor, C. I. and Muresan, A. (2012), Correlated criteria in decision models: Recurrent application of TOPSIS method, Applied Medical Informatics 30, 1, 55–63.

Wimatsari, G. A. M. S., Putra, K. G. D. and Buana, P.W. (2013), Multi-attribute decision making scholarship selection using a modified fuzzy TOPSIS, IJCSI International Journal of Computer Science Issues 10, 1 and 2, 309–317.

Pressman, R.S. (2005), Software Engineering, McGraw Hill Higher Education.

Hota, H.S., Singhai, S.K., Shukla, R. (2012), Application of fuzzy analytic hierarchy method in software engineering scenario, International Journal of Computer Applications (0975–8887) 57, 21, 45–50.

Milicic, D. (2004), Applying COCOMO II-A Case Study, Master Thesis Software Engineering, Thesis no: MSE-2004-19.

Zhu, K. J., Jing, Y., and Chang, D. Y. (1999), A discussion on extent analysis method and applications of fuzzy-AHP, European Journal of Operational Research, 116, 450–456.

Rao, R. V. (2007), Decision Making in the Manufacturing Environment, Springer-Verlag London Limited.

Saaty, T. L. (1980), The Analytical Hierarchy Process, McGraw Hill, New York.

Nalluri, S., Saraswathi, R. V., Ramasubbareddy, S., Govinda, K. , & Swetha, E. (2020). Chronic heart disease prediction using data mining techniques. In Data Engineering and Communication Technology (pp. 903–912). Springer, Singapore.

Virani, S. S., Alonso, A., Benjamin, E. J., Bittencourt, M. S., Callaway, C. W., Carson, A. P. , et al. (2020). Heart disease and stroke statistics—2020 update: A report from the American Heart Association. Circulation, 141(9), e139–e596.

U.S. National Library of Medicine. (2021). Heart disease prevention. MedlinePlus. Retrieved January 21, 2022 , from https://medlineplus.gov/howtopreventheartdisease.html.

World Health Organization. (2004). The Atlas of Heart Disease and Stroke.

Nason, E. (2007). An Overview of Cardiovascular Disease and Research. WR-467-RS.

R. Chadha , S. Mayank , A. Vardhan , and T. Pradhan . (2016). Application of data mining techniques on heart disease prediction: A survey. In Emerging Research in Computing, Information, Communication and Applications (pp. 413–426). Springer India, New Delhi.

A. A. Abdul-Aziz , P. Desikan , D. Prabhakaran , and L. F. Schroeder . (2019). Tackling the burden of cardiovascular diseases in India: The essential diagnostics list. Circulation: Cardiovascular Quality and Outcomes, 12(4), e005195.

Salman, O. H., Zaidan, A. A., Zaidan, B. B., Naserkalid, F. , & Hashim, M. (2017). Novel methodology for triage and prioritizing using “big data” patients with chronic heart diseases through telemedicine environmental. International Journal of Information Technology & Decision Making, 16(05), 1211–1245.

Raju, C., Philipsy, E., Chacko, S., Suresh, L. P. , & Rajan, S. D. (2018). A survey on predicting heart disease using data mining techniques. In 2018 Conference on Emerging Devices and Smart Systems (ICEDSS) (pp. 253–255). IEEE.

Marimuthu, M., Deivarani, S. , & Gayathri, R. (2019). Analysis of heart disease prediction using various machine learning techniques. In Advances in Computerized Analysis in Clinical and Medical Imaging (pp. 157–168). Chapman and Hall/CRC.

Singh, P., Singh, S. , & Pandi-Jain, G. S. (2018). Effective heart disease prediction system using data mining techniques. International Journal of Nanomedicine, 13, 121–124. https://doi.org/10.2147/ijn.s124998.

Pathak, A. K. , & Valan, J. A. (2020). A predictive model for heart disease diagnosis using fuzzy logic and decision tree. In Smart Computing Paradigms: New Progresses and Challenges (pp. 131–140). Springer, Singapore.

Mehmood, A., Iqbal, M., Mehmood, Z., Irtaza, A., Nawaz, M., Nazir, T. , & Masood, M. (2021). Prediction of heart disease using deep convolutional neural networks. Arabian Journal for Science and Engineering, 46(4), 3409–3422.

Isola, R., Carvalho, R. , & Tripathy, A. K. (2012). Knowledge discovery in medical systems using differential diagnosis, LAMSTAR, and k-NN. IEEE Transactions on Information Technology in Biomedicine, 16(6), 1287–1295.

Ayon, S. I., Islam, M. M. , & Hossain, M. R. (2020). Coronary artery heart disease prediction: a comparative study of computational intelligence techniques. IETE Journal of Research, 1–20.

Hassani, M. A., Tao, R., Kamyab, M. , & Mohammadi, M. H. (2020). An approach of predicting heart disease using a hybrid neural network and decision tree. In Proceedings of the 2020 5th International Conference on Big Data and Computing (pp. 84–89).

Shah, D., Patel, S. , & Bharti, S. K. (2020). Heart disease prediction using machine learning techniques. SN Computer Science, 1(6), 1–6.

Nagaprasad, S., Pushpalatha Reddy, T. , & Naga Lakshmi, S. (2020). Heart disease prediction propagation approach. International Journal of Machine Learning and Networked Collaborative Engineering, 4(2), 72–77. Retrieved from https://www.mlnce.net/index.php/Home/article/view/139.

Gavhane, A., Kokkula, G., Pandya, I. , & Devadkar, K. (2018). Prediction of heart disease using machine learning. In 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA) (pp. 1275–1278). IEEE.

Tasnim, F. , & Habiba, S. U. (2021). A comparative study on heart disease prediction using data mining techniques and feature selection. In 2021 2nd International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST) (pp. 338–341). IEEE.

Islam, M. T., Rafa, S. R. , & Kibria, M. G. (2020). Early prediction of heart disease using PCA and hybrid genetic algorithm with k-means. In 2020 23rd International Conference on Computer and Information Technology (ICCIT) (pp. 1–6). IEEE.
Vafaei, N., Ribeiro, R. A. , & Camarinha-Matos, L. M. (2018). Data normalisation techniques in decision making: case study with TOPSIS method. International Journal of Information and Decision Sciences, 10(1), 19–38.
Ma, W., Luo, X. , & Jiang, Y. (2017). Multicriteria decision making with cognitive limitations: A DS/AHPbased approach. International Journal of Intelligent Systems, 32(7), 686–721.
Lepri, B., Oliver, N., Letouzé, E., Pentland, A. , & Vinck, P. (2018). Fair, transparent, and accountable algorithmic decision-making processes. Philosophy & Technology, 31(4), 611–627.
Vafaei, N., Ribeiro, R. A. , & Camarinha-Matos, L. M. (2016). Normalization techniques for multi-criteria decision making: analytical hierarchy process case study. In Doctoral Conference on Computing, Electrical and Industrial Systems (pp. 261–269). Springer, Cham.
Lakshmi, T. M. , & Venkatesan, V. P. (2014). A comparison of various normalization in techniques for order performance by similarity to ideal solution (TOPSIS). International Journal of Computing Algorithm, 3, 882–888.
Chowdhury, P. , & Paul, S. K. (2020). Applications of MCDM methods in research on corporate sustainability: a systematic literature review. Management of Environmental Quality: An International Journal, 31, 2, 385–405.
Asadabadi, M. R., Chang, E. , & Saberi, M. (2019). Are MCDM methods useful? A critical review of analytic hierarchy process (AHP) and analytic network process (ANP). Cogent Engineering, 6(1), 1623153.
G. Adunlin , V. Diaby , and H. Xiao , (2015) Application of multi-criteria decision analysis in health care: a systematic review and bibliometric analysis. Health Expect, 18(6), 1894–1905.
T. D. C. Frazão , D. G. G. Camilo, E. L. S. Cabral , and R. P. Souza . (2018). Multi-criteria decision analysis (MCDA) in health care: A systematic review of the main characteristics and methodological steps. BMC Medical Informatics and Decision Making, 18(1), 90.
Aouadni, S., Rebai, A. , & Turskis, Z. (2017). The meaningful mixed data TOPSIS (TOPSIS-MMD) method and its application in supplier selection. Studies in Informatics and Control, 26(3), 353–363.
N. Vafaei, R. A. Ribeiro , and L. M. Camarinha-Matos (2016). Normalization techniques for multi-criteria decision making: Analytical hierarchy process case study. In Technological Innovation for Cyber-Physical Systems (pp. 261–269). Cham: Springer International Publishing.
Heart Disease UCI. (2018). Kaggle. https://www.kaggle.com/ronitf/heart-disease-uci.

State-of-the-Art Literature Review on Classification of Software Reliability Models

Acton, D. , Kourie, D. G. , Watson, B. W. (2014): Quality in software development: A pragmatic approach using metrics, SACJ, 52, 1–12.
Aggarwal, A. G. (2019): Multi release reliability growth modeling for open source software under imperfect debugging, In System Performance and Management Analytics, Springer, Singapore, pp. 77–86.
Ahmad, N. , Khan, M. G. M. , and Rafi, L. S. (2011): Analysis of an inflection S shaped software reliability model considering log-logistic testing-effort and imperfect debugging. International Journal of Computer Science and Network Security, 11(11), 161–171.
Amar, A. Radi, B. , and Hami, A. E. (2021): Reliability based design optimization applied to the high electron mobility transistor (HEMT), Microelectronics and Reliability, 124, 114299.
Bittanti, S. , Bolzern, P. , Pedrotti, E. , Pozzi, N. , and Scattolini, R. (1988): A flexible modeling approach for software reliability growth. In G. Goos , & J. Harmanis (Eds.), Software Reliability Modelling and Identification, Springer Verlag, Berlin, pp. 101–140.
Chang, Y. P. (2001): Estimation of parameters for nonhomogeneous Poisson process software reliability with change-point model, Communications in Statistics – Simulation and Computation, 30(3), 623–635.
Chatterjee, S. , and Shukla, A. (2016): Change point-based software model under imperfect debugging with revised concept of fault dependency, Journal of Risk and Reliability, 230(6),

579–597.
Chen, Y. , and Singpurwalla, N. D. (1997): Unification of software reliability models by self-exciting point processes, Advances Application Probability, 29, 337–352.
Choudhary, A. , Baghel, A. S. , Sangwan, O. P. (2018): Parameter estimation of software reliability model using firefly optimization. In: Data Engineering and Intelligent Computing, Springer, Singapore, pp. 407–415.
Erto, P. , Giorgio, M. , and Lepore, A. (2018): The generalized inflection S-shaped software reliability growth model. IEEE Transaction in Reliability, 69(1), 228–244.
Goel, A.L. (1985): Software reliability models: Assumptions, limitations and applicability. IEEE Transactions on Software Engineering, 12(2), 1411–1423. https://doi.org/10.1109/tse.1985.232177.
Goel, A. L. , and Okumoto, K. (1979a): A Markovian model for reliability and other performances measures of software system. In: Proceedings of COMPCON, IEEE Computer Society Press, Los Angeles, CA, pp.769–774.
Goel, A. L. , and Okumoto, K. (1979b): Time dependent error-detection rate model for software reliability and other performance measures. IEEE Transactions on Reliability, 28(3), 206–211.
Gokhale, S. , Hong, W.E. , Trivedi , and K., Horgan , (1998): An analytical approach to architecture based software reliability predication, Proceeding of the 3rd IEEE International Computer Performance and Dependability Symposium, 13–22.
Gokhale, S. S. , Wong, W.E. , Horgan, J. R. , and Trivedi, K. S. (2004): An analytical approach to architecture-based software performance and reliability prediction. Performance Evaluation, 58(4), 391–412, https://doi.org/10.1016/j.peva.2004.04.003.
Goos, G. and J. Harmanis (Eds.), Software Reliability Modeling and Identification, Springer, Berlin, pp. 101–140.
Hanagal, D. D. , and Bhalerao, N. N. (2016a): Analysis of NHPP software reliability growth model. International Journal of Statistics and Reliability Engineering, 3(2), 53–67.
Hanagal, D. D. , and Bhalerao, N. N. (2016b): Modeling and statistical inference on generalized inverse Weibull software reliability model. Journal of Indian Society of Probability and Statistics, 17(2), 145–160.
Hanagal, D. D. , and Bhalerao, N. N. (2017): Modeling on extended inverse Weibull software reliability growth model. Indian Association for Product, Quality and Reliability Transactions, 42(2), 75–95.
Hanagal, D. D. , and Bhalerao, N. N. (2018): Analysis of delayed S-shaped software reliability growth model with time dependent fault content rate function. Journal of Data Science, 17(3), 573–589.
Hanagal, D. D. , and Bhalerao, N. N. (2019): Modeling on generalized extended inverse Weibull software reliability growth model, Journal of Data Science, 16(4), 857–878.
Hanagal, D. D. , and Bhalerao, N. N. (2021): Software Reliability Growth Models, Infosys Science Foundation Series Published by Springer.
Hsu, C.J. and Huang, C.Y. (2010): A study on the applicability of modified genetic algorithms for the parameter estimation of software reliability modeling. In: Computer Software and Applications Conference (COMPSAC), IEEE 34th Annual, IEEE, pp. 531–540.
Hsu, C.J. , and Huang, C.Y. (2011): An adaptive reliability analysis using path testing for complex component-based software systems. IEEE Transactions on Reliability, 60(1), 158–170.
Huang, C.Y. (2005): Performance analysis of software reliability growth models with testing-effort and change-point. Journal of Systems and Software, 76(2), 181–194.
Huang, C.Y. , and Lyu, M.R. (2011): Estimation and analysis of some generalized multiple change-point software reliability models. IEEE Transactions on Reliability, 60(2), 498–514.
Huang, C. Y. , Lyu, M. R. and Kuo, S. Y. (2003): A unified scheme of some non homogenous Poisson process models for software reliability estimation. IEEE Transactions on Software Engineering, 29(3), 261–269.
Jeske, D. R. , and Zhang, X. (2005): Some successful approaches to software reliability modeling in industry. Journal of Systems and Software, 74(1), 85–99.
Jin, C. and Jin S.W. (2016): Parameter optimization of software reliability growth model with S-shaped testing-effort function using improved swarm intelligent optimization, Applied Soft Computing, 40, 283–291.
Kaliraj, S. , Vivek, D. , Kannan, M. , Karthick, K. , Dhasny Lydia, M. (2020): Critical review on software reliability models: Importance and application of reliability analysis in software

development. Materials Today: Proceedings.
Kapur, P. K. , and Garg, R. B. (1990): Optimal software release policies for software reliability growth model under imperfect debugging. Recherche Opertionnelle/Operations Research (RAIRO), 24, 295–305.
Kapur, P. K. , and Garg, R. B. (1992): A software reliability growth model for an error removal phenomenon, Journal of Software Engineering, 7, 291–294.
Kapur, P. K. , Garg, R. B. , and Kumar, S. (1999): Contributions to Hardware and Software Reliability, World Scientific Publishing Co. Ltd, Singapore.
Kapur, P.K. , Goswami, D.N. , and Gupta, A. (2004): A software reliability growth model with testing effort dependent learning function for distributed systems. International Journal of Reliability, Quality and Safety Engineering, 11(4), 365–377.
Kapur, P.K. , Gupta, D. , Gupta, A. , Jha, P.C. (2008): Effect of introduction of fault and imperfect debugging on release time, Ratio Mathematica, 18, 62–90.
Kaur, D. , & Sharma, M. (2015): Software reliability models: Time between failures and accuracy estimation, International Journal of Computer Science and Information Technologies, 6(4), 3370–3373.
Kim, T. , Lee, K. and Baik, J. (2015): An effective approach to estimating the parameters of software reliability growth models using a real-valued genetic algorithm. Journal of Systems and Software, 102, 134–144.
Komuro, Y. (1987): Evaluation of software products' testing phase application to software reliability growth models. In Reliability Theory and Application, ed. Osaki, S. and Cao, J. World Scientific, Singapore, pp. 188–194.
Kubat, P. and H. S. Kochc (1983): Pragmatic testing protocols to measure software reliability. IEEE Transactions on Reliability, R-32(4), 338–341.
Kuo, L. , and Yang, T. Y. (1995): Bayesian computation of software reliability. Journal Computations Graphical Statistics, 4, 65–82.
Lai, R. , and Garg, M. (2012): A detailed study of software reliability models. Journal of Software, 7(6), 1296–1306.
Li, F. , and Yi, Z. L. (2016): A new software reliability growth model: Multi generation faults and a power-law testing-effort function. Mathematical Problems in Engineering, 1–13.
Li, Q. Y. , and Pham, H. (2017): NHPP software reliability model considering the uncertainty of operating environments with imperfect debugging and testing coverage. Applied Mathematical Modelling, 51, 68–85.
Li, Q. Y. , and Pham, H. (2019): A generalized software reliability growth model with consideration of the uncertainty of operating environments. IEEE Access, 7, 84253–84267.
Li, Q.Y. , Li, H.F. , and Lu, M.Y. (2015): Incorporating S-shaped testing-effort functions into NHPP software reliability model with imperfect debugging, Journal of Systems Engineering and Electronics, 26(1), 190–207.
Lin, K. , and Chen Y. (2021): Two new multi-phase reliability growth models from the perspective of time between failures and their applications, Chinese Journal of Aeronautics, 34, 341–349.
Lyu, M. R. (1996): Handbook of Software Reliability Engineering, McGraw-Hill, New York.
Lyu, M. R. (1996): Handbook of Software Reliability Engineering, IEEE Computer Society Press, CA, 1996.
Mahapatra, G. S. , and Roy, P. (2012): Modified Jelinski-Moranda software reliability model with imperfect debugging phenomenon. International Journal of Computer Applications, 48, 38–46.
Musa, J. D. , Lannino, A. , and Okumoto, K. (1987): Software Reliability: Measurement, Prediction and Application, McGraw-Hill, New York.
Nakamura, H. , Uchihira, N. , Satoh, M. and Mizutani, H. (1985): A method of predicting software quality prior to the testing phase, Nikkei Electronics, 25(2), 233–260.
Nor, A.K.M. , Pedapati, S.R. , Muhammad, M. (2021): Reliability engineering applications in electronic, software, nuclear and aerospace industries: A 20 year review (2000–2020), Ain Shams Engineering Journal, 12(3), 3009–3019.
Obha, M. (1984): Software reliability analysis models. IBM Journal of Research Development, 28(4), 428–443.
Obha, M. , and Chou, X. (1989): Does imperfect debugging affect software reliability growth models? In Proceedings of the 4th International Conference on Software Engineering, Pittsburg, pp. 237–244.

Okamura, H. , Dohi, T. , Osaki, S. (2013): Software reliability growth models with normal failure time distributions. Reliability Engineering & System Safety, 116, 135–141.
Peng, Y.F. , Li, W.J. , Zhang, Q.P. , Hu , (2014): Testing effort dependent software reliability model for imperfect debugging process considering both detection and correction. Reliability Engineering & System Safety, 126, 37–43.
Pham, H. (2000): Software Reliability, Springer, New York.
Pham, H. (2006): System Software Reliability, Springer, New York.
Pham, H. (2007): An imperfect debugging fault-detection dependent parameter software. International Journal of Automation and Computing, 4(4), 325–328.
Pham, H. , and Zhang, X. (1997): An NHPP software reliability model and its comparison. International Journal of Reliability, Quality and Safety Engineering, 4, 269–282.
Pham, H. , and Zhang, X. (2003): NHPP software reliability and cost models with testing coverage, European Journal of Operational Research, 145(2), 443–454.
Pham, L. , and Pham, H. (2000): Software reliability models with time-dependent hazard function based on Bayesian approach. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, 30(1), 25–35.
Popstajanova, K. , and Trivedi, K. (2001): Architecture based approach to reliability assessment of software systems. Performance Evaluation, 45(2), 179–204.
Rahamneh, Z.A.L. , Reyalat, M. , Sheta, A. F. , Bani-Ahmad, S. , Al-Oqeili, S. (2011): A new software reliability growth model: Genetic-programming-based approach. Journal of Software Engineering and Applications, 4(8), 476–481.
Ramasamy, S. and Gopal, G. (2008): A software reliability growth model addressing learning. Journal of Applied Statistics, 35, 1151–1168.
Robinson, A.C. , Drake, R. R. , Swan, M. S. , Bennett, N. L. , Smith, T. M. , Hooper, R. , Laity, G. R. (2021): A software environment for effective reliability management for pulsed power design. Reliability Engineering & System Safety, 211, 107580.
Sahinoglu, M. (1992): Compound-Poisson software reliability model. IEEE Transactions on Software Engineering, 18(7), 624–630.
Schick, G. J. , and Wolverton, R. W. (1978): An analysis of competing software reliability models. IEEE Transactions on Software Engineering, 4(2), 104–120.
Shorthill, T. , Bao, H. , Zhang, H. , Ban, H. (2021): A novel approach for software reliability analysis of digital instrumentation and control systems in nuclear power plants. Annals of Nuclear Energy, 158, 108260.
Shyur, H.J. (2003): A stochastic software reliability model with imperfect debugging and change-point. Journal of Systems and Software, 66(2), 135–141.
Singh, V. B. , Yadav, K. , Kapur, R. , and Yadavalli, V. S. S. (2007): Considering the fault dependency concept with debugging time lag in software reliability growth modeling using a power function of testing time. International Journal of Automation and Computing, 4(4), 359–368.
Singpurwalla, N. D. (1991): Determining an optimal time interval for testing and debugging software. IEEE Transactions on Software Engineering, SE-17(4), 313–319.
Stark, G. , and Lyu, M. (1996): Software reliability tools. In The Handbook of Software Reliability Engineering, McGraw-Hill, New York.
Su, Y. S. and Huang, C. Y. (2007): Neural-network-based approaches for software reliability estimation using dynamic weighted combinational models. Journal of Systems and Software, 80(4), 606–615.
Subburaj, R , Gopal, G. and Kapur, P. K. (2012): A software reliability growth model for estimating debugging and the learning indices. International Journal of Performability Engineering, 8(5), 539–549.
Tahere, and Yaghoobi (2020): Parameter optimization of software reliability models using improved differential evolution algorithm. Mathematics and Computers in Simulation, 177, 46–62.
Teng, X. , & Pham, H. (2006): A new methodology for predicting software reliability in the random field environments. IEEE Transactions on Reliability, 53(3), 458–468.
Tokuno, K. , and Yamada, S. (2000): An imperfect debugging model with two types of hazard rates for software reliability assessment and assessment. Mathematical and Computer Modelling, 3, 343–352.

Uemura, M. , Yamada, S. and Fujino, K. (1990): Software reliability evaluation method: Application of delayed S-shaped NHPP model and other related models. Proceedings of the Annual Reliability and Maintainability Symposium, 467–472.
Van Pul, M. C. (1992): Asymptotic properties of a class of statistical models in software reliability. Scandinavian Journal of Statistics, 19(3), 235–253.
Wang, J. , Wu, Z. , Shu, Y. , and Zhang, Z. (2015): An imperfect software debugging model considering log-logistic distribution fault content function. Journal of Systems and Software, 100, 167–181.
Wang, J. , Wu, Z. , Shu, Y. and Zhang, Z. (2016): An optimized method for software reliability model based on non homogeneous Poisson process. Applied Mathematical Modelling, 40(13–14), 6324–6339.
Wang, W. L. , Pan, D. and Chen, M. H. (2006): Architecture-based software reliability modeling. Journal of Systems and Software, 79(1), 132–146.
Wood, A. (1996): Predicting software reliability. IEEE Computer, 11, 69–77.
Xie, M. (1991): Software Reliability Modelling, World Scientific, Singapore.
Xie, M. (2000): Software reliability models- past, present and future. In N. Limnios & M. Nikulin (Eds.), Recent Advances in Reliability: Methodology, Practice and Inference. Springer Science Business Media, LLC, New York.
Yamada, S. (1989): Software Reliability Assessment Technology, HBJ, Japan, Tokyo.
Yamada, S. (1991): Software quality/reliability measurement and assessment: Software reliability growth models and data analysis. Journal of Information Processing, 14, 254–266.
Yamada, S. (2014): Software Reliability Modeling Fundamentals and Applications. Springer: New York.
Yamada, S. , Isozokai, R , and Osaki, S. (1989): A software reliability evaluation tool, SRET. Transactions on IEICE, J72-D-I(1), 24–32.
Yamada, S. , Obha, M. , and Osaki, S. (1984): S-shaped software reliability growth models and their applications. IEEE Transactions on Reliability. 33, 289–292.
Yamada, S. , Tokuno, K. , and Osaki, S. (1992): Imperfect debugging models with fault introduction rate for software reliability assessment. International Journal of Systems Sciences, 23(12), 2241–2252.
Yamada, S. , Tokuno, K. , and Osaki, S. (1993): Software reliability measurement in imperfect debugging environment and its application. Reliability Engineering & System Safety Journal, 40(2), 139–147.
Yang, B. , Li, X. , Xie, M. , and Tan, F. (2010): A generic data-driven software reliability model with model mining technique. Reliability Engineering and System Safety, 95(6), 671–678.
Zhang, X. , and Pham, H. (2000): Comparisons of non-homogeneous Poisson process software reliability models and its applications. International Journal of Systems Science, 31(9), 1115–1123.
Zhang, X. , and Pham, H. (2006): Software field failure rate prediction before software deployment. Journal of Systems and Software, 79(3), 291–300.
Zhang, X. , Teng, H. and Pham, H. (2003): Considering fault removal efficiency in software reliability assessment. IEEE Transactions on System, Management and Cybernetics-Part A, 33(1), 114–120.

Survey on Software Reliability Modelling and Quality Improvement Techniques

Aasia Q. , Mehraj_Ud-Din D. , Quadri S.M.K. (2010), Improving software reliability using software engineering approach – a review, International Journal of Computer Applications (0975–8887) 10(5).
ANSI/IEEE. (1991), Standard Glossary of Software Engineering Terminology, STD-729-1991, ANSI/IEEE.
Ian S. (2007), Software Engineering, Eighth Edition (ISBN 13: 978-0-321-31379-9, ISBN 10: 0-321-31379-8). Pearson Education Limited. China Machine Press.

Jelinski Z. and Moranda P. (1972), Software reliability research, In Statistical Computer Performance Evaluation, W. Freiberger . New York: Academic, pp. 465–484

Pan J. (1999), Software Reliability, Carnegie Mellon University.

Musa J. D. (1993), Operational profiles in software- reliability engineering, IEEE Software 10(2), 14–32.

Musa J. D. Software Reliability Engineering: More Reliable Software, Faster and Cheaper, McGraw-Hill. ISBN 0-07-060319-7.

Musa J. D. (1997), Introduction to software reliability engineering and testing, 8th International Symposium on Software Reliability Engineering (Case Studies). November 2–5, 1997. Albuquerque, New Mexico.

Kumar M., Ahmad N. , Quadri S.M.K. (2005), Software reliability growth models and data analysis with a Pareto test effort, RAU Journal of Research 5(1–2), 124–128.

Shanmugam, L. and Florence, L. (2012), An overview of software reliability models, International Journal of Advanced Research in Computer Science and Software Engineering 2(10), 36–42.

Musa J. D. (1971), A theory of software reliability and its application, IEEE Transaction on Software Engineering SE-1, 312–327.

Reliability Analysis Center. (1996), Introduction to Software Reliability: A State of the Art Review. Reliability Analysis Center (RAC). http://rome.iitri.com/RAC/.

Raghvendra K. (2013), Approach for enhancing the reliability of software, Computing, Information Systems, Development Informatics & Allied Research 4(4).

Software Reliability Testing, Retrieved from http://en.wikipedia.org/w/index.php?title=Software_reliability_testing&oldid=645297567, 24th March, 2015 .

Nelson E. (1978), Estimating software reliability from test data, Microelectronics Reliability 17, 67–74.

Sinha S., Neeraj Kumar G. and Rajib M. (2019), Survey of combined hardware–software reliability prediction approaches from architectural and system failure viewpoint, International Journal of System Assurance Engineering and Management, Springer; The Society for Reliability, Engineering Quality and Operations Management (SREQOM), India, and Division of Operation and Maintenance, Lulea University of Technology, Sweden 10(4), 453–474.

Kim T. , Lee K. and Baik J. (2015), An effective approach to estimating the parameters of software reliability growth models using a real-valued genetic algorithm, Journal of Systems and Software 102, 134–144.

Rana R. , Staron M. , Berger C. , Hansson J. , Nilsson M. , Törner F. , et al. (2014), Selecting software reliability growth models and improving their predictive accuracy using historical projects data, Journal of Systems and Software 98, 59–78.

Ullah N. , Morisio M. and Vetro A. (2014), A method for selecting software reliability growth model to predict OSS residual defects, Computer 1–1.

Park J. and Baik J. (2015), Improving software reliability prediction through multi-criteria based dynamic model selection and combination, Journal of Systems and Software 101, 236–244.

Amin A. , Grunske L. , and Colman A. (2013), An approach to software reliability prediction based on time series modeling, Journal of Systems and Software 86, 1923–1932.

Multi-Criteria Decision Making for Software Vulnerabilities Analysis

H. Ghani , J. Luna , N. Suri . 2013. Quantitative assessment of software vulnerabilities based on economic-driven security metrics. International Conference on Risks and Security of Internet and Systems (CRiSIS).

R. Amankwah , P. K. Kudjo , S. Y. Antwi . 2017. Evaluation of software vulnerability detection methods and tools: A review. International Journal of Computer Applications 169, 8.

J. Willy , M. Amel , C. Ana . 2010. Software Vulnerabilities, Prevention and Detection Methods: A Review.

P. Zeng, G. Lin, L. Pan, Y. Tai and J. Zhang . 2020. Software vulnerability analysis and discovery using deep learning techniques: a survey. In IEEE Access, 8, pp. 197158–197172, https://doi.org/10.1109/ACCESS.2020.3034766.

B. Liu , L. Shi , Z. Cai , M. Li . 2012. Software vulnerability discovery techniques: A survey. Fourth International Conference on Multimedia Information Networking and Security. 978-0-7695-4852-4/12 $26.00. © 2012 IEEE. https://doi.org/10.1109/MINES.2012.202.
P. Godefroid, A. Kiezun, M. Y. Levin . 2008. Grammar-based whitebox fuzzing. Proceedings of the 2008 ACM SIGPLAN Conference on Programming Language Design and Implementation.
M. Bohme , C. Cadar, A. Roychoudhury . 2020. Fuzzing: Challenges and Reflections. National University of Singapore, Singapore IEEE Software Published by the IEEE Computer Society, IEEE.
M. Woo , S. K. Cha, S. Gottlieb , D. Brumley . 2013. Scheduling black-box mutational fuzzing. Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security. https://doi.org/10.1145/2508859.2516736.
T. Clarke . 2009. Fuzzing for Software Vulnerability Discovery. Technical Report RHUL-MA-2009-04. Department of Mathematics Royal Holloway, University of London.
G. Vache . 2009. Vulnerability analysis for a quantitative security evaluation, 3rd International Symposium on Empirical Software Engineering and Measurement, 2009, pp. 526–534, https://doi.org/10.1109/ESEM.2009.5315969.
O. H. Alhazmi and Y. K. Malaiya , 2005. Quantitative vulnerability assessment of systems software. Annual Reliability and Maintainability Symposium, 2005 Proceedings. pp. 615–620, https://doi.org/10.1109/RAMS.2005.1408432.
de Azevedo Reis, G., de Souza Filho, F.A., Nelson, D.R. et al. 2020. Development of a drought vulnerability index using MCDM and GIS: study case in Sao Paulo and Ceara, Brazil. Natural Hazards 104, 1781–1799. https://doi.org/10.1007/s11069-020-04247-7.
N. Liu, J. Zhang, H. Zhang and W. Liu . 2010. Security assessment for communication networks of power control systems using attack graph and MCDM. IEEE Transactions on Power Delivery 25, 3, 1492–1500, https://doi.org/10.1109/TPWRD.2009.2033930.
E. Kornyshova and C. Salinesi . 2007. MCDM techniques selection approaches: State of the art. 2007 IEEE Symposium on Computational Intelligence in Multi-Criteria Decision- Making, pp. 22–29, https://doi.org/10.1109/MCDM.2007.369412.
M. Velasquez , Patrick T. 2013. An analysis of multi-criteria decision making methods. International Journal of Operations Research 10, 2, 5666.
C. Luo , W. Bo , H. Kun , L. Yuesheng . 2020. Study on software vulnerability characteristics and its identification method. Mathematical Problems in Engineering 2020, Article ID 1583132, 6 p. https://doi.org/10.1155/2020/1583132.
B. Liu , L. Shi , Z. Cai , M. Li . 2012. Software vulnerability discovery techniques: a survey. Fourth International Conference on Multimedia Information Networking and Security.
L. Ping , S. Jin , Y. Xinfeng . 2011. Research on software security vulnerability detection technology. International Conference on Computer Science and Network Technology.
Yurdakul, M., Tansel, I. C. Y. 2009. Application of correlation test to criteria selection for multicriteria decision making (MCDM) models. The International Journal of Advanced Manufacturing Technology 40, 403–412. https://doi.org/10.1007/s00170-007-1324-1.
Arslan M. C. 2002. A Decision Support System for Machine Tool Selection, M.S. Thesis, Sabancı University.
M. Velasquez , P. T. Hester . 2013. An analysis of multi-criteria decision making methods. International Journal of Operations Research 10, 2, 5666. M. Aruldoss , T. Miranda Lakshmi , V. Prasanna Venkatesan , 2013. A survey on multi criteria decision making methods and its applications. American Journal of Information Systems 1, 1, 31–43. https://doi.org/10.12691-ajis-1-1-5.
U. S. Kashid, A. U. Kashid, S. N. Mehta . A review of mathematical multi-criteria decision models with a case study. International Conference on Efficacy of Software Tools for Mathematical Modeling (ICESTMM'19).
S. Opricovic , G.-H. Tzeng . 2004. Compromise solution by MCDM methods: A comparative analysis of VIKOR and TOPSIS. European Journal of Operational Research 156, 445–455.
S. Narang , P. K. Kapur, D. Damodaran and R. Majumdar . 2018. Prioritizing types of vulnerability on the basis of their severity in multi-version software systems using DEMATEL technique, 2018 7th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), pp. 162–167, https://doi.org/10.1109/ICRITO.2018.8748720.

H.A. Proper , D. Greefhorst . 2011. The Roles of Principles in Enterprise Architecture. https://doi.org/10.1007/978-3-642-20279-7-2.

Kotusev, Svyatoslav . 2019. Enterprise architecture and enterprise architecture artifacts: Questioning the old concept in light of new findings. Journal of Information Technology 34, 102–128. https://doi.org/10.1177/0268396218816273.

A. M. Ghaleb , H. Kaid , A. Alsamhan , S. H. Mian , and L. Hidri . 2020. Assessment and comparison of various MCDM approaches in the selection of manufacturing process. Advances in Materials Science and Engineering, Article ID 4039253, 16 p. https://doi.org/10.1155/2020/4039253.

On a Safety Evaluation of Artificial Intelligence-Based Systems to Software Reliability

Allen Currit, P. , M. Dyer , and H. D. Mills (1986). Certifying the reliability of software. IEEE Transactions on Software Engineering SE-12(1), 3–11.

Anscombe, F. J. (1973). Graphs in statistical analysis. American Statistician, 27(1), 17–21.

Baldwin, G. (1987). Implementation of physical units. SIGPLAN Notices 22(8), 45–50.

Ballestar, M.T. , Grau-Carles, P. and Sainz, J. (2019). Predicting customer quality in e-commerce social networks: a machine learning approach. Review of Managerial Science, 13(3), 589–603.

Baron, S. and C. Feehrer (1985). An Analysis of the Application of AI to the Development of Intelligent Aids for Light Crew Tasks. Contractor Report 3944, NASA Langley Research Center, Hampton, VA.

Beam, A.L. and Kohane, I.S. (2018). Big data and machine learning in health care. JAMA, 319(13), 1317–1318.

Bellman, K. L. and D. O. Walter . (1987). Testing rule-based expert systems. Course Notes for Analyzing the Reliability and Performance of Expert Systems, UCLA Extension.

Bhavsar, P. , Safro, I. , Bouaynaya, N. , Polikar, R. and Dera, D. , (2017). Machine learning in transportation data analytics. In Data Analytics for Intelligent Transportation Systems (pp. 283–307). Elsevier.

Boughaci, D. and Alkhawaldeh, A.A. (2020). Appropriate machine learning techniques for credit scoring and bankruptcy prediction in banking and finance: A comparative study. Risk and Decision Analysis, 1–10.

Braband, J , H. Gall , H. Schäbe (2018). Proven in use for software: Assigning an SIL based on statistics. In Handbook of RAMS in Railway systems – Theory and Practice, Q. Mahboob , E. Zio (Eds.), Boca Raton, FL, Taylor and Francis, Chapter 19, pp. 337–350.

Brunette, E.S. , R.C. Flemmer , C.L. Flemmer (2009). A review of artificial Intelligence, Proceedings of 4th International Conference on Autonomous Robots and Agents, Feb. 10–12, 2009, Wellington, pp. 385–392.

Char, D.S. , Shah, N.H. and Magnus, D. (2018). Implementing machine learning in health care addressing ethical challenges. The New England Journal of Medicine 378(11), 981.

Chen, S. H. , A. J. Jakeman , J.P. Norton (2008). Artificial Intelligence techniques: An introduction to their use for modelling environmental systems. Mathematics and Simulation 78, 379–400.

Corni, M. (2019). Is Artificial Intelligence Racist? (And Other Concerns), 817fa60d75e9, retrieved on October 25, 2018.

Cybenko, G. (1989). Approximations by superpositions of sigmoidal functions. Mathematics of Control, Signals, and Systems 2(4), 303–314.

Davis, R. (1980). Teiresias: Applications of meta-level knowledge to the construction, maintenance, and use of large knowledge bases. PhD thesis, Computer Science Department, Stanford University, CA, 1976. Reprinted with revisions in Knowledge-Based Systems in Artificial Intelligence, R. Davis and D. B. Lenat , (Eds.), McGraw-Hill, New York, NY.

Edwards, J.R. (2007). Polynomial regression and response surface methodology. Perspectives on Organizational Fit, 361–372.

Elmurngi, E. and Gherbi, A. (2017). An empirical study on detecting fake reviews using machine learning techniques. In 2017 Seventh International Conference on Innovative Computing

Technology (INTECH) (pp. 107–114). IEEE.
Ermagun, A. and Levinson, D. (2018). Spatiotemporal traffic forecasting: review and proposed directions. Transport Reviews 38(6), 786–814.
Freedman, D.A. (2009). Statistical Models: Theory and Practice. Cambridge University Press.
Genesereth, M. R. and N. J. Nilsson . (1987). Logical Foundations of Artificial Intelligence. Morgan Kaufmann Publishers Inc., Los Altos, CA.
Goodfellow, I. , Bengio, Y. and Courville, A. (2016). Deep Learning. MIT Press.
Harmon, P. and D. King (1985). Expert Systems: Artificial Intelligence in Business. John Wiley and Sons, New York, NY.
Hättasch, N. , N. Geisler (2019). The Deep Learning Hype, Presentation at 36C3.
IEC 61508 Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related Systems, 2010.
Ivanov, A.I. , E.N. Kuprianov , S.V. Tureev (2019). Neural network integration of classical statistical tests for processing small samples of biometrics data. Dependability (Moscow) 19(2), 22–27.
Kabir, H.D. , Khosravi, A. , Hosen, M.A. and Nahavandi, S. (2018). Neural network-based uncertainty quantification: A survey of methodologies and applications. IEEE Access, 6, 36218–36234.
Kapur, D. and D. R. Musser (1987). Proof by consistency. Artificial Intelligence, 31(2), 125–157.
Kou, G. , Chao, X. , Peng, Y. , Alsaadi, F.E. and Herrera-Viedma, E. (2019). Machine learning methods for systemic risk analysis in financial sectors. Technological and Economic Development of Economy, 25(5), 716–742.
Leo, M. , Sharma, S. and Maddulety, K. (2019). Machine learning in banking risk management: A literature review. Risks, 7(1), 29.
Leveson, N. G. (1984). Software safety in computer controlled systems. IEEE Computer 17(2), 48–55.
Leveson, N. G. (1986). Software Safety: Why, What and How. ACM Computing Surveys 18(2), 125–163.
Leveson, N. G. and P. R. Harvey (1983). Analyzing software safety. IEEE Transactions on Software Engineering SE-9(5), 569–579.
Lewis, R.J. (2000). An introduction to classification and regression tree (CART) analysis. In Annual Meeting of the Society for Academic Emergency Medicine in San Francisco, California (Vol. 14).
Li, X. , Zhang, W. and Ding, Q. (2019). Deep learning-based remaining useful life estimation of bearings using multi-scale feature extraction. Reliability Engineering & System Safety 182, 208–218.
Liaw, A. and Wiener, M. (2002). Classification and regression by random forest. R News 2(3), 18–22.
Minh, L.Q. , Duong, P.L.T. and Lee, M. (2018). Global sensitivity analysis and uncertainty quantification of crude distillation unit using surrogate model based on Gaussian process regression. Industrial & Engineering Chemistry Research, 57(14), 5035–5044.
Mosavi, A. and Varkonyi, A. (2017). Learning in robotics. International Journal of Computer Applications, 157(1), 8–11.
Murphy, K.P. (2012). Machine Learning: A Probabilistic Perspective. MIT Press.
Musa, J. D. A. Iannino , and K. Okumoto (1987). Software Reliability: Measurement, Prediction, Application. McGraw Hill, New York, NY.
Panch, T. Szolovits, P. and Atun, R. (2018). Artificial intelligence, machine learning and health systems. Journal of Global Health, 8(2).
Pearl, J. , D. Mackenzie (2018). The Book of Why, Penguin Science.
Polydoros, A.S. and Nalpantidis, L. , (2017). Survey of model-based reinforcement learning: Applications on robotics. Journal of Intelligent & Robotic Systems, 86(2), 153–173.
Putzer, H. (2019). Ein strukturierter Ansatz für funktional sichere KI, Presentation at DKE Funktionale Sicherheit, Erfurt.
Rath, M. , (2020). Machine learning and its use in e-commerce and e-business. In 39 Handbook of Research on Applications and Implementations of Machine Learning Techniques (pp. 111–127). IGI Global.
Reiter, R. (1987). A theory of diagnosis from first principles. Artificial Intelligence 32, 57–95.

Schäbe, H. (2018). SIL apportionment and SIL allocation. In Handbook of RAMS in Railway systems – Theory and Practice, Q. Mahboob , E. Zio (Eds.), Boca Raton, FL, Taylor and Francis, Chapter 5, pp. 69–78.
Schutte, P. C. and K. H. Abbott (1986). An artificial intelligence approach to onboard fault monitoring and diagnosis for aircraft applications. In Proceedings, AIAA Guidance and Control Conference, Williamsburg, VA.
Smola, A.J. and Schölkopf, B. (2004). A tutorial on support vector regression. Statistics and Computing 14(3), 199–222.
Stone, P. and Veloso, M. (2000). Multiagent systems: A survey from a machine learning perspective. Autonomous Robots 8(3), 345–383.
Underwriter Laboratories: Standard for Safety for the Evaluation of Autonomous Products, draft UL 4600, 2019.
Wang, J. J. , Y. Ma , L. Zhang , R. X. Gao , D. Wu (2018). Deep learning for smart manufacturing: Methods and applications. Journal of Manufacturing Systems 48, 144–156.
Wigger, P. (2018). Independent safety assessment – Process and methodology. In Handbook of RAMS in Railway Systems – Theory and Practice, Q. Mahboob , E. Zio (Eds.), Boca Raton, FL, Taylor and Francis, Chapter 5, pp. 475–485.
Zantalis, F. , Koulouras, G. , Karabetsos, S. and Kandris, D. (2019). A review of machine learning and IoT in smart transportation. Future Internet 11(4), 94.
Zio, E. (2009). Reliability engineering: Old problems and new challenges. Reliability Engineering & System Safety 94(2), 125–141.

Study and Estimation of Existing Software Quality Models to Predict the Reliability of Component-Based Software

S. McConnell , Rapid Development, Microsoft Press, 1996.
S. McConnell , Software Project Survival Guide, Microsoft Press, 1997
H. Gumuskaya , Core issues affecting software architecture in enterprise projects, World Academy of Science, Engineering and Technology, 9, 32–37, 2005.
Suman and M. Wadhwa , A comparative study of software quality models, International Journal of Computer Science and Information Technologies, 5, 4, 5634–5638, 2014.
F. Deissenboeck , E. Juergens , K. Lochmann , and F. Informatik , Software quality models: Purposes, usage scenarios and requirements, In Proceedings of the ICSE Workshop on Software Quality, pp. 9–14, 2009, https://doi.org/10.1109/WOSQ.2009.5071551.
M. Yan , X. Xia , X. Zhang, L. Xu , D. Yang and S. Li , Software quality assessment model: A systematic mapping study, Science China Information. Sciences, 62, Article no. 191101, 2019. https://doi.org/10.1007/s11432-018-9608-3.
ANSI/IEEE Std 729-1983, IEEE Standard Glossary of Software Engineering Terminology, The IEEE, Inc, New York.
ISO/IEC25010:2011(en)Systemsandsoftwareengineering—SystemsandsoftwareQualityRequirementsandEvaluation(SQuaRE)—Systemandsoftwarequalitymodels,2015.
M. Jedlicka , O. Moravcik and P. Schreiber , Survey to Software Reliability, Central European Conference on Information and Intelligent Systems, CECIIS, pp. 1–5, 2008.
S. Khatri R.S., Chillar and A. Chhikara , Analyzing the impact of software reliability growth models on object oriented systems during testing, International Journal of Enterprise Computing and Business Systems, 2, 1, pp. 1–10, 2012.
J. P. Miguel , D. Mauricio and G. Rodriguez , A review of software quality models for the evaluation of software products, International Journal of Software Engineering & Applications (IJSEA), 5, 6, pp. 31–53, 2014. https://doi.org/10.5121/ijsea.2014.5603.
J. Gao , H.-S. Tsao , and Y. Wu , Testing and Quality Assurance for Component-Based Software, Artech House, 2003.
X. Cai, M. R. Lyu , K.-F. Wong , and R. Ko , Component-based software engineering: Technologies, development frameworks, and quality assurance schemes, In Proceedings of 7th Asia-Pacific Conference on Software Engineering (APSEC), pp. 372–379, 2000.

F. Brosch , H. Koziolek , B. Buhnova , and R. Reussner , Parameterized Reliability Prediction for Component Based Software Architectures, QoSA 2010, LNCS 6093, pp. 36–51, 2010.
O. Dahiya , K. Solanki, S. Dalal and A. Dhankhar , An exploratory retrospective assessment on the usage of bio-inspired computing algorithms for optimization, International Journal of Emerging Trends in Engineering Research, 8, 2, 414–434, 2020. https://doi.org/10.30534/ijeter/2020/29822020.
Pressman, R. S. Software Engineering a Practitioner's Approach, 7th Ed. McGraw-Hill, Inc., 2012.
D. P. Narayani , and P. Uniyal , Comparative analysis of software quality models, International Journal of Computer Science and Management Research, 2, 3, 1911–1913, 2013.
D. Jamwal , Analysis of quality models for organizations, International Journal of Latest Trends in Computing, 1, 2, 19–23, 2010.
R. S. Jamwal , D. Jamwal and D. Padha , Comparative analysis of different software quality models, In Proceedings of 3rd National Conference, Computing For Nation Development, Bharati Vidyapeeth's Institute of Computer Applications and Management, pp. 1–5, 2009.
P. Nistala, K. V. Nori and R. Reddy , Software quality models: A systematic mapping study, In IEEE/ACM International Conference on Software and System Processes (ICSSP), pp. 125–134, 2019.
B. Kitchenham and S. Linkman , The SQUID approach to defining a quality model, Software Quality Journal, 6, 211–223, 1997.
M. Ortega , Construction of a systematic quality model for evaluating a software product, Software Quality Journal, 11, 3, 219–242, 2003.
R. Marinescu and D. Ratiu , Quantifying the quality of object-oriented design: The factor-strategy model, In 11th Working Conference on Reverse Engineering, Delft, pp. 192–201, 2004.
F. Khomh and Y. G. Gueheneuc , Dequalite: Building design-based software quality models, In Proceedings of the 15th Conference on Pattern Languages of Programs, pp. 1–7, 2008, https://doi.org/10.1145/1753196.1753199.
A. Finne , Towards a quality meta-model for information systems, Software Quality Journal, 19, 4, 663–688, 2011.
R. S. Kenett , and E. Baker , Process Improvement and CMMI® for Systems and Software, CRC Press, 2010.

Performance of Multi-Criteria Decision-Making Model in Software Engineering – A Survey

Albayrak, E., Erensal, Y. C. (2005). A study bank selection decision in Turkey using the extended fuzzy AHP method. Proceeding of 35th International Conference on Computers and Industrial Engineering, Istanbul, Turkey.
Aldlaigan, A., Buttle, F.A. (2002). A new measure of bank service quality. International Journal of Service Industry Management, 13, 38–362.
Business Credits (2006). Non-financial data can predict future profitability. BusinessCredits, 108(4), 57.
Nikoomaram, H., M. Mohammadi, M. JavadTaghipouria and Y. Taghipourian (2009). Training Performance Evaluation of Administration Sciences Instructors by Fuzzy MCDM Approach. Tehran, Iran.
Y. TanselIç (2012). Development of a credit limit allocation model for banks using an integrated Fuzzy TOPSIS and linear programming. Expert System with Applications, 39(5), 5309–5316.
T. Ozcan , N. Celebi (2011). Comparative analysis of multi-criteria decision making methodologies and implementation of a warehouse location selection problem. Expert System with Applications, 38, 9773–9779.
Mohammad SaeedZaeri , Amir Sadeghi , Amir Naderi , et al. (2011). Application of multi criteria decision making technique to evaluation suppliers in supply chain management, African Journal of Mathematics and Computer Science Research, 4(3), 100–106.
Schinas O. (2007). Examining the use and application of multi - criteria decision making techniques in safety assessment, International Symposium on Maritime Safety, Security & Environmental Protection, Athens.

Anjali Awasthia, S.S. Chauhanb, S.K. Goyalb (2000). A Multi- Criteria Decision Making Approach for Location Planning for Urban Distribution Centers under Uncertainty. CIISE, Montreal, Canada.
T.C. Chu (2002). Facility location selection using fuzzy TOPSIS under group decisions, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 10(6), 687–701.
Mahdi Zarghami, Ferenc Szidarovszky (2011). Revising the OWA Operator for Multi Criteria Decision Making Problems under Uncertainty a Faculty of Civil Engineering, Tabriz, Iran.
Doraid Dalalah, Mohammed Hayajneh, Farhan Batieha (2011). A fuzzy multi-criteria decision making model for supplier selection, Expert Systems with Applications, 38, 8384–8391.
Yong-Sheng Ding , Zhi-HuaHu, Wen-Bin Zhang (2011). Multi-criteria decision making approach based on immune co-evolutionary algorithm with application to garment matching problem, Expert Systems with Applications, 38, 10377–10383.
Y. Peng , Y. Zhang, Y. Tang, S. Li (2011). An incident information management framework based on data integration, datamining, and multi-criteria decision making, Decision Support Systems, 51, 316–327.
M. Dursun , E. ErtugrulKarsak , Melis Almula Karadayi (2011). Assessment of health-care waste treatment alternatives using fuzzy multi-criteria decision making approaches, Expert Systems with Applications, 38, 10377–10383.
S. Rezaiana , S. Ali Joziba (2012). Health-safety and environmental risk assessment of refineries using of multi criteria decision making method, APCBEE Procedia, 3, 235–238.
K. Sadeghzadeh, M. Bagher Salehi (2011). Mathematical analysis of fuel cell strategic technologies development solutions in the automotive industry by the TOPSIS multi-criteria decision making method, International Journal of Hydrogen Energy, 36, 13272–13280.
Hung-Yi Wua , Gwo-Hshiung Tzenga , Yi-Hsuan Chen (2011). A Fuzzy MCDM Approach for Evaluating Banking Performance based on Balanced Scorecard, Taiwan.
Ne Lu, CengizKahramanb (2007). Fuzzy performance evaluation in Turkish banking sector using analytic hierarchy process and TOPSIS. Expert Systems with Applications, 36, 11699–11709.
Ashton, C. (1998). Balanced scorecard benefits Nat west bank. International Journal of Retail and Distribution Management, 26(10), 400–407.
Tsai, W. H., Yang, C. C., Leu, J. D., Lee, Y. F. , & Yang, C. H. (2001). An integrated group decision making support model for corporate financing decisions. Group Decision and Negotiation, 1–25. C.T. Chen , A fuzzy approach to select the location of the distribution center, Fuzzy Sets and Systems, 118, 65–73.
S.Y. Chou, Y.H. Chang, C.Y. Shen (2008). A fuzzy simple additive weighting system under group decision making for facility location selection with objective/subjective attributes, European Journal of Operational Research, 189(1), 132–145.
A. Mazumdar (2010). Application of multi-criteria decision Making (MCDM) approaches on teachers Performance evaluation and appraisal.
Chen, C.W.E. (2000). Extensions of the TOPSIS for group decision-making under fuzzy environment. Fuzzy Sets and Systems, 114, 1–9.
Rabah Medjoudj , Djamil Aissan , Klaus Dieter Haim (2013). Power customer satisfaction and profitability analysis using multi-criteria decision making methods, Electrical Power and Energy Systems, 45, 331–339.
V. Lakshmana Gomathi Nayagam , S. Murali Krishnan , Geetha Sivaraman , (2011). Multi-criteria decision-making method based on interval-valued intuitionistic fuzzy sets, The Journal of Analysis, 1464–1467.
H.C.W. Lau , C.W.Y. Wong , P.K.H. Lau , K.F. Pun , B. Jiang , K.S. Chin , (2003). A fuzzy multi-criteria decision support procedure for enhancing information delivery in extended enterprise networks, Engineering Applications of Artificial Intelligence, 16, 1–9.
Shu-Hsien Liao , and Kuo-Chung Lu (2002). Evaluating anti-armor weapon using ranking fuzzy numbers, Tamsui Oxford Journal of Mathematical Sciences, 11(1), 33–48.
M. Haghighi , A. Divandari, M. Keimasi (2010). The impact of 3D e-readiness on e-banking development in Iran: A fuzzy AHP analysis, Expert Systems with Applications, 37(6), 4084–4093.
Anderson, W., Jr., Cox, J. E. P. , & Fulcher, D. (1976). “Bank selection decisions and marketing segmentation”. Journal of Marketing, 40(1), 40–45.

Arshadi, N., Lawrence, E. C. (1987). An empirical investigation of new bank performance. Journal of Banking and Finance, 11(1), 33–48.
Ashton, C. (1998). Balanced scorecard benefits Nat West Bank. International Journal of Retail and Distribution Management, 26(10), 400–407.
Athanassopoulos, Giokas D. (2000). On-going use of data envelopment analysis in banking institutions. Evidence from the Commercial Bank of Greece. Interfaces, 30(2), 81–95.
Bauer, P. W., Berger, A. N., Ferrier, G. D. , & Humphrey, D. B. (1998). Consistency conditions for regulatory analysis of financial institutions: A comparison of frontier efficiency methods. Journal of Economic and Business, 50(2), 85–114.
Beccalli, A. (2007). Does IT investment improve bank performance? Evidence from Europe. Journal of Banking & Finance, 31, 2205–2230.
Caballero, R., Cerda, E., Munoz, M.M. , Rey, L. (2004). Stochastic approach versus multi objective approach for obtaining efficient solutions in stochastic multi objective programming problems. European Journal of Operational Research, 158, 633–648.
Changchit, C., Terrell, M.P. (1993). A multi-objective reservoir operation model with stochastic inflows. Computers and Industrial Engineering, 24(2), 303–313.
Chen, S.J., Hwang, C.L. (1991). Fuzzy Multiple Attribute Decision Making. Springer-Verlag, Berlin.
Y. Y. Jusoh , K. Chamili , N. C. Pa , and J. H. Yahaya (2014). Open Source software selection using an analytical hierarchy process (AIIP), American Journal of Software Engineering and Applications, 3(6), 83–89.
Vinay S, S. Aithal and S. Adiga (2014). Integrating goals after prioritization and evaluation a goal-oriented requirements engineering methods, International Journal of Software Engineering & Applications (IJSEA), 5(6).
Sumeet Kaur Sehra , Yadwinder Singh Brar and Navdeep Kaur (2016). Application of multi-criteria decision making in software engineering, International Journal of Advanced Computer Science and Applications, 7(7).
Mary, S.A.S.A. and Suganya, G. (2016). Multi-criteria decision making using electre. Circuits and Systems, 7, 1008–1020. http://dx.doi.org/10.4236/cs.2016.76085.
Vanessa B.S. , et al. (2016). A multicriteria approach for selection of agile methodologies in software development projects, 2016 IEEE International Conference on Systems, Man and Cybernetics, SMC 2016, October 9–12, Budhapest, Hungary.
E. N. Madi , J. M. Garibaldi , C. Wagner , An exploration of issues and limitations in current methods of TOPSIS and fuzzy TOPSIS, 2016 IEEE International Conference on Fuzzy Systems (FUZZ), 978-1-5090-0626-7/16/$3l.00 ©2016 IEEE.
Khan, J. (2016). Requirements prioritization using analytic network process (ANP). International Journal of Scientific & Engineering Research, 7(11).
R. Santos , A. Albuquerque , P. R. Pinheiro (2016). Towards the applied hybrid model in requirements prioritization. Procedia Computer Science, 91, 909–918.
E. Elsherbeiny , Hadeel, Ahmed , Abd El-Aziz, R. Nagy (2017). Decision support for requirements prioritization using data analysis. Egyptian Computer Science Journal (ECS), 41.
Q. Raneem , A.-S. Alaa, H. Q. Mais, H. Amjad (2017). Requirements Prioritization Techniques Review and Analysis, pp. 258–263. 10.1109/ICTCS.2017.55.
Hassan, A. and Ramadan, N. (2017). A Fuzzy approach for Wieger's method to rank priorities in Requirement Engineering. CiiT International Journal of Fuzzy Systems, 9, 189–196.
Hassan, A. and Ramadan, N. (2018). A proposed hybrid prioritization technique for software requirements based on fuzzy logic. CiiT International Journal of Fuzzy Systems, 10, 45–52.

Optimization Software Development Plan

Saliu, M. , and Ruhe, G. , (2005). Software release planning for evolving systems. Innovations in Systems and Software Engineering, 1, 189-204, 2005. Doi: 10.1007/s11334-005-0012-2.
Natt-och-Dag, J., Regnell, B., Gervasi, V., and Brinkkemper, S. , A linguistic- engineering approach to large-scale requirements management, IEEE Software, 22, 1, 32–39, 2005.
Ahmed, M. , Saliu, M. O. , and AlGhamdi, J. , Adaptive fuzzy logic-based framework for software development effort prediction, Information and Software Technology, 47, 1, 31–48,

2005.
Bagnall, A. J., Rayward-Smith, V. J. , and Whittley, I. M. , The next release problem. Information and Software Technology, 43, 14, 883–890, 2001.
Carlshamre, P. , Release planning in market-driven software product development: Provoking an understanding, Requirements Engineering, 7, 3, 139–151, 2002.
Ruhe, G. , Software release planning, In Handbook of Software Engineering and Knowledge Engineering, Vol. 3, Chang, S. K. (Ed.): World Scientific Publishing, 2005.
http://www.lucidchart.com/documents/editNewOrRegister/9da7fb63-6026-4521-b8b3-3a81645e71d2.
Fisseha, B. , Data Scientist. https://www.linkedin.com/in/fisseha-berhane-phd-543b5717?trk=nav_responsive_tab_profile.
ScienceDirect. The RO method is defined as an alternative modelling method for handling optimization problems with uncertain parameters. In Classical and Recent Aspects of Power System Optimization, 2018.
Blake, S. , How to Approach Your Agile Release Plan for Successful Development, Head of Marketing, Wollongong.
https://www.knowledgehut.com/tutorials.
Lehman, M. M. , Laws of software evolution revisited. In Proceedings of 5th European Workshop on Software Process Technology (EWSPT'96), Nancy, France, 1996, pp. 108–124.
Denne M, Cleland-Huang J. , The incremental funding method: data driven software development. IEEE Software, 21, 3, 39–47, 2004.
Saliu O, Ruhe G. , Supporting software release planning decisions for evolving systems. In Proceedings of 29th IEEE/NASA Software Engineering Workshop, Greenbelt, USA, 6–7 April, 2005.

A Time-Variant Software Stability Model for Error Detection

AASHTO. 2017. Specifications for Bridge Construction (LRFD). AASHTO, Washington, DC, 8th ed.
Abadi, M. et al. 2016. TensorFlow: A large-scale machine learning system. In Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI'16). The USENIX Association is based in Berkeley, California, pp. 265–283.
Ahammed, M. and R. E. Melchers , 1997. Underground pipes subjected to coupled loads and corrosion: A probabilistic analysis. Eng. Struct. 19(12), 988–994. https://doi.org/10.1016/S0141-0296(97)\s00043-6.
Akiyama, M. , D. M. Frangopol , M. Arai , and S. Koshimura , 2012. A probabilistic analysis of bridge structural performance under tsunami threat. In Proceedings of the 2012 Structures Congress. ASCE, Reston, VA, pp. 1919–1928.
Ang, A. and W. Tang , 2007. The Application of Probability Ideas in Engineering Planning and Design, with a Focus on Civil and Environmental Engineering. Wiley, New York.
API (Application Programming Interface) (American Petroleum Institute) . 1987. Planning, Designing, and Constructing Fixed Offshore Platforms Is Recommended. API, Washington, DC, 17th ed.
Arnell, N. W. and S. N. Gosling , 2016. Climate change's effects on river flood risk on a worldwide scale. Clim. Change, 134, 3, 387–401. https://doi.org/10.1007/s10584-014-1084-5.
Arneson, L. A. , L. W. Zevenbergen , P. F. Lagasse , and P. E. Clopper , 2012. Bridge Scour Is Being Assessed. Circular No. 18 on Hydraulic Engineering (HEC-18). Federal Highway Administration, Washington, DC.
Ataei, N. and J. E. Padgett , 2012. Probabilistic modelling of bridge deck unseating during hurricane events. J. Bridge Eng., 18, 4, 275–286. https://doi.org/10.1061/(ASCE)BE.1943-5592.0000371.
Balomenos, G. P. and J. E. Padgett , 2018. Fragility analysis of pile-supported wharves and piers exposed to storm surge and waves. J. Waterway Port. Coast. Ocean Eng. 144, 2. https://doi.org/10.1061/(ASCE)WW.1943-5460.0000436.
Banerjee, S. and G. Ganesh Prasad , 2013. Seismic risk assessment of reinforced concrete bridges in flood-prone areas. Struct. Infrastruct. Eng. 9, 9, 952–968.

https://doi.org/10.1080/15732479.2011.649292.
Bickel, S. , M. Brückner , and T. Scheffer , 2009. Discriminative learning under covariate shift. J. Mach. Learn. Res. 10, Sep, 2137–2155.
Brekke, L. , B. L. Thrasher , E. P. Maurer , and T. Pruitt , 2013. Climate and Hydrological Projections from CMIP3 and CMIP5: Release of Downscaled CMIP5 Climate Projections, Comparison with Previous Data, and Overview of User Needs. Technical Services Center, US Department of the Interior, Bureau of Reclamation, Denver.
Breusers, H. N. C. , G. Nicollet , and H. W. Shen , 1977. Around cylindrical piers, local scour. J. Hydraul. Res. 15, 3, 211–252.
Briaud, J. L. , H. C. Chen , K. W. Kwak , S. W. Han , and F. C. K. Ting , 2001. Prediction of scour rate at bridge piers using a multiflood and multilayer approach. J. Geotech. Geoenviron. Eng. 127, 2, 114–125. https://doi.org/10.1061/(ASCE)1090-0241.
Broding, W. C. , F. W. Diederich , and P. S. Parker , 1964. Based on a reliability design criterion, structural optimization and design. J. Spacecraft Rock. 1, 1, 56–61. https://doi.org/10.2514/3.27592.
Brown, D. A. , and L. C. Reese , 1988. A Large-Scale Pile Group Subjected to Cyclic Lateral Loading and Its Behaviour. Geotechnical Engineering Center, Texas University at Austin, Austin, TX.
Buratti, N. , B. Ferracuti , and M. Savoia , 2010. Response surface with random factors for seismic fragility of reinforced concrete frames. Struct. Saf. 32, 1, 42–51. https://doi.org/10.1016/j.strusafe.2009.06.003.
Chan, C. L. , and Low, B. K. , 2012. Probabilistic study of laterally loaded piles using response surface and neural network techniques, Comput. Geotech. 43, Jun, 101–110. https://doi.org/10.1016/j.compgeo.2012.03.001.
Cook, W. , P. J. Barr , and M. W. Halling , 2013. Rate of bridge failure. J. Perform. Constr. Facil. 29, 3, 04014080. https://doi.org/10.1061/(ASCE)CF.1943-5509.0000571.
Croke, B. F. W. , F. Andrews , J. Spate , and S. M. Cuddy , 2005. Second Edition of the IHACRES User Guide. 2005/19 Technical Report. iCAM, School of Resources, Environment and Society, Australian National University Press, Canberra, Australia.
de Santana Gomes, W. J. 2019. Structural reliability analysis using adaptive artificial neural networks. ASME J. Risk Uncertain. B, 5, 4, 1–8. https://doi.org/10.1115/1.4044040.
Decker, J. B. , K. M. Rollins , and J. C. Ellsworth , 2008. Analysis and prediction of pile corrosion rates based on long-term field performance. J. Geotech. Geoenviron. Eng. 134, 3, 341–351. https://doi.org/10.1061/(ASCE)1090-0241.
Dieter, G. 1958. Engineering Design: A Materials and Processing Approach (McGraw-Hill Series in Mechanical Engineering). 3rd ed. McGraw-Hill, Boston.
Ding, L. 2019. Corrosion Behavior of H-Pile Steel in Different Soils. Doctoral dissertation, Clemson University, Clemson. https://tigerprints.clemson.edu/alldissertations/2334.
Dunnavant, T. W. and M. W. O'Neill . 1986. Analysis of vertical pile groups subjected to lateral load using design-oriented approaches. In Proceedings of the International Conference on Numerical Methods in Offshore Piling, Nantes, France. Universiti Teknologi Petronas, Paris, pp. 303–316.
ElGawady, M. A. , M. M. Abdulazeez , A. Ramadan , B. Sherstha , A. Gheni , E. Gomaa , Y. Darwish , 2019. Behavior and Restoration of Corroded Steel H-Piles Phase I (Axial Behaviour). 25-1121-0005-133-1 is the number of the report. Mid-America Transportation Center, Lincoln, Nebraska.
Ettouney, M. M. and S. Alampalli , 2011. Civil Engineering Applications and Control of Infrastructure Health. Taylor and Francis, CRC Press, Boca Raton, FL.
FEMA , 2011. The Principles and Techniques of Planning, Locating, Designing, Constructing, and Maintaining Residential Buildings in Coastal Locations Are Outlined in this Document. 4th ed. FEMA, Washington, DC.
Ferrario, E. , N. Pedroni , E. Zio , and F. Lopez-Caballero , 2017. Seismic analysis of structural systems using bootstrapped artificial neural networks. Struct. Saf. 67, July, 70–84. https://doi.org/10.1016/j.strusafe.2017.03.003.
FHWA (Federal Highway Administration) . 2016. Dataset for the National Bridge Inventory (NBI). On November 21, 2018 , I Was Able to Access This Information. https://www.fhwa.dot.gov/bridge/nbi/ascii2016.cfm.

Frost, A. J. et al., 2011. Under Australian conditions, a comparison of multi-site daily rainfall downscaling algorithms. J. Hydrol. 408, 1–2, 1–18. https://doi.org/10.1016/j.jhydrol.2011.06.021.
Galambos, T. V. , International Journal of Steel Structures, 4, 4, 223–230.
Gidaris, I. , J. E. Padgett , A. R. Barbosa , S. Chen , D. Cox , B. Webb , and A. Cerato , 2017. State-of-the-art review of multiple-hazard fragility and restoration models of highway bridges for regional risk and resilience assessment in the United States. J. Struct. Eng. 143, 3, 04016188. https://doi.org/10.1061/(ASCE)ST.1943-541X.0001672.
Haldar, S. and D. Basu , 2014. Resistance factors for laterally loaded piles in clay, geo-characterization and modeling for sustainability. In Geo-Congress 2014. ASCE, Reston, VA, pp. 3333–3342.
Hannigan, P. J. F. Rausche , G. E. Likins , B. R. Robinson , and M. L. Becker , 2016. Volume II Design and Construction of Driven Pile Foundations, Geotechnical Engineering Circular No. 12. FHWA-NHI-16-010 is the reference number. Federal Highway Administration, Washington, DC.
Hawkins, E. and R. Sutton , 2009. The potential to reduce uncertainty in regional climate predictions. Bull. Am. Meteorol. Soc. 90, 8, 1095–1108. https://doi.org/10.1175/2009BAMS2607.1.
Hazus , 2018. User Instructions for the Hazus Flood Model. FEMA, Washington, DC.
Hosmer, D. W., Jr. , S. Lemeshow , and R. X. Sturdivant , 2013. Applied Logistic Regression, volume 398, Wiley, New York.
Humphrey, G. B. , M. S. Gibbs , G. C. Dandy , and H. R. Maier , 2016. A hybrid approach to monthly streamflow forecasting: Integrating hydrological model outputs into a Bayesian artificial neural network. J. Hydrol. 540, Sep, 623–640. https://doi.org/10.1016/j.jhydrol.2016.06.026.
Hung, C.-C. , and W.-G. Yau . 2014. The response of scoured bridge piers to flood-induced stresses. Eng. Struct. 80, 241–250. https://doi.org/10.1016/j.engstruct.2014.09.009.
Jiang, S. H. , S. B. Wu , C. B. Zhou , and L. M. Zhang , Li, D. Q. , 2013. For structural reliability analysis with complex performance function, modelling multivariate distributions using Monte Carlo simulation. J. Risk Reliability Proc. Inst. Mech. Eng. 227, 2, 109–118. https://doi.org/10.1177/1748006X13476821.
Johnson, P. A. , P. E. Clopper , L. W. Zevenbergen , and P. F. Lagasse . 2015. Estimating scour in bridges: Quantifying uncertainty and reliability. J. Hydraul. Eng. 141, 7: 04015013. https://doi.org/10.1061/(ASCE)HY.1943-7900.0001017.
Kadar, I. and L. Nagy . 2017. An investigation of the coefficients of variation and types of distributions for several soil characteristics. In Proceedings of the 6th International Conference of Young Geotechnical Engineers (iYGEC6). The International Society for Soil Mechanics and Geotechnical Engineering publishes a Journal Called Soil Mechanics and Geotechnical Engineering.
Karpathy, A. and L. Fei-Fei , 2015. Deep visual-semantic alignments for image description generation. In IEEE Conference on Computer Vision and Pattern Recognition. IEEE, New York, pp. 3128–3137.
Khandel, O. and M. Soliman , 2019. An integrated methodology for estimating the impact of climate change on the risk of bridge failure due to flooding and scour caused by flooding. J. Bridge Eng. 24, 9, 04019090. https://doi.org/10.1061/(ASCE)BE.1943-5592.0001473.
Kim, H. , S. H. Sim , J. Lee , Y. J. Lee , and J. M. Kim . 2017. Analysis of flood fragility in bridges with numerous collapse modes. Adv. Mech. Eng. 9, 3, 168781401769641. https://doi.org/10.1177/1687814017696415.
Kingma, D. P. and J. Ba , 2014. Adam: A Stochastic Optimization Approach. The following is a preprint that was submitted on December 22, 2014. http://arxiv.org/abs/1412.6980.
Kroetz, H. M. , R. K. Tessari , and A. T. Beck , 2017. Performance of global metamodeling techniques in the solution of structural reliability problems. Adv. Eng. Software 114, 394–404. https://doi.org/10.1016/j.advengsoft.2017.08.001.
Kucera, V. and E. Mattsson , 1987. Atmospheric corrosion. F. B. Mansfeld’s Corrosion Mechanics, Chap. 5. Marcel Dekker, New York, 211–284.
Liaw, A. and M. Wiener , 2002. Classification and regression by random forest, R News, 2, 3, 18–22.
Mann, N. R. , N. D. Singpurwalla , and R. E. Schafer , 1974. Methods for Analysing Reliability and Life Data Statistically. Wiley, New York.
MathWorks . 2016. SIMULINK for Technical Computing. http://www.mathworks.com.

Maurer, E. P. , H. G. Hidalgo , T. Das , M. D. Dettinger , and D. R. Cayan , 2010. The value of daily large-scale climate data in assessing climate change consequences on daily streamflow in California. Hydrol. Earth Syst. Sci. 14, 6, 1125–1138. https://doi.org/10.5194/hess-14-1125-2010.
Mazzoni, S. , F. McKenna , M. H. Scott , and G. L. Fenves , 2006. Manual for the OpenSees Command Language. Pacific Earthquake Engineering Research (PEER) Center, 264. Univ. of California, Berkeley, CA.
McPherson, R. 2016. Impacts of Climate Change on Flows in the Red River Basin. South Centeral Climate Science Center, Norman, OK.
Melchers, R. E. , and A. T. Beck . 2018. Structural Reliability Analysis and Prediction. 3rd ed. Wiley, New York.
Mosher, R. L. 1984. Load Transfer Criteria for Numerical Analysis of Axial Loaded Piles in Sand. US Army Engineering and Waterways Experimental Station, Automatic Data Processing Center, Vicksburg, MI.
Moss, R. H. , et al. 2010. The next generation of scenarios for climate change research and assessment. Nature 463, 7282, 747. https://doi.org/10.1038/nature08823.
Muis, S. , B. Güneralp , B. Jongman , J. C. Aerts , and P. J. Ward , 2015. Flood risk and adaptation strategies under climate change and urban expansion: A probabilistic analysis using global data. Sci. Total Environ. 538, 445–457. https://doi.org/10.1016/j.scitotenv.2015.08.068.
Neumann, J. E. , et al. 2015. Climate change risks to US infrastructure: Impacts on roads, bridges, coastal development, and urban drainage. Clim. Change 131, 1, 97–109. https://doi.org/10.1007/s10584-013-1037-4.
NOAA (National Oceanic and Atmospheric Administration) , 2018. Climate Data Online: Dataset Discovery. Accessed November 10, 2018 . http://www.noaa.gov/.
O'Neill, M. W. 1983. Group action in offshore piles. In Geotechnical Practice in Offshore Engineering. ASCE, Reston, VA. pp. 25–64.
Ohsaki, Y. 1982. Corrosion of steel piles driven in soil deposits. Soils Found. 22, 3, 57–76. https://doi.org/10.3208/sandf1972.22.3_57.
Park, J. , and P. Towashiraporn . 2014. Rapid seismic damage assessment of railway bridges using the response-surface statistical model. Struct. Saf. 47, 1–12. https://doi.org/10.1016/j.strusafe.2013.10.001.
Prasad, G. and S. Banerjee , The effect of flood-induced scour on the seismic vulnerability of bridges. J. Earthquake Eng. 17, 6, 803–828. https://doi.org/10.1080/13632469.2013.771593.
Rampasek, L. , and A. Goldenberg . 2016. Tensorflow: Biology's gateway to deep learning? Cell Syst. 2, 1, 12–14. https://doi.org/10.1016/j.cels.2016.01.009.
Reese, C. L. , T. S. Wang , A. J. Arrellaga , J. Hendrix , and L. Vasquez . 2016. A Group for the Analysis of a Group of Piles Subjected to Vertical and Lateral Loading (User's Manual). Ensoft, Austin, TX.
Samadi, S. Z. , G. Sagareswar , and M. Tajiki , 2010. Comparison of general circulation models: Methodology for selecting the best GCM in Kermanshah Synoptic Station, Iran. Int. J. Global Warm. 2, 4, 347–365. https://doi.org/10.1504/IJGW.2010.037590.
Schlosser, F. , and M. Bastick , 1991. Reinforced earth. In Foundation Engineering Handbook, H.-Y. Fang (ed.), Van Nostrand Reinhold, New York, pp. 778–786.
Sheffield, J. , A. P. Barrett , B. Colle , D. Nelun Fernando , R. Fu , K. L. Geil , Q. Hu , J. Kinter , S. Kumar , and B. Langenbrunner , 2013. North American climate in CMIP5 experiments. Part I: Evaluation of historical simulations of continental and regional climatology. J. Clim. 26, 23, 9209–9245. https://doi.org/10.1175/JCLI-D-12-00592.1.
Shelhamer, E. , J. Donahue , S. Karayev , J. Long , R. Girshick , S. Guadarrama , and T. Darrell . Jia, Y. , 2014. Caffe: Fast feature embedding with convolutional architecture. Proceedings of the 22nd ACM International Conference on Multimedia. Association for Computing Machinery, New York, pp. 675–678.
Shen, C. 2018. A transdisciplinary review of deep learning research and its relevance for water resources scientists. Water Resour. Res. 54, 11, 8558–8593. https://doi.org/10.1029/2018WR022643.
Shooman, M. L. 1968. Probabilistic Reliability: An Engineering Approach. McGraw-Hill, New York.
Shrestha, S. , T. V. Bach , and V. P. Pandey , 2016. Climate change impacts on groundwater resources in Mekong Delta under representative concentration pathways (RCPs) scenarios.

Environ. Sci. Pol. 61, 1–13. https://doi.org/10.1016/j.envsci.2016.03.010.
Singh, T. , M. Pal , and V. K. Arora , 2019. Modeling oblique load carrying capacity of batter pile groups using neural network, random forest regression and M5 model tree. Front. Struct. Civ. Eng. 13, 3, 674–685. https://doi.org/10.1007/s11709-018-0505-3.
Song, L. K. , G. C. Bai , C. W. Fei , and J. Wen , 2018. Reliability-based fatigue life prediction for complex structure with time-varying surrogate modeling. Adv. Mater. Sci. Eng. 2018, 3469465. https://doi.org/10.1155/2018/3469465.
Sun, W. , B. Zheng , and W. Qian , 2016. Computer aided lung cancer diagnosis with deep learning algorithms. In Vol. 9785 of Medical Imaging 2016: Computer-Aided Diagnosis, 97850Z. International Society for Optics and Photonics, Bellingham, WA.
Taylor, K. E. , R. J. Stouffer , and G. A. Meehl , 2012. An overview of CMIP5 and the experiment design. Bull. Am. Meteorol. Soc. 93, 4, 485–498. https://doi.org/10.1175/BAMS-D-11-00094.1.
The Joint Committee on Structural Safety (JCSS) , 2001. Model Code for Probabilistic. On the 10th of November, 2019, I checked my email. https://www.jcss-lc.org/jcss-probabilistic-model-code/.
Tian, Y. , K. Pei , S. Jana , and B. Ray , 2018. Deeptest: Automated testing of deep-neural-network-driven autonomous cars. In Proceeding of 40th International Conference on Software Engineering. Association for Computing Machinery, New York, pp. 303–314.
Turner, D. 2016. Fragility Assessment of Bridge Superstructures under Hydrodynamic Forces. Master's thesis, Dept. of Civil and Environmental Engineering, Colorado State Univ.
USGS. 2018. National Water Information Service. Accessed October 1, 2018 . https://waterdata.usgs.gov/nwis.
Vijayvergiya, V. N. 1977. Load-movement characteristics of piles. In Proceedings of Ports '77 Conference. ASCE, New York.
Wang, N. 2010. Reliability-Based Condition Assessment of Existing Highway Bridges. Ph.D. dissertation, School of Civil and Environmental Engineering, Georgia Institute of Technology.
Wang, S. S. , L. Zhao , J. H. Yoon , P. Klotzbach , and R. R. Gillies . 2018a. Quantitative attribution of climate effects on Hurricane Harvey's extreme rainfall in Texas. Environ. Res. Lett. 13, 5, 054014. https://doi.org/10.1088/1748-9326/aabb85.
Wang, Z. , J. E. Padgett , and L. Duenas-Osorio . 2014b. Risk-consistent calibration of load factors for the design of reinforced concrete bridges under the combined effects of earthquake and scour hazards. Eng. Struct. 79, 86–95. https://doi.org/10.1016/j.engstruct.2014.07.005.
Wang, Z. , L. Duenas-Osorio , and J. E. Padgett . 2014a. Influence of scour effects on the seismic response of reinforced concrete bridges. Eng. Struct. 76, 202–214. https://doi.org/10.1016/j.engstruct.2014.06.026.
Wang, Z. , N. Pedroni , I. Zentner , and E. Zio , 2018b. Seismic fragility analysis with artificial neural networks: Application to nuclear power plant equipment. Eng. Struct. 162, May, 213–225. https://doi.org/10.1016/j.engstruct.2018.02.024.
Weigel, A. P. , R. Knutti , M. A. Liniger , and C. Appenzeller , 2010. Risks of model weighting in multimodel climate projections. J. Clim. 23, 15, 4175–4191. https://doi.org/10.1175/2010JCLI3594.1.
Westra, S. , M. Thyer , M. Leonard , D. Kavetski , and M. Lambert , 2014. A strategy for diagnosing and interpreting hydrological model nonstationarity. Water Resour. Res. 50, 6, 5090–5113. https://doi.org/10.1002/2013WR014719.
Winsemius, H. C. , et al. 2016. Global drivers of future river flood risk. Nat. Clim. Change 6, 4, 381. https://doi.org/10.1038/nclimate2893.
Yip, S. , C. A. Ferro , D. B. Stephenson , and E. Hawkins . 2011. A simple, coherent framework for partitioning uncertainty in climate predictions. J. Clim. 24, 17, 4634–4643. https://doi.org/10.1175/2011JCLI4085.1.
Zhang, Z. , P. Klein , and K. Friedrich , 2002. Dynamic mechanical properties of PTFE based short carbon fibre reinforced composites: Experiment and artificial neural network prediction. Compos. Sci. Technol. 62, 7–8, 1001–1009. https://doi.org/10.1016/S0266-3538(02)00036-2.

Software Vulnerability Analysis

Wu, S. , et al. Software vulnerability analysis technology progress. Journal of Tsinghua University (Natural Science) 10 (2012): 1309–1319.

Balzarotti, D. , et al. Saner: Composing static and dynamic analysis to validate sanitization in web applications. Security and Privacy, 2008. SP 2008. IEEE Symposium on IEEE, 2008.

Sebastiani, F. Machine learning in automated text categorization. ACM Computing Surveys (CSUR) 34.1 (2002): 1–47.

Shabtai, A. , et al. Detection of malicious code by applying machine learning classifiers on static features: A state-of-the-art survey. Information Security Technical Report 14.1 (2009): 16–29.

Yamaguchi, F., L. Felix , and K. Rieck . Vulnerability extrapolation: Assisted discovery of vulnerabilities using machine learning. Proceedings of the 5th USENIX Conference on Offensive Technologies. USENIX Association, 2011.

Yamaguchi, F., M. Lottmann , and K. Rieck . Generalized vulnerability extrapolation using abstract syntax trees. Proceedings of the 28th Annual Computer Security Applications Conference. ACM, 2012.

Yamaguchi, F. , et al. Chucky: Exposing missing checks in source code for vulnerability discovery. Proceedings of the 2013 ACM SIGSAC conference on Computer & Communications Security. ACM, 2013.

Yamaguchi, F. , et al. Modeling and discovering vulnerabilities with code property graphs. Security and Privacy (SP), 2014 IEEE Symposium on. IEEE, 2014.

Yamaguchi, F. , et al. Automatic Inference of Search Patterns for Taint-Style Vulnerabilities, IEEE Symposium on Security and Privacy, 797–812. IEEE, 2015.

Zhang, S., D. Caragea , and X. Ou . An empirical study on using the national vulnerability database to predict software vulnerabilities. Database and Expert Systems Applications. Springer, Berlin Heidelberg, 2011.

Grieco, G. , et al. Toward Large-Scale Vulnerability Discovery Using Machine Learning. Proceedings of the Sixth ACM on Conference on Data and Application Security and Privacy, CODASPY 2016. ACM, New Orleans, LA, USA, March 9-11, 2016 : 85–96.

Wang, Y., Y. Wang , and J. Ren . Software vulnerabilities detection using rapid density-based clustering. Journal of Computational Information Systems 8.14(2011): 3295–3302.

Cheng, H. , et al. Identifying bug signatures using discriminative graph mining. Proceedings of the Eighteenth International Symposium on Software Testing and Analysis. ACM, 2009.

Wijayasekara, D. , et al. Mining bug databases for unidentified software vulnerabilities. Human System Interactions (HSI), 2012 5th International Conference on. IEEE, 2012.

Neuhaus, S. , et al. Predicting vulnerable software components. Proceedings of the 14th ACM Conference on Computer and Communications Security. ACM, 2007.

Mokhov, S., J. Paquet , and M. Debbabi . MARFCAT: Fast code analysis for defects and vulnerabilities. Software Analytics (SWAN), 2015 IEEE 1st International Workshop on. IEEE, 2015.

Almorsy, M., J. Grundy , and A. S. Ibrahim . Supporting automated vulnerability analysis using formalized vulnerability signatures. Proceedings of the 27th IEEE/ACM International Conference on Automated Software Engineering. ACM, 2012.

Medeiros, I., N. F. Neves , and M. Correia . Automatic detection and correction of web application vulnerabilities using data mining to predict false positives. Proceedings of the 23rd International Conference on World Wide Web. ACM, 2014.

Shar, L. K. , and H. B. Kuan Tan . Predicting SQL injection and cross site scripting vulnerabilities through mining input sanitization patterns. Information and Software Technology 55.10(2013): 1767–1780.

Liu, H. , et al. PF-miner: A new paired functions mining method for Android kernel in error paths. Computer Software and Applications Conference (COMPSAC), 2014 IEEE 38th Annual. IEEE, 2014.

Liu, H.-Q. , et al. BP-Miner: mining paired functions from the binary code of drivers for error handling. 2014 21st Asia-Pacific Software Engineering Conference 58(2014): 5A4–5A9.