

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/342491855>

# 01 – Hybrid Mobile Learning – Book Chapter 2 – Me and Addisu

Conference Paper · November 2018

---

CITATIONS

0

READS

397

6 authors, including:



Asrat M. Beyene  
Addis Ababa Science and Technology University

19 PUBLICATIONS 8 CITATIONS

[SEE PROFILE](#)



Sanjay Misra  
Østfold University College

771 PUBLICATIONS 6,701 CITATIONS

[SEE PROFILE](#)



Robertas Damaševičius  
620 PUBLICATIONS 10,437 CITATIONS

[SEE PROFILE](#)



Ravin Ahuja  
Sri Vishwakarma Skill University Gurugram India

95 PUBLICATIONS 374 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Achieving Sustainable Development Goals through ICT/Software Engineering [View project](#)



Representing Contextual Relations with Sanskrit Word Embeddings [View project](#)

Shampa Chakraverty · Anil Goel  
Sanjay Misra *Editors*

---

# Towards Extensible and Adaptable Methods in Computing

# Towards Extensible and Adaptable Methods in Computing

Shampa Chakraverty · Anil Goel  
Sanjay Misra  
Editors

# Towards Extensible and Adaptable Methods in Computing



Springer

*Editors*

Shampa Chakraverty  
Department of Computer Engineering,  
Netaji Subhas Institute of Technology  
University of Delhi  
New Delhi, India

Anil Goel  
SAP Canada  
Waterloo, ON, Canada

Sanjay Misra   
Department of Electrical and Information  
Engineering  
Covenant University  
Ota, Nigeria

ISBN 978-981-13-2347-8      ISBN 978-981-13-2348-5 (eBook)  
<https://doi.org/10.1007/978-981-13-2348-5>

Library of Congress Control Number: 2018952626

© Springer Nature Singapore Pte Ltd. 2018

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors, and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Singapore Pte Ltd.  
The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721,  
Singapore

# Preface

Extensible and adaptable computing refers to the array of methods and techniques that systematically tackle the future growth of systems by responding proactively to change. This mandates a synergistic coordination amongst various facets of computing. Agile software development is a significant driver towards this paradigm shift and has indeed become the industry de facto standard. The ever-evolving data is another component that requires new methods of storage, transmission and processing. The Web which hosts almost all applications and data is potent with latent intelligence, ready to be mined and utilized for extending the applications and making them respond seamlessly to changing contexts. Innovative machine learning tools enable us to extricate patterns of information from repositories and adapt to changes in real time.

Our journey towards extensible and adaptable methods in computing investigates various challenges in the above areas. Accordingly, this book is divided into the following parts: agile software development, data management, machine learning, Web intelligence and computing in education. The first four domains of computing work together in mutually complementary ways to build automated systems that scale well to meet the demands of changing context and requirements. The fifth part highlights an important application of adaptable computing that enables lifelong learning for all. The concept of this book emanated from the deliberations of the international conference *Towards Extensible and Adaptable Methods in Computing* that took place during March 26–28, 2018, in New Delhi. The top research papers presented in it were selected to prepare its chapters.

The first part on *agile software development* addresses some of the important challenges in developing quality software within collaborative environments such as risk management, test case prioritization, open-source software reliability and predicting software change proneness. The second part on *data management* presents elegant solutions for cost-efficient storage of data, transmitting data securely and processing data in specific applications such as health care. The third part on *machine learning* showcases innovative algorithms and applications including portfolio optimization, disruption classification and outlier detection. The fourth part on *Web intelligence* covers emerging Web applications in dynamic social

contexts including metaphor detection in natural language processing, language identification and sentiment analysis. It also underscores Web security issues such as fraud detection and trust and reputation systems. The fifth part on *computing in education* presents computer-aided pedagogical methods that adapt and personalize to each learner, thus overcoming the constraints of traditional methods.

We wish to thank our section editors Ritu Sibal, Anand Gupta, Sushma Nagpal, Swati Aggarwal and Pinaki Chakraborty for their invaluable contribution. We are grateful to our publishing coordinators Krati Srivastav, Antony Raj Joseph, Suvira Srivastava, Sona Chahal and Nidhi Chandhoke for their support and encouragement. It is through the dedicated efforts of all the authors and reviewers that this book has been compiled, and we are deeply thankful to all of them.

New Delhi, India  
Ota, Nigeria  
Waterloo, Canada  
June 2018

Prof. Shampa Chakraverty  
Prof. Sanjay Misra  
Dr. Anil Goel

# Contents

## **Part I Agile Software Development**

Ritu Sibal

<b>Risk Assessment Framework: ADRIM Process Model for Global Software Development .....</b>	3
Chamundeswari Arumugam, Sriraghav Kameswaran and Baskaran Kaliamourthy	
1 Introduction .....	3
2 Literature Survey .....	4
3 Research Methodology .....	5
3.1 Mitigation Strategy for Various Risks .....	5
3.2 ADRIM Process Model .....	6
4 Results and Discussion .....	8
5 Conclusion .....	10
References .....	11
<b>An Extended Test Case Prioritization Technique Using Script and Linguistic Parameters in a Distributed Agile Environment .....</b>	13
Anita and Naresh Chauhan	
1 Introduction .....	13
2 Literature Survey .....	14
3 Modified User Story .....	15
4 Sprint—Story Prioritization .....	15
5 Linguistic Parameters-Noun and Verb .....	17
5.1 Agile Change Management (ACM)—User Story .....	17
6 Conclusion .....	23
References .....	26

<b>AutoJet: Web Application Automation Tool . . . . .</b>	<b>27</b>
Sheetika Kapoor and Kalpana Sagar	
1 Introduction . . . . .	27
2 Related Work . . . . .	29
3 Methodology . . . . .	30
3.1 Context of Proposal . . . . .	30
3.2 Participants . . . . .	31
3.3 The AutoJet . . . . .	32
3.4 Adaption of Autojet in Agile Methodology . . . . .	40
4 Conclusion and Future Scope . . . . .	41
References . . . . .	41
<b>Prioritization of User Story Acceptance Tests in Agile Software Development Using Meta-Heuristic Techniques and Comparative Analysis . . . . .</b>	<b>43</b>
Ritu Sibal, Preeti Kaur and Chayanika Sharma	
1 Introduction . . . . .	43
2 Basic Concepts . . . . .	44
2.1 Given-When-Then Format . . . . .	44
2.2 Overview of Meta-Heuristic Algorithms . . . . .	45
3 Proposed Approach . . . . .	47
3.1 Case Study: Bank Management System . . . . .	49
3.2 User Story: Account holder withdraws cash . . . . .	50
3.3 Acceptance Criteria . . . . .	50
4 Conclusion and Future Work . . . . .	54
References . . . . .	54
<b>Software Reliability Assessment Using Deep Learning Technique . . . . .</b>	<b>57</b>
Suyash Shukla, Ranjan Kumar Behera, Sanjay Misra and Santanu Kumar Rath	
1 Introduction . . . . .	57
2 Related Work . . . . .	58
3 Bug Tracking System . . . . .	59
4 Identification of Critical Fault Based on Neural Network . . . . .	59
5 Identification of Critical Fault Based on Deep Learning . . . . .	61
5.1 Analysis of Dataset . . . . .	62
5.2 Creation of the Model . . . . .	62
5.3 Compilation of the Model . . . . .	63
5.4 Fitting the Model . . . . .	63
5.5 Evaluation of the Model . . . . .	64
6 Dataset . . . . .	64
7 Analysis of Result . . . . .	64
8 Conclusion . . . . .	67
References . . . . .	67

<b>Empirical Validation of OO Metrics and Machine Learning Algorithms for Software Change Proneness Prediction . . . . .</b>	69
Anushree Agrawal and Rakesh Kumar Singh	
1 Introduction . . . . .	69
2 Related Work . . . . .	70
3 Approach . . . . .	71
3.1 Independent Variables . . . . .	71
3.2 Dependent Variable . . . . .	71
3.3 Prediction Model . . . . .	71
3.4 Validation Method . . . . .	73
4 Empirical Data Collection . . . . .	73
4.1 Descriptive Statistics . . . . .	74
5 Result Analysis . . . . .	75
5.1 Univariate LR Analysis Results . . . . .	75
5.2 Model Evaluation Using ROC Curve . . . . .	80
5.3 Friedman Test Result . . . . .	81
6 Conclusion . . . . .	83
References . . . . .	83

## Part II Data Management

Anand Gupta

<b>Extending Database Cache Using SSDs . . . . .</b>	89
Prateek Agarwal and Vaibhav Nalawade	
1 Introduction . . . . .	89
2 Configuring NV Cache . . . . .	90
3 NV Cache Design . . . . .	90
3.1 Buffer Cache Description . . . . .	90
3.2 Data Layout . . . . .	91
3.3 Page Search . . . . .	92
3.4 Page Writes . . . . .	92
3.5 Lazy Cleaner Task . . . . .	93
3.6 Page Eviction . . . . .	94
4 Performance Results . . . . .	94
4.1 Benchmark . . . . .	95
4.2 Results . . . . .	95
5 Related Work . . . . .	96
6 Enhancements and Future Work . . . . .	97
7 Conclusion . . . . .	97
References . . . . .	98

<b>Cloud-Based Healthcare Monitoring System Using Storm and Kafka . . . . .</b>	99
N. Sudhakar Yadav, B. Eswara Reddy and K. G. Srinivasa	
1 Introduction . . . . .	99
2 Related Work . . . . .	100
3 Proposed System . . . . .	101
3.1 Web Portal . . . . .	103
3.2 Data Adapter and Integrator . . . . .	103
3.3 Apache Kafka . . . . .	103
3.4 Storm . . . . .	104
4 Experiment and Results . . . . .	104
5 Conclusion . . . . .	106
References . . . . .	106
<b>Honeynet Data Analysis and Distributed SSH Brute-Force Attacks . . . . .</b>	107
Gokul Kannan Sadasivam, Chittaranjan Hota and Bhojan Anand	
1 Introduction . . . . .	107
2 Related Work . . . . .	108
3 Honeynet Architecture . . . . .	109
4 General Characteristics . . . . .	110
4.1 Source of the Attacks . . . . .	111
5 Secure Shell (SSH) Traffic Analysis . . . . .	113
6 Conclusion . . . . .	117
References . . . . .	117
<b>Efficient Data Transmission in WSN: Techniques and Future Challenges . . . . .</b>	119
Nishi Gupta, Shikha Gupta and Satbir Jain	
1 Introduction . . . . .	119
2 Routing in WSN . . . . .	120
2.1 Classification . . . . .	121
2.2 Advantages . . . . .	123
3 Routing Techniques . . . . .	123
4 Future Scope . . . . .	127
5 Conclusion . . . . .	127
References . . . . .	128
<b>A Study of Epidemic Spreading and Rumor Spreading over Complex Networks . . . . .</b>	131
Prem Kumar, Puneet Verma and Anurag Singh	
1 Introduction . . . . .	131
1.1 Random Networks . . . . .	132
1.2 Scale-free Network . . . . .	132
1.3 Properties . . . . .	132
1.4 Characteristics of Some Real Network Data Available and Widely Used . . . . .	133

2	Analysis of Epidemic . . . . .	134
2.1	SIR Model . . . . .	134
2.2	SIS Model . . . . .	135
3	Analysis of Epidemic Using SIR Model . . . . .	136
3.1	Impact of Epidemic: Curves for the Variations of Final Number of People Recovered Versus $\lambda$ (Rate of Infection)/ $\beta$ (Rate of Recovery). . . . .	137
3.2	Curves for the Variations of Total Timestamps Survived Versus Variations Other Properties . . . . .	138
4	Analysis of Rumor Spreading . . . . .	139
5	Comparison of Plots for the Random Networks and Scale-free Network . . . . .	141
6	Conclusion . . . . .	141
	References . . . . .	143
	<b>Medical Alert System Using Social Data . . . . .</b>	145
	Kumar Abhishek, M. P. Singh, Prakhar Shrivastav and Suraj Thakre	
1	Introduction . . . . .	145
2	Related Works . . . . .	146
3	System Implementation . . . . .	146
3.1	Data Collection . . . . .	147
3.2	Data Processing . . . . .	148
3.3	Structured Tweets . . . . .	150
3.4	Data Processing . . . . .	151
4	Conclusion . . . . .	153
	References . . . . .	153

## Part III Machine Learning

Swati Aggarwal

	<b>A Novel Framework for Portfolio Optimization Based on Modified Simulated Annealing Algorithm Using ANN, RBFN, and ABC Algorithms . . . . .</b>	157
	Chanchal Kumar, M. N. Doja and Mirza Allim Baig	
1	Introduction . . . . .	157
2	Background . . . . .	159
2.1	Portfolio Optimization . . . . .	159
2.2	Problem Definition . . . . .	159
2.3	Module 1. (Algorithm 1) Modified Simulated Annealing Scheme . . . . .	161
2.4	Module 2. Definition of Function frozen() . . . . .	164
2.5	Module 3. Description of Steps Used for Finding the Modified Value of the Parameter Step. . . . .	164
2.6	Module 4. Definition of random_move() Function . . . . .	165

2.7	Module 5. Computing the Radius Using Radial Basis Function Network (RBFN) with the Different Values of Lower Bound (lb) and Upper Bound (ub) . . . . .	165
2.8	Module 6. Overview of Improved ABC Algorithm . . . . .	167
2.9	Module 7. Overview of Back-propagation Network . . . . .	167
3	Empirical Results . . . . .	170
4	Conclusion . . . . .	174
	References . . . . .	177

**A Proposed Method for Disruption Classification in Tokamak Using Convolutional Neural Network . . . . .** 179

Priyanka Sharma, Swati Jain, Vaibhav Jain, Sutapa Ranjan, R. Manchanda, Daniel Raju, J. Ghosh and R. L. Tanna

1	Introduction . . . . .	179
1.1	Motivation . . . . .	181
1.2	Stages and Types of Disruption . . . . .	181
2	Aditya Tokamak and It's Diagnostics . . . . .	183
3	Disruption Classification and Prediction Techniques . . . . .	184
4	Deep Learning Techniques and CNN . . . . .	185
5	Proposed Work . . . . .	189
5.1	Database Preparation . . . . .	190
5.2	Disruption Prediction Using CNN (Proposed Method) . . . . .	190
6	Conclusion . . . . .	191
	References . . . . .	192

**Comparative Evaluation of Machine Learning Algorithms for Network Intrusion Detection Using Weka . . . . .** 195

Nureni Ayofe Azeez, Obinna Justin Asuzu, Sanjay Misra, Adewole Adewumi, Ravin Ahuja and Rytis Maskeliunas

1	Introduction . . . . .	195
2	Literature Review . . . . .	196
3	Methodology . . . . .	198
3.1	Data Exploration . . . . .	198
3.2	Classification Models . . . . .	198
3.3	Data Analysis . . . . .	201
4	Implementation . . . . .	201
4.1	Cleaning up the Data . . . . .	201
4.2	Feature Engineering . . . . .	202
5	Results and Discussion . . . . .	205
6	Conclusion . . . . .	207
	References . . . . .	207

<b>Super-Intelligent Machine Operations in Twenty-First-Century Manufacturing Industries: A Boost or Doom to Political and Human Development? . . . . .</b>	209
I. A. P. Wogu, S. Misra, P. A. Assibong, S. O. Ogiri, R. Damasevicius and R. Maskeliunas	
1 Introduction . . . . .	209
1.1 The Problem . . . . .	210
1.2 Objectives of the Study . . . . .	211
1.3 Methodology and Theoretical Foundations for the Study . . . . .	211
2 The Advent of Artificial Intelligence . . . . .	211
2.1 Artificial Intelligence (AI) . . . . .	212
3 AI Technology and Manufacturing Industries in the Twenty-First Century . . . . .	213
3.1 AI Machines in Today's Manufacturing Industries . . . . .	213
3.2 Existential Threats Issues in Today's Manufacturing Industries (MI) . . . . .	215
3.3 AI, Karl Marx Alienation Theory and Human Development . . . . .	217
3.4 Analysis of Selected Studies on the Impact of AI in MI . . . . .	218
4 Super-Intelligent Machines and Human Development in the Twenty-First Century: Boost or Doom? . . . . .	219
4.1 A Critical Review of Some Studies Considered for This Research . . . . .	219
5 Conclusion . . . . .	220
5.1 Summary of Findings . . . . .	220
5.2 Conclusion . . . . .	221
5.3 Recommendation . . . . .	221
References . . . . .	222
<b>Exploring Ensembles for Unsupervised Outlier Detection: An Empirical Analysis . . . . .</b>	225
Akanksha Mukhiya and Rajeev Kumar	
1 Introduction . . . . .	225
2 Member Selection . . . . .	226
2.1 Assessing Accuracy and Diversity of Detectors . . . . .	226
2.2 Selection Models . . . . .	227
2.3 Issues and Challenges of Member Selection . . . . .	228
3 Combination . . . . .	228
3.1 Combination of Outlier Ranks . . . . .	230
3.2 Combination of Outlier Scores . . . . .	230
3.3 Issues and Challenges of Combination . . . . .	232
4 Significance of Outlier Detection Ensembles . . . . .	232
4.1 Dataset and Ensemble Description . . . . .	233
4.2 Observations . . . . .	234

5 Conclusion . . . . .	236
References . . . . .	236

## **Part IV Web Intelligence**

Sushma Nagpal

<b>Effect of Classifiers on Type-III Metaphor Detection . . . . .</b>	241
Sunny Rai, Shampa Chakraverty and Ayush Garg	
1 Introduction . . . . .	241
2 Related Work . . . . .	243
3 Effect of Classifiers on Type-III Metaphor Detection. . . . .	244
3.1 Datasets . . . . .	244
3.2 Comparison . . . . .	244
4 Conclusion . . . . .	247
References . . . . .	247
<b>Multi-class Classification of Sentiments in Hindi Sentences Based on Intensities . . . . .</b>	251
Kanika Garg and D. K. Lobiyal	
1 Introduction . . . . .	251
2 Related Work . . . . .	252
3 Feature Engineering . . . . .	254
3.1 Weighting Schemes . . . . .	254
3.2 Proposed Term Weighting Scheme . . . . .	256
3.3 Features Used . . . . .	257
4 Method . . . . .	258
4.1 Experimental Set-up . . . . .	259
5 Results and Discussions . . . . .	261
6 Conclusion and Future Work . . . . .	264
Appendix . . . . .	265
References . . . . .	265
<b>Language Identification for Hindi Language Transliterated Text in Roman Script Using Generative Adversarial Networks . . . . .</b>	267
Deepak Kumar Sharma, Anurag Singh and Abhishek Saroha	
1 Introduction . . . . .	267
2 Related Works . . . . .	268
3 Methodology . . . . .	270
3.1 Language Identification . . . . .	270
3.2 Feature Extraction . . . . .	270
3.3 Review of Autoencoders and GAN . . . . .	272
3.4 Propagation in Model . . . . .	274
3.5 Training the Model . . . . .	275

4	Results . . . . .	277
4.1	Training Results . . . . .	277
4.2	Comparative Results . . . . .	277
5	Conclusion . . . . .	279
	References . . . . .	279
<b>An Improved Similarity Measure to Alleviate Sparsity Problem in Context-Aware Recommender Systems . . . . .</b>		281
Veer Sain Dixit and Parul Jain		
1	Introduction . . . . .	281
2	Related Work . . . . .	283
3	The Proposal . . . . .	283
3.1	Proposed Similarity Measure for Collaborative Filtering (O-CHSM and E-CHSM) . . . . .	284
3.2	The Formalization of Similarity Measures . . . . .	285
3.3	Context-Aware Recommendation Unit Based on Contextual Hybrid Similarity Measure . . . . .	288
3.4	Group Recommendation Unit . . . . .	289
4	Experimental Evaluation . . . . .	290
4.1	Data Preparation and Evaluation Metrics . . . . .	290
4.2	Compared Methods . . . . .	291
4.3	Results and Analysis . . . . .	291
5	Conclusions and Future Work . . . . .	294
	References . . . . .	294
<b>Trust and Reputation-Based Model to Prevent Denial-of-Service Attacks in Mobile Agent System . . . . .</b>		297
Praveen Mittal and Manas Kumar Mishra		
1	Introduction . . . . .	297
1.1	A Mobile Agent Kit . . . . .	297
1.2	Advantages of Mobile Agents . . . . .	298
1.3	Mobile Agent Applications . . . . .	299
1.4	Security Issues . . . . .	299
2	Related Work . . . . .	300
3	Proposed Model . . . . .	301
4	Results and Analysis . . . . .	304
5	Conclusion . . . . .	306
	References . . . . .	307
<b>Fraud Detection in Online Transactions Using Supervised Learning Techniques . . . . .</b>		309
Akshi Kumar and Garima Gupta		
1	Introduction . . . . .	309
2	Related Work . . . . .	310

3 Models . . . . .	311
4 System Architecture . . . . .	311
4.1 Dataset . . . . .	315
5 Results and Analysis . . . . .	315
5.1 Results . . . . .	315
6 Conclusion . . . . .	318
References . . . . .	320

## **Part V Computing in Education**

Pinaki Chakrabarty

<b>Real-Time Printed Text Reader for Visually Impaired . . . . .</b>	327
Ashutosh Dadhich and Kamlesh Dutta	
1 Introduction . . . . .	327
2 Related Work . . . . .	328
3 Proposed System . . . . .	331
3.1 System Overview . . . . .	332
3.2 System Functionalities . . . . .	334
4 Evaluation and Result Analysis . . . . .	335
5 Conclusion . . . . .	337
References . . . . .	337
<b>Intelligent Task Assignment in a Crowdsourcing Platform . . . . .</b>	339
A. Vijayalakshmi and Chittaranjan Hota	
1 Introduction . . . . .	339
2 Related Work . . . . .	341
3 Trust Algorithm for Task Assignment . . . . .	342
3.1 Calculation of Belief . . . . .	342
3.2 Calculation of Knowledge . . . . .	343
4 Experimental Result and Analysis . . . . .	345
5 Conclusions . . . . .	348
References . . . . .	348
<b>Teaching Algorithms Using an Android Application . . . . .</b>	351
Dipika Jain and Pawan Kumar	
1 Introduction . . . . .	351
2 Experimental Studies . . . . .	353
2.1 Literature Survey . . . . .	354
2.2 Experimental Results . . . . .	355
3 Review and Suggestion . . . . .	357
4 Conclusion . . . . .	360
References . . . . .	361

<b>Keyword Extraction Using Graph Centrality and WordNet . . . . .</b>	363
Chhavi Sharma, Minni Jain and Ayush Aggarwal	
1 Introduction . . . . .	363
2 Related Work . . . . .	364
3 Proposed Scheme . . . . .	365
3.1 Centrality Measures . . . . .	367
3.2 Algorithm . . . . .	368
4 Experimentation and Results . . . . .	369
5 Observation and Future Work . . . . .	370
References . . . . .	371
<b>Hybrid Mobile Learning Architecture for Higher Education . . . . .</b>	373
Asrat Mulatu, Addisu Ambessa, Sanjay Misra, Adewole Adewumi, Robertas Damaševičius and Ravin Ahuja	
1 Introduction . . . . .	373
2 Related Works . . . . .	375
3 The Proposed Architecture . . . . .	377
4 Prototype Implementation . . . . .	378
5 Validation and Evaluation . . . . .	380
6 Conclusions and Future Works . . . . .	381
References . . . . .	382
<b>Using Collaborative Robotics as a Way to Engage Students . . . . .</b>	385
Lina Narbutaitė, Robertas Damaševičius, Egidijus Kazanavičius and Sanjay Misra	
1 Introduction . . . . .	385
2 Pedagogical Backgrounds and Preconditions . . . . .	387
3 Methodology . . . . .	389
4 Case Study . . . . .	391
5 Evaluation . . . . .	393
6 Conclusion . . . . .	394
References . . . . .	395
<b>Assessing Scratch Programmers' Development of Computational Thinking with Transaction-Level Data . . . . .</b>	399
Milan J. Srinivas, Michelle M. Roy, Jyotsna N. Sagri and Viraj Kumar	
1 Introduction and Related Work . . . . .	399
2 Logging Transaction-Level Data Streams in Scratch . . . . .	400
3 Visualizing Learners' Computational Thinking Development . . . . .	402
4 Conclusions and Extensions . . . . .	406
References . . . . .	406

## About the Editors

**Dr. Shampa Chakraverty** is Professor in the Computer Engineering Division at Netaji Subhas Institute of Technology, New Delhi. She completed her B.E. in electronics and communication engineering at Delhi University, her M.Tech. in integrated electronics and circuits at IIT Delhi, and her Ph.D. at Delhi University.

Her research interests include sentiment, emotion and human language analysis, e-learning and engineering pedagogy, computer security—trust and digital watermarking, and design exploration of multiprocessor architectures.

**Dr. Anil Goel** is Vice President of engineering at SAP Canada, where he is also head of global development for a number of products and technologies related to SAP's HANA platform. He earned a Ph.D. (CS) from the University of Waterloo, Canada, and an M.Tech. (CS) from IIT Delhi.

**Dr. Sanjay Misra** is Professor of computer (software) engineering at Covenant University (CU), Nigeria. He received his M.E. in software engineering and his Ph.D. in information engineering from the University of Alcala, Spain.

His research interests are in information engineering, software engineering, Web engineering, software quality assurance, software process improvement, cloud computing and cybersecurity.

# **Part I**

## **Agile Software Development**

**Dr. Ritu Sibal Section Editor**

### **Editorial**

The field of software development has witnessed a gradual transition from prescriptive software development models to agile software development methods. Agile methods are flexible and amenable to changing business and customer requirements. This part is an amalgamation of some fine research works in the field of agile software development. Arumugam et al. propose a multi-agent framework for dynamically identifying the risks involved in various phases of global software development (GSD) and propose strategies for mitigating these risks. A major problem faced by teams involved in GSD is that of language. Distributed teams following agile software development have to additionally face the challenge of ethnic and cultural differences. Anita and Chauhan propose a technique based on linguistic parameters to understand user requirements expressed in different languages. Further, the authors propose noun and verb parameters to perform test case prioritization in agile software development.

Systematic and automated testing forms the bedrock of successful agile projects. Kapoor and Sagar develop a Web automation testing tool that empowers testers to create automated test scripts and execute test cases effectively and fastly. Sibal et al. propose a metaheuristic-based technique to prioritize user acceptance tests. Both these papers address the primary concerns of reducing the time to delivery and improving the return on investment (ROI) with agile software development.

Till date, software reliability measurement has been majorly dominated by traditional statistical reliability models. The use of machine learning and soft computing techniques provides new and more pragmatic approaches for measuring software reliability. Shukla et al. propose a deep learning technique to track bugs and estimate fault levels to improve the reliability of open-source software. There is an ardent need to dynamically predict changes, in the form of both defects and enhancements in software systems to tackle them in a timely manner. Agrawal and Singh evaluated and compared the performance of some popular statistical and machine learning methods for predicting change proneness in software systems. They validated the performance of these techniques using five open-source software

packages and demonstrated the superior performance of the machine learning approach in certain cases.

**Section Reviewers:**

Ankita Bansal  
Arun Sharma  
Bharti Suri  
Chamundeswari Amurugam  
Daya Gupta  
Deepak Sharma  
Har Deo Thakur  
Hardeep Singh  
Lokesh Jain  
Manu Sood  
Manoj Gaur  
Preeti Kaur  
Priti Bansal  
Rashina Hoda  
Ritu Sibal  
Rohit Beniwal  
Ruchi Sharma  
Sangeeta Srivastava  
Sangeeta Sabharwal,  
Sanjay Misra  
Sulabh Tyagi  
Vallidevi  
Vandana Bhattacharjee

# Risk Assessment Framework: ADRIM Process Model for Global Software Development



**Chamundeswari Arumugam, Sriraghav Kameswaran and Baskaran Kaliamourthy**

## 1 Introduction

Nowadays, IT companies witness a technological revolution. There appears to be an acceleration in the technological advancements, contributing to drastic changes in the way IT organizations develop and deliver products to their customers. As a matter of fact, GSD activities are distributed across the globe and hence it paves a way for the software practitioners to move from one region to another to take up the task up the GSD assignment. A software practitioner who takes up this GSD assignment is known as a global practitioner in this paper. Some of the challenges faced by the global practitioners are discrepancies in time zones, communication difficulties, adaptability to new environment, immigration trends and cultural differences.

Many software organizations tend to work closer on delivering the products to the customers rather than focusing on the risk involved. Only if risk is assessed, project completion time and resource usage can be speculated. Failure of risk assessment may lead to uncertainty in project completion. Therefore, risk associated with each phase of GSD has to be identified and mitigated.

Many IT companies are nowadays shifting their base from waterfall model of software development to agile model. In waterfall model, there will be a non-iterative sequential flow of requirement analysis, design and implementation, testing, maintenance. But agile model is an extension of incremental software development model

---

C. Arumugam (✉) · S. Kameswaran

Department of Computer Science and Engineering, SSN College of Engineering, Chennai, India  
e-mail: chamundeswaria@ssn.edu.in

S. Kameswaran

e-mail: sriraghav13104@cse.ssn.edu.in

B. Kaliamourthy

UST-Global (PepsiCo), Plano, TX, USA  
e-mail: baskaran@outlook.in

where the tasks are time boxed—divided to small time frames to deliver specific features for a release. Each agile iteration consists of a waterfall lifecycle. Agile methodology is used not only for software development, but also for risk identification and mitigation. The primary objective of this work is to propose an agile-based dynamic risk assessment framework that is capable of identifying the risk associated with GSD phases dynamically and mitigating it in the subsequent time box.

The Section-wise discussion of this paper is as follows. The existing research work is discussed in Sect. 2. Section 3 explains in detail the design framework of the proposed ADRIM process model. Results and discussion is explained in Sect. 4. Finally, Sect. 5 provides conclusion of this work.

## 2 Literature Survey

Kaplan and Garrick [1] define risk to be a superposition of uncertainty and loss when taking on a decision. Some researchers define risk as a negative occurrence that is likely to be caused by external or internal vulnerabilities. Chittister and Haimes [2] defined technical risk as follow: “*Software technical risk is a measure of the probability and severity of adverse effects that are inherent in the development of software and associated with its intended functions and performance requirements*”.

Upon studying the project delivery statistics of the software projects in IT companies, it is evident that a few companies fail to complete the project on time and with resources allocated. The reason accounting for this behavior is lack of proper risk management throughout the project lifecycle. Schwalbe [3] states project risk management as follows: “*Project risk management includes identifying, analyzing and responding risks related to the project*”. Also, identified risk as one of the knowledge area of project management that need to be facilitated to meet the project objective.

Many researchers [4–11] already contributed on GSD projects related to software project risk management strategies. Boehm’s [4] classification of risk management methods forms the basis for many risk management frameworks. Arumugam et al. [12] proposed an approach to quantify the global practitioners and organizational risk in GSD projects using multi agent simulation model. Several risk management approaches in existing literature have been discussed. Since IT companies have started to adopt agile based approaches for software development, there is a need to study how agile systems evolved and how risk can be assessed dynamically in an agile environment.

The “Agile Movement” was first witnessed in 2001 by a group of software professionals. They published the manifesto for agile software development [13]. Agile project management with scrum practices is more visible in terms of progress tracking and is capable of adapting to changing requirements as the project proceeds. A complete design framework of Agile based Dynamic Risk Identification and Mitigation (ADRIM) process is proposed in this paper. This process can be applied in GSD project stages to quantify the risk, and in turn explore the mitigation strategies using multi agent simulation model.

### 3 Research Methodology

#### 3.1 Mitigation Strategy for Various Risks

Conceptualization model is a vital activity in developing a system dynamics [14–17]. GSD projects undergo four phases in their development process. The various phases in GSD are forming, storming, norming and performing [18–20]. Each phase of the GSD project has a risk associated with it. Work culture, foster relationship, transparent process and visibility risks [12] are already existing and it is used in this paper for exploring associated risk related to project risk. The risk and its phases are represented in Fig. 1. Mere identification of risk is not enough. Identifying and mitigating the risk is the need of the hour [21]. Each risk state set has two decision variables—positive and negative variable, depending on whether the action favours the organization or not. States with negative decision variable must undergo mitigation. Each risk is explained and the mitigation strategy for each risk is listed as follows.

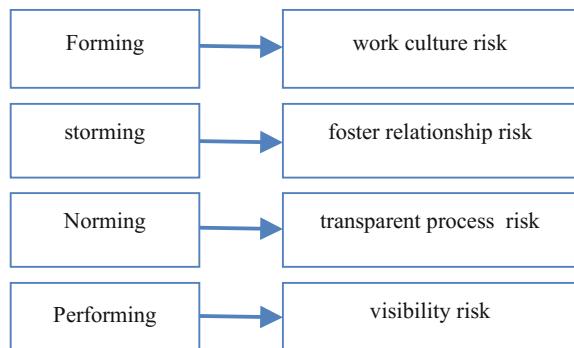
**Work Culture Risk (WCR)**—The skillsets of the practitioner are examined depending on which he is assigned with the task in GSD project. WCR depends on whether the task is taken up by the practitioner or not.

If a particular practitioner has denied the task assigned, the project delivery manager must find another person with similar skill sets to suit the project requirements. This task is of course definite subject to two facts:

1. The number of resource persons required for a particular project will be a definite number.
2. A person cannot deny more than a given number of projects in his tenure in the company.

**Foster Relationship Risk (FRR)**—After taking up the task assigned, the global practitioner should come up with a task plan for the work he is assigned with. The

**Fig. 1** GSD phase and its corresponding risk [12]



working status of the practitioner can hence be explored. FRR depends on whether the task plan is acknowledged by the committee or not.

Consider the task plan produced by a practitioner has some pitfalls and hence the evaluation committee denies his plan. Rather, experienced software professionals should mentor him on bringing up an effective and efficient task plan that is capable enough to deliver quality product at low project time and cost.

**Transparent Process Risk (TPR)**—Once the task plan is approved, the progress status must be communicated by the global practitioner to his peers and the project review members. TPR depends upon whether task is completed by the practitioner or not.

If the practitioner is unable to complete the task, it is the duty of the project lead to find out why. Some of the reasons include lack of project hours to complete, technical and setup issues in the project environment, lack of guidance in the concept, software bugs, and errors in the project code. Taking corresponding strategy would enable the practitioner to complete the task in an efficient way and deadline can be achieved.

**Visibility Risk (VR)**—Once the global practitioner finished building the deliverable, the software testing team has to validate it before it is delivered to the customer. This phase of GSD is where optimal solutions for developing the product can be explored. VR depends on whether the deliverable built by the practitioner is valid or not.

The practitioner who had been working on the product for a very long time would have got a deeper understanding. The product has to be subjected to various levels of software testing before it is released to the market and delivered to the customer. If some features built in the product do not comply with customer requirements, the practitioner must update the same in the product. Table 1 explains the various risk states identified in various phases of GSD along with the mitigation measures for each risk state.

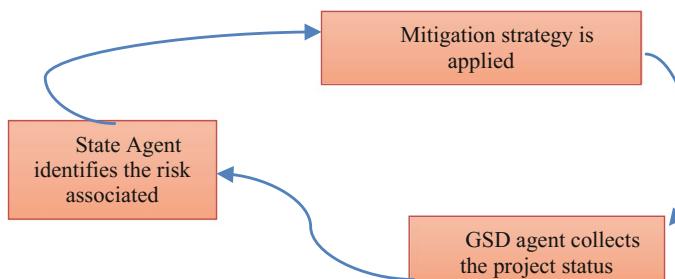
Table 1 briefly discusses the identified risk along with associated risk supporting each stage in software development. For each risk the decision variable may be positive and negative. Positive denotes no mitigation, while negative implies the urge of mitigation strategy. The mitigation strategy for each risk is summarized in the Table 1.

### 3.2 ADRIM Process Model

The dynamic risk and associated risk in a GSD project in all stages can be captured using multi agent. A multi agent collects all the information related to risk and its associated one for the success of the project completion. Collection is depicted using ADRIM process model in this work and represented in Fig. 2. The stage agent and GSD agent are two multi agents used in this study to collect the information related to the risk and possible applicability of mitigation strategy. Stage agent identifies the risk and associated risk in a four stage GSD project. In a particular stage of a project, if the decision variable is positive, it proceeds with next stage else hangs for

**Table 1** Mitigation strategy associated with each risk

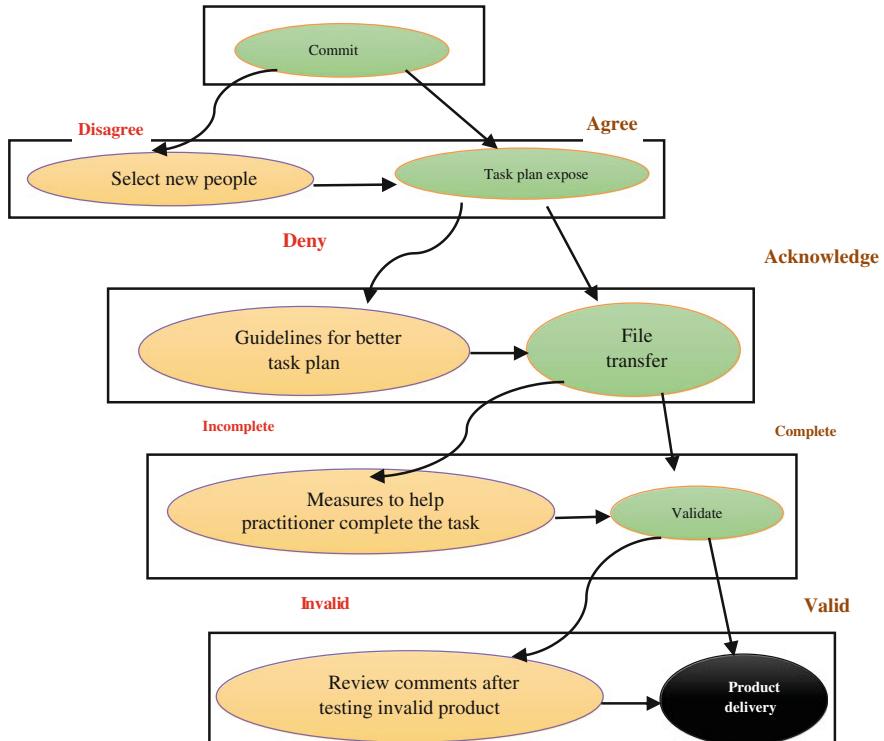
Risk name	Associated risks	Risk state	Decision variable	Mitigation strategy
WCR	Commit task Skill set issues Location preferences	Commit	Agree (Positive) Disagree (Negative)	Select new people with skill set and location preference
FRR	Complete task plan expose Pitfall task plan expose	Task plan expose	Acknowledge (Positive) Deny (Negative)	Mentor for effective and efficient task plan
TPR	File transfer Lack of project hrs Tech issues in prj env Lack of guidance Software bugs	File transfer	Complete (Positive) Incomplete (Negative)	Measure to help practitioner complete the task
VR	Product validate Subject to validation Undeliverable product Don't comply with req	Validate	Valid (Positive) Invalid (Negative)	Update the product based on review comments after testing

**Fig. 2** ADRIM process model

the mitigation strategy to complete. The GSD agent collects the project status for all the stages.

A binary tree is created to explain about the ADRIM process model. Each node in the binary tree denotes a state. A branch from a node signifies either a positive or a negative decision variable corresponding to the state. Nodes with positive decision variable proceed to the next state on its right branch. But nodes with negative decision variable must proceed to the “remedy” node on its left branch, to perform the remedial measure as recommended and then proceed to the next state.

The binary tree is shown in Fig. 3. There are totally four stages in GSD process and hence there are four state nodes. All the green colored nodes are state variables representing the risk associated with that stage of GSD. Each state has a corresponding remedial measure associated with it. Therefore there are four remedy nodes. The orange colored nodes are remedy nodes corresponding to risk in the previous level



**Fig. 3** Binary tree for ADRIM process model

in the binary tree. Product delivery forms the last step in any development lifecycle. So totally there are five levels in the binary tree. Each practitioner in the team is expected to arrive at the “product delivery” node at the end of project tenure.

Agile processes are always time boxed and each time box is called a sprint. Sprint duration varies from two to three weeks, depending on the agile method adopted by the company. By the end of each sprint, IT companies release a deliverable to their customers. This deliverable will constitute a set of features targeted for that sprint release. In the context of ADRIM procedure, each time box in general consists of remedial measure recommended for previous stage and risk evaluation of that stage. This is represented in Fig. 3 as a rectangular box in the binary tree.

## 4 Results and Discussion

Risk can be categorized based on risk level, as low to high depending on the criticality. Formula to access the risk is as follows.

**Table 2** Sample data of dynamic risk assessment on a Day 2 of ADRIM process

Risk assessment	Total no. of risks	No. of risk, mitigated	No. of risks, pending	Total risk, R
Forming	3	1	2	0.85
Storming	2	1	1	0.63
Norming	5	3	2	0.47
Performing	4	3	1	0.25
GSD	14	8	6	0.46

$$\text{Risk (R)} = \text{risk probability} \times \text{risk impact.} \quad (1)$$

Risk level of each risk is defined as—"very low", "low", "medium", "high" and "very high". Impact of risk level of each risk lie within the range of 1(very low) to 5(very high). Risk probability is assessed based on the information collected by a stage agent.

$$\text{Risk probability}_{\text{stage}} = \text{No. of risks pending in a stage} / \text{Total no. of risks identified in the stage} \quad (2)$$

Similarly, the risk probability based on a GSD agent can be assessed as follows.

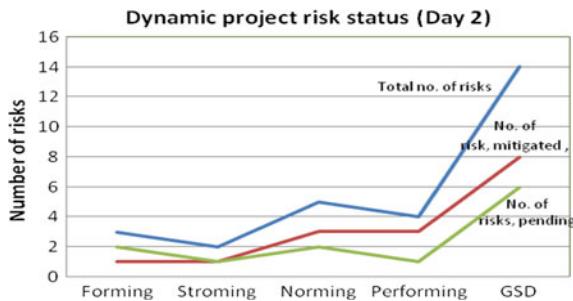
$$\text{Risk probability}_{\text{GSD}} = \text{No. of risk, pending in all stages} / \text{Total no. of GSD risks} \quad (3)$$

As ADRIM process maps the identified risk to "product delivery" node using a mitigation strategy, risks at each stage gradually reduces. A sample data table consisting of risks identified, mitigated and pending in each stage and overall risk is shown in Table 2. Taking "Forming" stage into account, the 3 risk levels have identified to be High (impact=4), Low (impact=2) and Very low (impact=1). Consider that the "Very low" risk has been mitigated. The pending risks are H and L. The risk associated with "forming" stage is calculated as:

$$\text{Risk}_{\text{forming stage}} = \frac{4 \times 1 + 2 \times 1}{4 \times 1 + 2 \times 1 + 1 \times 1} = 0.85 \quad (4)$$

Similarly, the risk factor R associated with each stage of GSD is computed. The overall risk is also calculated and represented as a graph in Fig. 4.

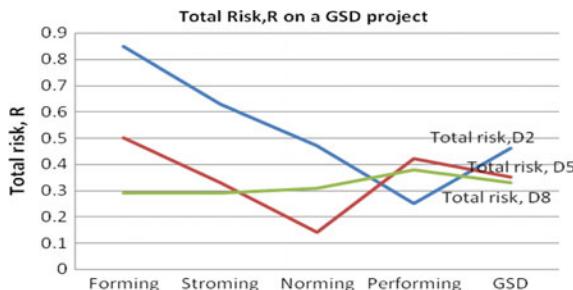
ADRIM is a dynamic process and hence it can be applied on a daily basis. The risk status on various days in the sprint was recorded as shown in Table 3. The total risk is calculated and plotted in Fig. 5. This chart would help in progress tracking and identifying the areas to focus and improve.



**Fig. 4** Risk status in ADRIM procedure on Day 2

**Table 3** Sample data of dynamic risk assessment on various days of the sprint

Risk category	Total risk Day 2	Total risk Day 5	Total risk Day 8
Forming stage	0.857	0.5	0.29
Storming stage	0.63	0.33	0.29
Norming	0.47	0.14	0.31
Performing	0.25	0.42	0.38
GSD	0.46	0.35	0.33



**Fig. 5** Risk status in ADRIM procedure on various days of the sprint

## 5 Conclusion

Measuring the GSD project risk serves as an essential deciding factor in project completion. A clear visibility into project status is attained by the proposed risk assessment procedure in this regard. Also, it provides the probability of risk by evaluating and ranking them in their order of importance or urgency. High risk is ranked as urgent risk, which needs to be focused at the earliest. However, many risk evaluation strategies currently employed in several IT companies terminate once the project risk is identified. But ADRIM is a dynamic process—once the risk is identified, remedial strategy are executed in next stage thereby mitigating the risk. Here in this work, multi agent enables to dynamically identify the risk and its associated risk

factors in a GSD project to apply the mitigation strategy in completing the project successfully. Further, agile-driven risk assessment approaches are very lesser and there is a huge scope to work on this field to come up with many novel approaches for risk assessment in agile environment.

## References

1. Kaplan, S., & Garrick, B. J. (1981). On the quantitative definition of risk. *Risk Analysis*, 1, 11–27.
2. Chittister, C., & Haimes, Y. Y. (1994). Assessment and management of software technical risk. *IEEE Transactions on Systems, Man and Cybernetics*, 24, 187–202.
3. Schwalbe, K. (2000). *Information technology project management*. Cambridge, MA: Course Technology.
4. Boehm, B. W. (1989). Software Risk Management Tutorial. IEEE CS Press.
5. Browning, T. R. (2014). A quantitative framework for managing project value, risk and opportunity. *IEEE Transactions on Engineering Management*, 61(4), 583–598.
6. Ebert, C., Murthy, B. K., & Jha, N. N. (2008). Managing risks in global software engineering: Principles and practices. In: *IEEE International Conference on Global Software Engineering*, pp. 131–140.
7. Lamersdorf, A., Münch, J., Torre, A. F. V., & Sánchez, C. R. (2011). A risk-driven model for work allocation in global software development projects. In: *Sixth IEEE International Conference on Global Software Engineering*, pp. 15–24.
8. Moe, N. B., & Smite, D. (2008). Understanding a lack of trust in global software teams: A multiple-case study. *Software Process Improvement and Practice*, 13, 217–231.
9. Nurdianni, I., Jabangwe, R., Smite, D., & Damian, D. (2011). Risk identification and risk mitigation instruments for global software development: Systematic review and survey results. In: *Sixth IEEE International Conference on Global Software Engineering Workshops*, pp. 36–41.
10. Usman, M., Azam, F., & Hashmi, N. (2014). Analysing and reducing risk factor in 3-C's model communication phase used in global software development. In: *International Conference on Information Science and Applications*, pp. 1–4.
11. Verner, J. M., Brereton, O. P., Kitchenham, B. A., Turner, M., & Niazi, M. (2014). Risks and risk mitigation in global software development: A tertiary study. *Information and Software Technology*, 56, 54–78.
12. Arumugam, C., Kameswaran, S., & Kaliamourthy, B. (2017). Global software development: A design framework to measure the risk of the global practitioners. In: *ACM ICCCT*, Allahabad, India
13. Beck, K., Beedle, M., Bennekum van, A., Cockburn, A., Cunningham, W., Fowler, M., et al. (2001). Manifesto for Agile Software Development 2002 <http://AgileManifesto.org>.
14. Alshammri, M. (2015). Simulation modeling of human aspects in software project environment. In: *ASWEC'15*, II, pp. 145–146.
15. Baxter, G., Sommerville, I. (2008). Socio-technical systems: From design methods to systems engineering. Submitted to The Journal of Human-Computer Studies.
16. Conchúir, E. O., Ågerfalk, P. J., Olsson, H. H., & Fitzgerald, B. (2009). Global software development: where are the benefits? *Communications of the ACM*, 52, 127–131.
17. Kellner, M. I., Madachy, R. J., & Raffo, D. M. (1999). Software process modeling and simulation: Why, What, How. *Journal of Systems and Software*, 46, 2/3.

18. Joslin, D., Poole, W. (2005). Agent-based simulation for software project planning. In: *Proceedings of the Winter Simulation Conference*.
19. Lock, R., & Sommerville, I., *Socio technical systems engineering handbook*.
20. Pressman, R. (2005). *Software engineering: A practitioner's approach*. McGraw-Hill.
21. Smite, D., & Borzovs, J. (2008). Managing uncertainty in globally distributed software development projects, University of Latvia, CSIT, 733, pp. 9–23.
22. Schwaber, K., & Beedle, M. (2002). *Agile software development with scrum*. Upper Saddle River, NJ: Prentice-Hall.

# An Extended Test Case Prioritization Technique Using Script and Linguistic Parameters in a Distributed Agile Environment



Anita and Naresh Chauhan

## 1 Introduction

Software testing is one of the aspects of software development that captures attention of every stakeholder. Reason for this attention is client. Primary goal of an organization is to satisfy client by delivering good quality product in less time. Secondary goal may be retaining client by maintaining relationship, retaining team members having specified expertise in specific domain, establishing standard processes, maintaining infrastructure, etc. Primary goal is highly dependent on secondary goals. Also, market popularity of any organization is determined on the basis of its product quality. Marketing executive may invite more clients by mentioning quality policies, standards and client satisfaction level for existing projects. In this way, organizations flourish with their standard policies.

Agile software development [1] has replaced traditional way of working and many organizations have transitioned their work style to agile. Agile is not specific to software rather its origin is from manufacturing industry and now, agile has been implemented in various sectors. Agile [2] is accepted among many organizations as agile is based on agile manifesto which were created by 17 agile practitioners. Their findings were not similar as they were from different programming methodologies. Finally, they reach to four values.

Agile is designed for handling the frequently changing requirements. Requirements are termed as user story in an agile context. Further, in this paper a method

---

Anita (✉)  
Evalueserve, Gurugram, Haryana, India  
e-mail: anitaarora\_20@rediffmail.com

N. Chauhan (✉)  
YMCAUS&T, Faridabad, Haryana, India  
e-mail: nareshchauhan19@yahoo.com

is proposed for prioritizing user stories and their test cases in a distributed environment. Section 2 of this paper discusses literature survey. Sections 4 and 5 reveals the proposed model for said problem. Section 6 concludes the proposed method by mentioning its future scope.

## 2 Literature Survey

Regression testing [3–8] is one of the software testing types that are needed for maintainability for the developed software. This is performed for not introducing any new bugs in the existing system when any new functionality is introduced by the client during sprint or iteration. In an agile context, regression testing is an ongoing activity. Pair programming is one of the way by which regression testing is implemented at unit level of sprint. Code is reviewed by the reviewer at the time of development of code. Regression testing is performed by various ways namely retest all, regression test selection [9] (RTS), test suite reduction (TSR) and test case prioritization [10, 11] (TCP). Out of all these, TCP is most popular among traditional practitioners and agile practitioners. In light of user stories, prioritization of user stories may be performed using client interest, user story complexity in terms of story point, user story effort/cost/time estimation value, market demand etc. After doing this prioritization, second prioritization may be performed that is valid for test cases of the particular user story. The methods in prior art are customer requirement based, coverage based, cost effective based, and chronographic history based.

Existing TCP method is based on planning game [12] of agile. In this technique, customer prioritizes acceptance tests in coordination with the test engineer and defines the value they bring to the user. Prioritized acceptance tests are then available to the next development phases. Problem with this technique is data on its effectiveness with existing test cases do not justify its adoption for regression testing.

Existing Agile TCP techniques are analytically reviewed in paper [13] titled “Analytical Review on Test Cases Prioritization Techniques”. In this paper, all types of TCP techniques such as customer requirement based technique, code coverage technique, cost technique, chronographic history technique are compared based on various criteria’s. Above specified paper also mentions about advantages and disadvantages of existing TCP methods for regression testing.

In agile context, customer requirement based technique may be combined with the proposed (novel) approach using noun and verb approach so as to generate more prominent results. Selection of this approach out of existing approaches is done on the basis of agile principles which say that client satisfaction is most important. First two agile principles are mentioned here for more clarity.

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer’s competitive advantage.

### 3 Modified User Story

Verbal communication [14] especially, email writing faces different challenges while understanding requirements from C. Adobe Captivate is the tool that is used for understanding the requirements in a better way, stated by C. This understanding is done by using punctuations, with the requirements. The process is used by CR and Team members. Using adobe captivate tool, text may be converted to audio. By using punctuations, pauses may be inserted in the audio. Comma is more frequently used punctuation. Capital letters are used for reading abbreviations. In this way, email writing text can be converted to audio. This audio, when played, will appear as if, C himself is telling her requirements. C may not be expert in captivate tool. If she is, in that case, she can send the audio of her requirements in French language; otherwise, CR will convert requirements into audio files using English Language. Also, CR may collaborate with team or with different stakeholders [15] in framing user story and ready story using captivate tool [16].

In it, slides notes or script is to be written for epic, user story or ready story. CR may work in collaboration with team to complete this. Once scripts with graphics (.cptx files) are converted to audio files, these converted audio files are published and uploaded on the central database at the workplace. These uploaded files (having extensions .swf, .avi) may also be accessed by team members who are working at distributed locations.

For the given project requirement, user stories are shown in Table 1. Project requirement is related with a user and database. There may be many types of users depending upon their job profiles. Also, their requirements may accordingly vary.

Table 1, includes the user stories, as per the requirement of every type of user. Also, these user stories are written by incorporating as much punctuation as possible so as to have clear understanding of the requirements.

### 4 Sprint—Story Prioritization

The proposed work includes two levels of prioritization, namely, story prioritization and test case prioritization. In this section, story prioritization has been described in detail. Story prioritization may be performed by a method in which punctuation marks are of great importance. For the purpose of attaining quality, punctuated user story has to be reviewed two times. One review may be performed by scrum master, if SCRUM methodology is followed, or team member and other review by CR. Freezing of user story is the important step, as further work is to be implemented on this final user story. Feedback cycle is an ongoing activity in agile environment. Negative feedback has to be overcome by introducing required changes. Next step is to prioritize user stories by using excel formulas. Len and Substitute formulae have been used to count the number of individual punctuation in the user story. For example, following combined formula has been used.

**Table 1** “User stories”

S. no.	User story
1	As a searcher, I want to search patents of US, EP and India, on the basis of, bibliographic details, such as, application number, publication number, priority date, inventor, assignee, date of patent, so as, to perform searches comprising invalidity search, claim mapping, claim charting
2	As a lawyer, I want to search legal status of patents, by mentioning publication number, application number, so as to handle infringement suit of different assignee of US, EP or India
3	As an Inventor, I want to search, patents of software domain, comprising, agile software development, software engineering, software quality, software testing etc. so as, to perform “state of the art” search, for countries: US, EP and, India
4	As a researcher, I want to search, patents of US, as, US is the software hub of software patents, so as, to improve upon, my findings
5	As a drafter of a patent, I want to search, related patents, using software keywords, so as, to learn drafting skills, of software domain
6	As a business analyst, I want to search, patents of Countries: US, EP or India, so as, to analyze market trend of patents, in software domain, for different assignees
7	As a statistical professional, I want to, search revenue spend on patents, by assignee of countries: US, EP and India
8	As an examiner, I want to search, Data Base, for finding prior arts of a given patent by mentioning, its bibliographic details, including application number and, priority date
9	As an analyzer, I want to, categorize patent of different classes, such as, United States classification (USC), Cooperative patent classification (CPC), International patent classification (IPC)
10	As a petitioner, I want to, download patent document, depending upon its legal status: active, pending, inactive, abandoned and revoked, so as, to read patent sections, such as drawings, detailed description, summary, objects of the invention, background, abstract, claims

$$= \text{LEN}(\text{B2}) - \text{LEN}(\text{SUBSTITUTE}(\text{B2}, ", ", ""))$$

where, B2 represents any cell in the excel sheet, and 2nd argument of the Substitute function is the specific punctuation that is to be search in the B2 cell. Also, sum function is used to calculate total number of occurrences of punctuations in a cell. Respective formula for the same is:

$$= \text{SUM}(\text{C2}: \text{K2})$$

where, C2 to K2 range is selected for performing addition of specific values. After applying these functions, sort was performed so as to get the most risky user story.

The priority order for the user stories, in terms of risk, of a given project. Order is as follows:

$$2 < 7 < 5 < 8 < 4 < 6 < 9 < 3 < 1 < 10$$

User story 10 is the most risky story and user story 2 is the least risky story. In this way, story point marking for a user story becomes easy and also, effort estimation for completing any user story can be calculated. In addition, more risky story would have more number of confirming points.

## 5 Linguistic Parameters-Noun and Verb

The process of story building starts as soon as prioritization step is over. Till now, we have covered understanding C requirements, user story framing and user story prioritization. Next step is to create ready stories for the respective user stories. On the similar basis, for user story 1–10, ready story is written, in collaboration with the CR. Story number 10 is the most risky story, so, this story is considered as seed story, for the purpose of calculating linguistic parameters.

After getting answers of these confirming points, client acceptance strategy is set. This strategy helps in preparing definition of done (DOD) for all stakeholders.

### 5.1 Agile Change Management (ACM)—User Story

C or CR may introduce complex requirement at end time. The specified requirement implementation may have substantial effect on the existing system working. Effective execution of regression testing may stop unwanted introductions of bugs. Now, question is how to implement regression testing effectively. There are many ways of implementing regression testing but, this paper describes test case prioritization: a method for running test cases. Whenever, there is any change in the existing story or new user story is introduced in the list of backlog items, planning changes from various angles. Various angles could be

- Run the existing sprint to its full pace.
- Stop existing sprint and focus on new requirement.
- Do Technical management.

Here, this last step is tricky and unique one. By looking at the new user story, CR may take a decision to instruct team members regarding step 1 and step 2. Further, she may communicate the respective decision to scrum master of the team or product owner as they are usually involve with the prioritizing of the ready stories or user stories. In this case, CR take this decision depending upon his technical understanding and user story requirement As she is the person who is involved before, during or after the sprint.

After following this decision, onus lies on team to do it fast so as to entertain more backlog work and completion of the current sprint. This may be achieved by following test case prioritization technique of the regression testing. In paper Titled “*A Linguistic Approach for Test Case Prioritization in an Agile Environment*”, authored

by Anita Arora and Naresh Chauhan, published in 13th Annual International STC 2013 (Software testing Conference), held at Bangalore on 4th–5th Dec 2013, test case prioritization was performed on the basis of sentence priority score. In this paper some of the steps of the prioritization have been implemented using Microsoft Excel 2007. So, test case prioritization process has been started by finding noun and verb of ready story of new requirement in the existing implemented user stories' noun and verb tables. Finding of noun and verb is performed by making use of inbuilt formulae of Excel.

For reference, most risky user story 10 has been considered as the new user story for performing test case prioritization and rest of the user stories 1–9 are the user stories which are delivered to the C after proper quality check. User story 10 has the following elaborated requirements based on the confirming points (ready story) which are output of communication among team members and CR. One important point is that framing ready story is mandatory step for getting good hold on the client's perspective for the effective requirement development. Here, CR inputs are of great importance. Following are the confirming points for user story 10.

- Petitioner has access to active patents in .pdf format.
- Petitioner has access to revoked patents' case history in .pdf format.
- Petitioner has option of saving patent subsections as per the space available on her system.
- Petitioner has access to pending published applications in .pdf format.
- In the database various types of legal status are searched based on the unique publication number, application number or patent number or petition number.

In implementing test case prioritization, foremost step is to find noun and verb in these confirming points. Confirming points and ready story are used interchangeably.

If the nouns or verbs are to be found in the existing user stories 1–9 then following excel formula can be used:

$$\text{IF}(\text{ISNUMBER}(\text{SEARCH}("Patent", \text{Cell number})), "YES", "NO") \quad (\text{F1})$$

In this case, ready story's nouns such as Patent, Subsection etc. of new user story 10 are found in user stories 1–9. Using same formula, nouns may be searched in the existing ready story's 1–9 if time permits.

Similarly, verb search may be performed for user story or ready story.

$$\text{COUNTIF}(\text{C2 : I11}, "YES") \quad (\text{F2})$$

Further, to count, total number of "YES" for nouns, Count if function may be used (refer (F2) shown above). This function returns 3, 4, 1, 1, 1, 1, 1, 2 and 1, as number of nouns in respect of user stories 1–9. Similarly, total number of verbs is calculated. The count if function returns 1, 1, 1, 1, 1, 1, 1, 1, 0, as number of verbs in respect of user stories 1–9. Using multiplication operator on these count values of noun and verb, effect (refer Fig. 1) is calculated for new ready story in respect of user stories 1–9.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Story No.	No. of Noun	No. of Verb	Effect										
2	2	4	1	4										
3	1	3	1	3										
4	8	2	1	2										
5	3	1	1	1										
6	4	1	1	1										
7	5	1	1	1										
8	6	1	1	1										
9	7	1	1	1										
10	9	1	0	0										
11														
12														
13														
14														
15														
16														
17														
18														

**Fig. 1** Effect

By reviewing, effect values; it is clear that user story's 9 has effect value zero. So, user stories 1–8 as per sorted effect values are:

$$2 > 1 > 8 > 3, 4, 5, 6, 7$$

User story 2 with effect value 4 is the user story that is most adversely affected by the change introduced by C in terms of user story 10. Similarly, user stories 3, 4, 5, 6, 7 with effect value of 1 are of equal importance and are least affected by the change introduced by C in terms of user story 10. Considering user story 2, as the seed story for performing test case prioritization (TCP). Before performing TCP on the confirming points, ready story with corresponding test cases is shown below in Table 2.

For explaining extended TCP, CRP value 1 has been considered as the relevant case for discussion (Table 3).

CRP Client Representative Priority (1: Highest and 5: Lowest Priority)

TC Test Case

CP Confirming Point

RSP Ready Story Point (based on Fibonacci Series)

NP Number of Punctuations

NV Number of Verbs

SPS Sentence Priority Score

Sentence priority score SPS has been calculated on the basis of two new parameters RSP and NP along with CRP and NV. Extended formula for the SPS is shown below in (F3). Also, summation has been avoided as SPS is calculated for every test case separately.

**Table 2** Ready story 2

2	User story	As a lawyer, I want to search legal status of patents, by mentioning publication number, application number, so as to handle infringement suit of different assignee of US, EP or India
CRP	Confirming points	
3	CP-1	<i>Lawyer can enter her credentials on the user interface, by doing prior registration</i>
	TC-1.1	Credentials may include lawyer patent agent number, country code (where he/she) is practicing and password
	TC-1.2	Password for the lawyer may include maximum 8 alphanumeric characters
	TC-1.3	Password has to be changed on monthly basis
	TC-1.4	Country Code of the lawyer can be combination of, three alphabetic characters
	TC-1.5	Depending upon the country code, patent agent text field should appear
	TC-1.6	Registration details need approval from administrator
1	CP-2	<i>Lawyer can search legal status of patents, in the database, on the basis of publication number and application number</i>
	TC-2.1	Lawyer can search legal status of patents in the database on the basis of different color codes
	TC-2.2	Records in the database have to be indexed by hashing concept
	TC-2.3	Active patents link should display patents with complete specification
	TC-2.4	Pending patents link should display title, abstract and claims of the patent application
	TC-2.5	Abandoned/inactive patent application should display reasons for abandonment/inactiveness along with Title of the patent application
	TC-2.6	Revoked patent application should display grounds on which patents was revoked
2	CP-3	<i>Lawyer can perform the search for different Assignee of different domains</i>
	TC-3.1	Lawyer should be able to perform the search on the basis of Full name of the Assignee
	TC-3.2	Lawyer should be able to perform the search on the basis of abbreviated form of the Assignee
	TC-3.3	Lawyer should be able to perform the search on the basis of Assignee of telecommunication domain
	TC-3.4	Lawyer should be able to perform the search on the basis of Assignee of software domain

(continued)

**Table 2** (continued)

	TC-3.5	Lawyer should be able to perform the search on the basis of number of years of establishment of the Assignee
	TC-3.6	Lawyer should be able to perform the search on the basis of number of patents being revoked by the opposition party
1	CP-4	<i>Lawyer can perform the search on the basis of number of infringement cases against Assignee</i>
	TC-4.1	Lawyer should be able to perform the search on the basis of type of patent infringement done by the Assignee
	TC-4.2	Lawyer should be able to perform the search on the basis of amount of compensation paid for infringement
	TC-4.3	Lawyer should be able to perform the search based on the number of pending infringement cases against the Assignee
	TC-4.4	Lawyer should be able to perform the search based on the number of active infringement cases against the Assignee
	TC-4.5	Lawyer should be able to perform the search based on the number of resolved infringement cases against the Assignee
4	TC-4.6	Lawyer should be able to perform the search based on the duration of infringement cases against the Assignee
	CP-5	<i>Lawyer can Analyze results after performing search</i>
	TC-5.1	Lawyer should be able to analyze the search results based on pie charts
	TC-5.2	Lawyer should be able to analyze the search results based on graphs
	TC-5.3	Lawyer should be able to analyze the search results based on year range
	TC-5.4	Lawyer should be able to analyze the search results based on Assignees
	TC-5.5	Lawyer should be able to analyze the search results based on Country code
5	TC-5.6	Lawyer should be able to compare the search results based on domain area of the Assignee
	CP-6	<i>Lawyer can sort the results obtained by applying search criteria on the basis of Country codes</i>
	TC-6.1	Sorting of results are to be performed from ascending to descending order
	TC-6.2	Sorting of results are to be performed from descending to ascending order

**Table 3** Sentence priority score

CRP	RSP	CP/TC no.	NP	NV	SPS
1	8	CP-2	–	–	–
		TC-2.1	3	1	24
		TC-2.2	2	1	16
		TC-2.3	2	1	16
		TC-2.4	2	1	16
		TC-2.5	4	1	32
	5	TC-2.6	2	1	16
		CP-4	–	–	–
		TC-4.1	3	1	15
		TC-4.2	3	1	15
		TC-4.3	3	1	15
		TC-4.4	3	1	15
		TC-4.5	3	1	15
		TC-4.6	3	1	15

**Table 4** “SPS/TCP sorting”

CP/TC no.	SPS
TC-2.5	32
TC-2.1	24
TC-2.2	16
TC-2.3	16
TC-2.4	16
TC-2.6	16
TC-4.1	15
TC-4.2	15
TC-4.3	15
TC-4.4	15
TC-4.5	15
TC-4.6	15

$$\text{SPS} = \text{RSP} * \text{CRP} * \text{NV} * \text{NP} \quad (\text{F3})$$

Need for introducing this extended TCP is:

- To have more quality by counting number of punctuations as more relationship would be identified.
- To resolve the issue by introducing RSP in the formula when NV is same for all test cases. Sorted SPS is shown in Table 4.

Highest value in SPS column represents highest priority of the corresponding test case. This sentence priority score which is based upon NP and RSP is less time consuming as compared with random/no ordering of test cases. The core measure for

effectiveness of build is velocity. If velocity measure is showing that the tested part is following the definition of done then acceptance criteria is met. Another metric is (average percentage of fault detection) APFD (as shown in formula (F4)). APFD can be calculated as follows:

$$\text{APFD} = 1 - \{(Tf1 + Tf2 + \dots + Tfm)/mn\} + (1/2n) \quad (\text{F4})$$

where, n be the no. of test cases and

m is the no. of faults.

(Tf1, ..., Tfm) are the position of first test T that exposes the fault.

Table 5 is a table showing occurrence of faults by running test cases. For TCP, APFD comes out to be 83.3%. Using no ordering method (refer Table 6), APFD comes out to be 75.8%. Thus the prioritized test cases yield better fault detection than the non-prioritized test cases.

$$\begin{aligned} \text{APFD} &= 1 - \{(1 + 3 + 2 + 1 + 2 + 1 + 5 + 2 + 7 + 1)/10 * 12 + 1/2 * 12 \\ &= 1 - 25/120 + 1/24 \\ &= 0.833 \end{aligned}$$

$$\begin{aligned} \text{APFD} &= 1 - \{(5 + 2 + 1 + 3 + 1 + 5 + 4 + 1 + 7 + 5)/10 * 12 + 1/2 * 12 \\ &= 1 - 34/120 + 1/24 \\ &= 0.758 \end{aligned}$$

## 6 Conclusion

This paper suggests a method for test case prioritization which is based on linguistic parameters such as noun, verb and punctuations in distributed agile environment. Here, two level of prioritization is disclosed namely story prioritization and test case prioritization. Stories and ready story are formulated in collaboration with CR. In addition, story prioritization has been performed using punctuation parameter. Further, a technique has been proposed and proved by calculating sentence priority score of different confirming points based on linguistic parameters. Finally, APFD metric has been used to validate the need of test case prioritization as compared to no ordering approach for test cases. Results show that TCP yields 83% efficiency as compared with 75.8% efficiency of no ordering approach. Microsoft Excel has been used as the tool for performing all the required calculations and representations.

The proposed method for TCP is not fully automated. In future, work may be done to implement every step of the algorithm for saving time and faster execution.

The proposed regression test case prioritization method which is based on linguistic parameters such as noun and verb may be implemented by considering other parameters such as adjectives, adverbs, accent and many more. This will improve the

**Table 5** “TCP ordering APFD”

Faults	Test cases										
	TC-2.5	TC-2.1	TC-2.2	TC-2.3	TC-2.4	TC-2.6	TC-4.1	TC-4.2	TC-4.3	TC-4.4	TC-4.5
F1	*									*	
F2		*							*		
F3	*								*		*
F4	*			*				*			*
F5	*			*				*			
F6	*			*				*			*
F7				*			*				
F8		*		*			*				
F9						*					
F10	*								*		

**Table 6** “No ordering APFD”

Faults	Test cases										
	TC-2.1	TC-2.2	TC-2.3	TC-2.4	TC-2.5	TC-2.6	TC-4.1	TC-4.2	TC-4.3	TC-4.4	TC-4.5
F1	*				*					*	
F2	*							*			
F3	*							*			*
F4		*			*					*	
F5	*			*							
F6			*		*			*			*
F7				*				*			
F8	*		*								
F9						*					
F10						*					

accuracy in the existing algorithm. Further, macros may be implemented for same functionality using VB script. Other than this, effectiveness may be measured using other metrics of the software testing.

**Acknowledgements** We would like to thank the reviewers for their useful suggestions that helped us to improve our work. We would also like to extend our gratitude to our employers who provided us with the needed research facilities.

## References

1. Cao, L., & Balasubramanium, R. (2007). Agile software development: Ad-hoc practices or sound principles?. In: *IEEE ITPRO*, pp. 41–46 March–April 2007.
2. Agile Alliance. (2001). Principles Behind the Agile Manifesto. [www.Agilemanifesto.org/principles.html](http://www.Agilemanifesto.org/principles.html).
3. Pettichord, B. (2004). Agile testing challenges. In: *Pacific Northwest Software Quality Conference* (2004).
4. VersionOne (2014). 8th Annual State of Agile Survey. <http://stateofagile.versionone.com/>.
5. Dhalait, S. A. D., & TCS Limited. Agile unit and regression testing framework for domain specific languages. Publication number US20130159963.
6. Meszaros, G. (2003). Agile regression testing using record and play. In: *OOPSLA*, Anaheim, California, 26–30 Oct 2003. ACM 1-58113-751-6/03/0010.
7. Anita, A., & Naresh, C. (2010). Testing in an agile environment: A project. In: *International Conference on Next Generation Communication and Computing Systems (ICNGC2S-10)*, NITTTR, Chandigarh, India, 25–26 Dec 2010.
8. Crispin, L., & Gregory, J., *Agile testing: A practical guide for testers and agile teams* (1st ed.). ISBN-13: 978-0321534460.
9. Arora, A., & Chauhan, N. (2014). A regression test selection technique by optimizing user stories in an agile environment. In: *2014 IEEE International Conference on Advance Computing Conference (IACC)*, held in ITM University on 22nd–23rd Feb 2014, Gurgaon. Proceedings published on IEEE explorer.
10. Anita, A., & Naresh, C. (2013). A linguistic approach for test case prioritization in an agile environment. In: *13th Annual International Software Testing Conference 2013*, held at Bangalore.
11. Pradeepa, R., & Vimala Devi, K. (2013). Effectiveness of test case prioritization using APFD metric: Survey. In: *International Conference on Research Trends in Computer Technologies (ICRTCT—2013)*. Proceedings published in *International Journal of Computer Applications® (IJCA)* 0975–8887.
12. Ponaraseri, S., Susi, A., & Tonella, P. (2018). Using the planning game for test case prioritization. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.623.8446&rep=rep1&type=pdf>.
13. Sultan, Z., Bhatti, S. N., Abbas, R., Shah, S. A. A. (2017). Analytical review on test cases prioritization techniques: An empirical study. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 8(2), 293–302.
14. <http://humanresources.about.com/od/interpersonalcommunicatio1/qt/nonverbal-communication-in-the-workplace.htm>.
15. Arora, A., & Chauhan, N. (2013). A simplest agile life cycle for all stakeholders. In: *CSI Sponsored International Conference on Software Engineering (CONSEG-2013)*, held in Pune from 15–17 November, India.
16. <http://www.adobe.com/support/captivate/gettingstarted.html>.

# AutoJet: Web Application Automation Tool



Sheetika Kapoor and Kalpana Sagar

## 1 Introduction

In the IT industry where software development is important, software testing is also crucial for the success of a project. It is too costly to fix a bug after release than testing and fixing before release [1]. Long-term project continuously demands new functionality due to which the whole system can be impacted. Furthermore, IT industry now emphasizes on adopting agile framework [2]. The agile framework includes small cycles of release with a working product at the end of each release. After each build released, testers must validate that addition of new functionality does not impact existing system called regression testing. Automation plays an important role in regression testing as it reduces the cost and increases efficiency to test same scenarios again and again before each release. Twenty to fifty percent of software development cost involves for system testing, and test automation is proposed to lower this cost [3].

**Automation Testing** is a process of automating the execution of test cases using various automation tools by comparing its actual results with expected results [4]. Expected results are stored in tools in various formats, e.g., excel sheets, text files, CSV files, and TSV files. **Automation Testing Framework** provides a structure for implementing test automation. It consists of an application under test, test scenarios, test cases, test steps, test results, and expected results. It provides methods of comparing test results with expected results and then the storing status of test case execution in pass or fail [5]. **Automation Testing Tools** can be open source or paid. The selection of tools depends upon nature of the project to be automated. Further,

---

S. Kapoor (✉) · K. Sagar  
Department of Computer Science and Engineering, USICT,  
GGSIPU, New Delhi, Delhi, India  
e-mail: sheetikakapoor@gmail.com

K. Sagar  
e-mail: sagarkalpana87@gmail.com

to automate test cases, it is necessary to decide which test harness<sup>1</sup> is suitable for a project and which tool should be used. So, it entirely depends upon the criticality of project, budget, and time span. For example, there are different tools for Web applications, mobile applications, cloud services, GUI testing, and performance testing. The success of test automation tool depends upon its simplicity, usability, supported features as well as on time and tester's skills required to fulfill the purpose of having efficient automation scripts.

Selection of right tool is a crucial step in test automation process as it can result in high cost. Further, selection of inappropriate tools may result in project failure. There are certain **principles that must be adopted for selection of right tools:** (1) It should not be more complex; (2) it should be maintainable; (3) it should be easy to use by any tester; (4) it should be powerful and efficient enough to automate more test cases in less time [6]

Web applications are critical in nature as its UI interacts directly with customers. A single defect leakage can cause high potential loss which depends upon the severity of that defect. Defects in the Web site would not only cause wealth loss but also degrades stature of the company [7]. Web applications need to be validated across various platforms. Existing tools provide a fragile solution which results in test breakage due to even minute functionality changes. Moreover, UI of Web applications tends to change frequently by modifying existing functionalities or introducing the new ones. It emphasizes the need of automated test scripts to validate existing functionalities of Web applications [8]. The efforts needed for its regression testing are much more than that needed for its system testing [9]. In Web sites, there could be various possible ways to test functionality that results in increased number of test cases that to be validated each time during regression testing to state the stability of new build which in turn increase test execution time and efforts [10].

In 2016–17, a survey has been conducted by Capgemini, HPE, and Sogeti which is called World Quality Report. In this report, top 7 challenges are addressed in the domain of application development that highly affects the automation testing. As per the report, 45% challenges are due to lack of effective build/integration automation and 41% are due to reliance on manual testing. In this report, CIO of Retail, Sweden, has mentioned that “Automation plays a major role in the transition to DevOps. Use of different automated tools is key to the success of DevOps programs”. His statement is also supported by VP of application, Italy as mentioned that “The biggest challenge in testing because of the transition to DevOps is Automation, as it is the core of the successful DevOps Cycle” [11].

Further, we have conducted a survey<sup>2</sup> to discover challenges and perspective of testers concerning Web automation testing. In this survey, participants have addressed the challenges they faced during creation and execution of automation scripts. In this survey, the following issues have been addressed: (1) gap in the knowledge

---

<sup>1</sup>Test Harness is a test automation framework responsible for accepting test data, execute the test and validate results by comparing expected and the actual outcome.

<sup>2</sup><https://www.surveymonkey.com/r/2PJ7NNW>.

of programming language between an automation and manual tester, (2) lack of integrated tool.

In this paper, we are proposing Web automation tool AutoJet as a solution of various challenges addressed in the surveys, e.g., integration of various tools needed for test harness, lack of experienced resources, instability, lack of right automation tool, challenges with test data. Furthermore, the paper is structured as follows: Sect. 2 describes related work. Section 3 describes adopted methodology, whereas Sect. 4 describes conclusion and future work.

## 2 Related Work

Various test automation harness and tools have been developed to provide ease to automate tests. Most widely adopted automation tools are Selenium, QTP, SOAP UI. Moreover, automation frameworks, e.g., Miroslav Bures and Martin Filipsky, have been proposed to extend Selenium WebDriver to SmartDriver to reduce maintenance and implementation cost. They have proposed a three-layered architecture that includes reusable components, Page Objects, and Tests. Further, the framework constitutes of two modules: structuring and maintenance module that are applicable for Web UI and Mobile Applications. It has Tests module that allows creating tests which can be easily read by testers and business analysts [12].

Stocco et al. [13] have proposed a tool named as APOGEN which automates the creation of Page Objects of the Web application. This tool has implemented some functionalities for these automatically created Page Objects.

Nguyen et al. [14] have proposed a tool named as GUITAR. It can be added as a plug-in for better flexibility and extensibility. This tool provides a framework for model-based testing and could run on six different GUI platforms.

Chaini et al. [15] have proposed a Web automation framework in which test scripts are created and stored in function library which runs against test data and includes test and error logs. The framework emphasizes on how to analyze failed scripts. Disha Garg et al. [16] have proposed a keyword-driven framework<sup>3</sup> for Web automation testing. This framework is based on Marven tool. It has specified functionalities against each keyword and executes scripts based upon keywords using Marven tool.

Leotta et al. [17] have done a case study on the efforts needed to maintain different locators used in Selenium framework and concludes that ID locators used in conjunction with LinkText are the best solution to modify the test automation pack to the new release of the application.

Gojare et al. [18] have proposed Web automation framework using Selenium and TestNG<sup>4</sup> which includes features of sending the report and takes screenshots. Rishab

<sup>3</sup>Keyword Driven Framework performs test execution by applying actions on web elements based on keywords like click, input.

<sup>4</sup>TestNG is an open source testing framework used in integration with selenium for automated test execution.

Jain et al. [19] have also proposed similar framework using TestNG which could integrate with any Selenium WebDriver-based framework. Manual testers could execute test scripts but need knowledge of framework, tool and programming language. Kumar and Saxena [20] have proposed data-driven framework which is based on Selenium WebDriver and TestNG. This framework would support execution of the same script with different test data which is stored in external files, e.g., excel, CSV, database, and XML. Testers just need to modify test data, and it would not impact framework code.

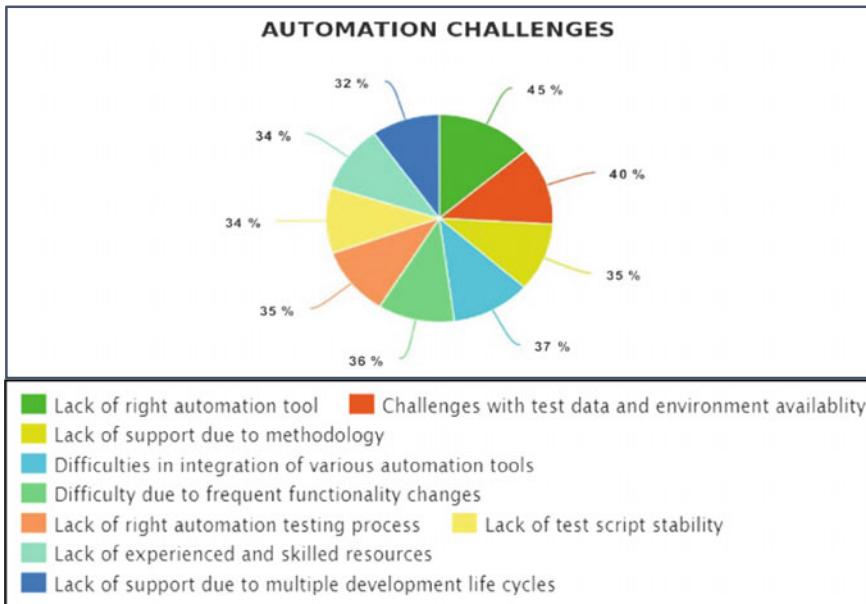
However, many frameworks and tools have been developed to support test automation by providing features to reduce the efforts of testers. These frameworks provide features to automate test scenarios and generate efficient test scripts. Still, automation testing coverage reduces significantly in 2016. As per World Quality Report 2016–2017, it is due to the need of multi-skilled people and lack of solution of challenges in the world of automation testing [11]. For automation testing, testers need to acquire knowledge of programming languages. Further, the understanding of automation framework is required and results in need of skilled testers or developers for automation testing. In the World Quality Report 2016–17, various statistical data have been mentioned that reveals the current state of automation testing. In this report, Deepika Mamnani, Principal, Financial Services SBU, Capgemini, stated skill shortage problems around testers. They lack in various skills like writing TDD or BDD (20%), development and coding skills (19%), functional test automation expertise (18%). IT Director of Pharmaceuticals, Ireland, further mentioned that “The number one challenge is finding skilled people”. A quantitative study has been performed on automation challenges. Results are described in Fig. 1 [11].

This study reveals that 45% challenges are due to lack of right automation tool. Moreover, 35% are due to lack of experienced and skill testers. As proposed, AutoJet would provide support to testers in such a way that they are able to create automated scripts with the help of its user interface component without any need of programming language.

### 3 Methodology

#### 3.1 Context of Proposal

Nowadays, various automation testing tools have been evolved to provide ease to testers but still failed to increase coverage of automation testing in software projects. In this study, a survey has been conducted to welcome participants to come up with present challenges they face while working on automation testing during software development projects. Challenges accentuated in latest World Quality Report 2016–2017 are also addressed and considered in proposed Web automation tool. An innovative tool named AutoJet is proposed in this paper that aims to provide the solution of addressed challenges to endorse automation testing in software development projects.



**Fig. 1** Automation challenges mentioned in World Quality Report 2016–17 [11]

### 3.2 Participants

A survey has been conducted with 75 participants across six organizations and four universities. Participants from different backgrounds include manual tester, automation tester, test lead, test manager, test analyst, developer, and professor. The age group of participants ranges between 24 and 65 years with an average of 42 years. The list of participants includes 32 females and 43 males. Details of participants are mentioned in Table 1. The challenges that they have faced during Web automation testing are illustrated in Table 2.

In this survey, 42.67% participants have addressed challenges related to programming language gap. Furthermore, 6.67% have addressed installation issues. 9.33% have addressed issues with integration of various tools to have complete test automation framework while 16% participants have embraced on stability issues and 25.33% on lack of usability of test automation tool or framework.

**Table 1** Participants details

Participants	Age group			Gender	
	24–37	38–51	52–65	Male	Female
Count	26	31	18	32	43
Percentage (%)	34.67	41.33	24	42.67	57.33

**Table 2** Challenges reported

Qualification of participants	Challenge area				
	Programming language	Installation	Integration	Stability	Usability
Manual tester	12	2	1	1	8
Automation tester	3	1	0	0	2
Test lead	9	0	3	1	1
Test manager	1	0	2	2	0
Test analyst	4	1	1	1	3
Developer	0	0	0	4	0
Professor	1	0	0	3	3
Student	2	1	0	0	2

### 3.3 The AutoJet

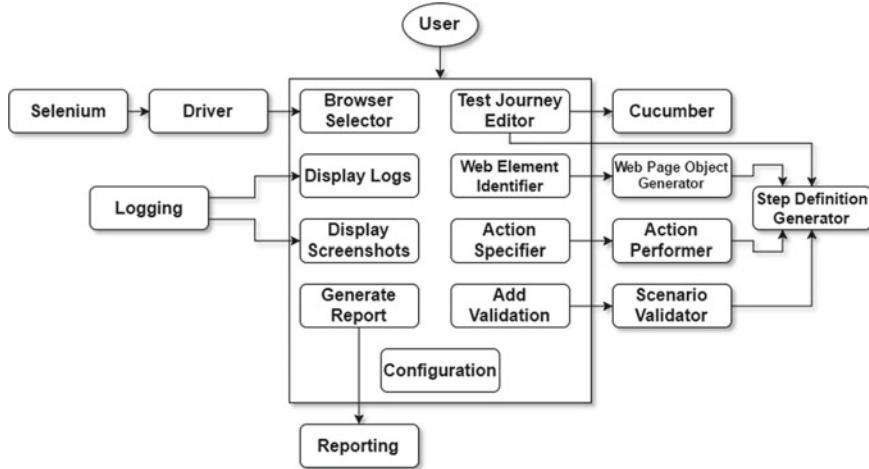
AutoJet is a JAVA-based Web site automation tool, proposed to automate test cases as fast as Jet. This innovative tool is capable of automating test scenarios without any need of extensive programming knowledge. By using this tool, even a manual tester can create automated test scripts efficiently and fastly. Furthermore, it aims to provide the following solutions for addressed challenges in test automation:

- Reduction in the skills gap among testers
- Reduction in time to automate test scenarios
- Minimum efforts for integrating various tools
- Capable of automating test scenarios without extensive programming knowledge
- To have more generalized test cases for common understanding among testers, developers, manager, and clients.

AutoJet is an extension of Selenium WebDriver in integration with TestNG as Unit Test Framework and Cucumber for writing test cases in BDD<sup>5</sup> format. AutoJet allows testers to specify Web element via its identification using HTML tag and selects Web actions to be performed on that Web element. It asks the tester to specify validation criteria which are used to decide whether test scenario is passed or not. To validate one complete test flow, test journeys are created by AutoJet automatically. Test execution results are reported to stakeholders<sup>6</sup> via two ways: (1) view report and (2) email test report to the intended audience. AutoJet comprises of six components as follows: Test Journeys, User Interface, Web Page Objects, Step Definitions, Reporting, and Logging. The architecture of AutoJet is defined in Fig. 2.

<sup>5</sup>BDD is Behavior Driven Development used to write test cases in a more generalized form that enables tester, developer, manager and clients to have a common understanding.

<sup>6</sup>Stakeholders are those people who are interested in the success of the project like Tester, Test Lead, Test Manager, Project Manager, Client.



**Fig. 2** Architecture of AutoJet

### 3.3.1 Test Journeys

Test Journeys are test case scenarios to verify end-to-end flow of Web application. Test journeys include business-critical scenarios which enhance client's confidence during User Acceptance Testing. Generally, one Test Journey is created against one end-to-end flow. These journeys are not written in a similar fashion as in case of writing system testing test cases. These test cases can cause ambiguity among stakeholders as created from the perspective of a tester whereas test journeys need to be written in the BDD format to have a common understanding among all stakeholders. Each file under Test Journey would be called Feature File. Its template would be in the following format:

```

Feature:<Description of Test Journey>
Scenario Outline:<Title of Test Scenario>
Given<Pre-Requisite of Test Journey>
And<To add more Pre-Requisite conditions>
When<Actions to be performed>
And<Add more actions to be performed>
Then<Expected Result>
And<Add more to expected results>
Examples:
|<Test Data>|
  
```

'Given', 'When' and 'Then' are mandatory statements and to be specified only once. 'And' statements are optional and could be specified any number of times depending upon the scenario to be tested. 'Test Data' would be mapped to the parameterized test

steps. The iterations of a specific Test Journey execution depend upon the number of rows of Test Data. It will execute for each set of Test Data specified in Feature File.

For Example:

**Feature:** Validation of login into the Web application

**Scenario Outline:** User login with valid credentials

**Given** User is at Web application login page

**When** User enters <Username> and <Password>

**Then** Home page should be displayed

**Examples:**

| Username|Password|

|TestUser|TestPassword|

Here username and password are parameters which will receive values from Test Data specified under ‘Examples’ section.

There could be more than one scenario per feature file, but they should be in the same context.

### 3.3.2 User Interface

User interface of AutoJet provides various screens that allow testers to (1) add Test Journeys, (2) create Web Page Objects by specifying HTML tag and its value, (3) specify actions, and (4) validations to be performed on page object.

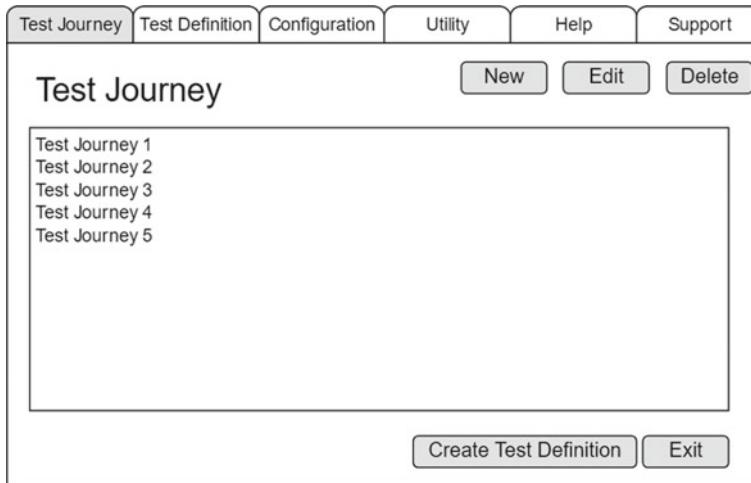
The first tab would be Test Journey. It will display all existing journeys which could be modified or deleted. New test journeys could be added via ‘New’ button. It allows testers to create test cases in BDD format called Test Journeys. BDD format is most widely used format in the field of automation testing as it is written in general English language which provides a common understanding of test scenarios among all stakeholders. ‘Create Test Definition’ would navigate the user to Test Definition tab corresponding to newly created test journey.

The second tab would be Test Definition. This tab enforces tester to write Test Journeys in BDD format. It provides a drop-down list to select the type of Web element and HTML tag as identifier. Textbox in the second screen would allow a tester to specify the value of the identifier. Furthermore, tester selects the action to be performed from drop-down list which is customized as per the type of Web element. Moreover, a tester needs to specify validation criteria to decide upon its success or failure. Test Definitions created could be modified and delete accordingly.

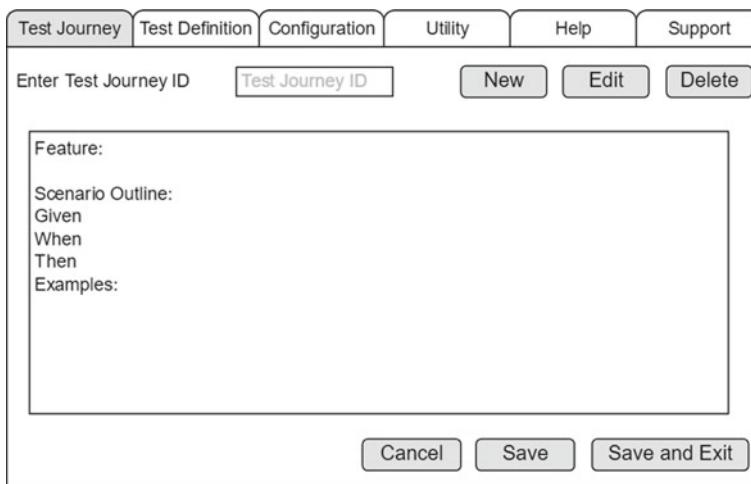
The third tab is Configuration screen that allows a tester to set various application-related configurations such as URL of application, browser, and email addresses of concerned stakeholder to whom Test Report is to be sent after each run of Automation Pack.

The fourth tab is Utility screen. This screen would provide various features like view report, logs, screenshots, and email report. Reports will present test execution data via various graphical techniques. These graphs help to provide a concise view of test execution to clients and other stakeholders.

Rest two tabs Help and Support are included to provide support to its users. The email address would be provided to report any issue or inconvenience while using this tool. Feedback option will also be there to enhance AutoJet. Some templates of User Interface are mentioned in Figs. 3, 4, 5, 6 and 7.



**Fig. 3** Test journey screen



**Fig. 4** New test journey screen

Test Journey   Test Definition   Configuration   Utility   Help   Support

Map Test Journey   Test Journey ID ▾

Step 1: Create Page Object

Select Element   Type ▾   Identifier ▾   Value

Choose Action   Actions ▾

+   -

Add Validation

Add Next Step   Save and Exit

**Fig. 5** Test definition screen

Test Journey   Test Definition   Configuration   Utility   Help   Support

## Configurations

Web Application URL   URL

Select Browser   Select Browser ▾

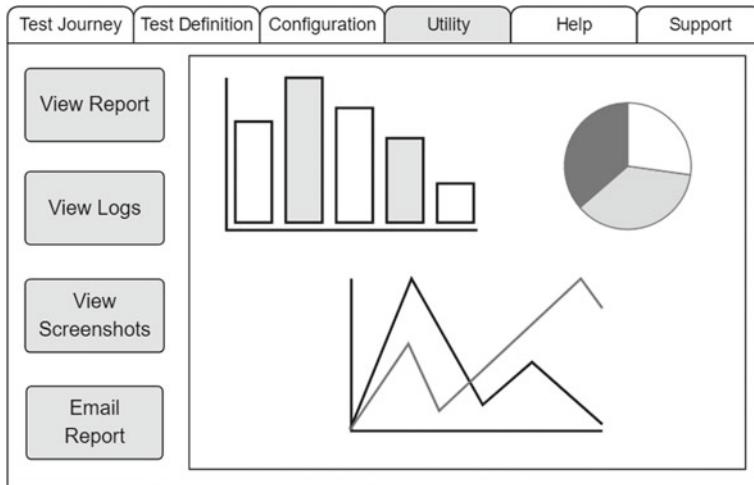
Select Report Recipients   Email Address

Save and Exit   Cancel

**Fig. 6** Configuration screen

### 3.3.3 Web Page Objects

Web Page Objects are identifiers that aim to identify each Web page element uniquely. These Web Elements are the components constitute Web application like textbox, checkbox, label, button, radio button, drop-down. Tester specifies type of Web element, HTML tag, and its value on Test Definition screen of AutoJet which automatically generates Web Page Objects via its framework. This functionality will use Selenium tool which performs specified actions based upon details provided by the



**Fig. 7** Utility screen

user on Test Definition screen. The pseudocode of automatic creation of Web Page Objects is as follows:

**Algorithm:**

**Input:** BrowserName, Identifier, Value

**Steps:**

1. Driver=getDriver(BrowserName)
2. Web Element=createPageObject(Identifier, Value)

This algorithm will initiate driver of browser selected by the user in configuration along with the creation of Web Page Objects which takes Identifier and its Value as parameters.

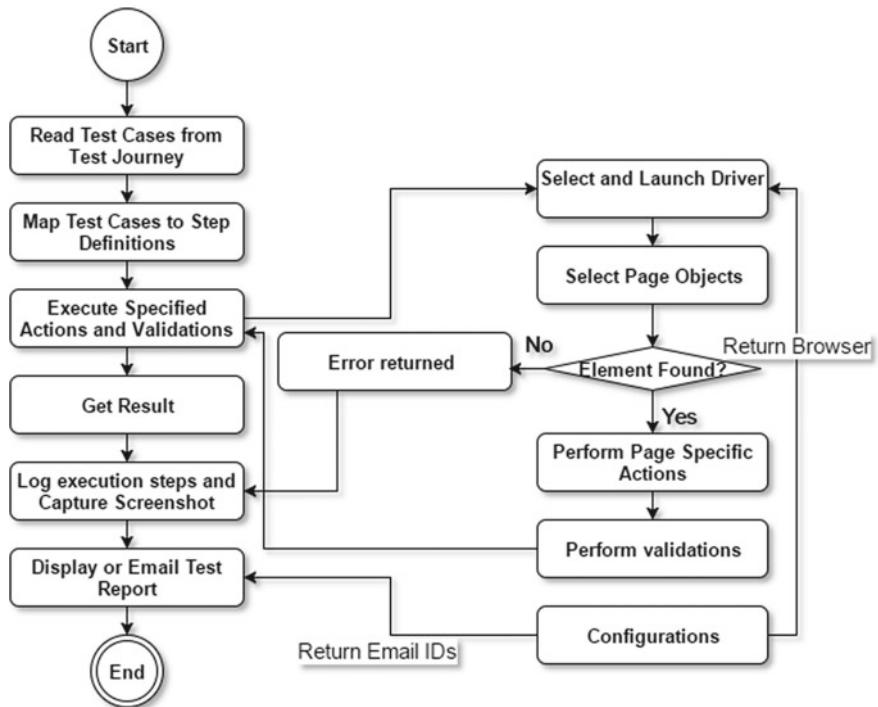
**Sample Web Page Objects are:**

```
WebElement Username=driver.findElement(By.id("uname"));
WebElement Password=driver.findElement(By.id("pswrd"));
```

### 3.3.4 Step Definitions

Step Definitions contain one JAVA file per Test Journey that contains functions for each action to be performed on Web page objects as specified by tester on User Interface of AutoJet. This Step Definitions files call various common functions internally included in its framework to maintain its modular architecture. It will act as a heart of this tool which integrates other modules of AutoJet. This framework will validate response at each step and log any deviation in the form of log file and screenshots. Its framework is specified in Fig. 8.

The pseudocode of flow of Autojet's framework is as follows:



**Fig. 8** Framework of AutoJet

**Algorithm:**

Input: Web Element, Actions, Validation, URL, BrowserName, Recipients

Steps:

- 1.Driver=getDriver(BrowserName)
- 2.launchApplication(Driver, URL)
- 3.if (SelectedJourneyCount>=1) then
  - for each SelectedJourney
    - if (ActionCount>=1) then
      - for each Action
        - execute Action,Web Element
      - else return “No action exists”
    - if (ValidationCount>=1) then
      - for each Validation
        - assert Validation
      - else return “No validation exists”
    - if (assert Validation) then
      - Result.Add(TestCaseID, “Pass”)
      - else Result.Add(TestCaseID, “Fail”)
- 4.emailReport(Recipients, Result)

This algorithm will select Test Journey to be executed and launch the application using browser's driver. Web page objects are selected on the basis of data specified in Journey and perform specified actions. Each step will be validated against expected

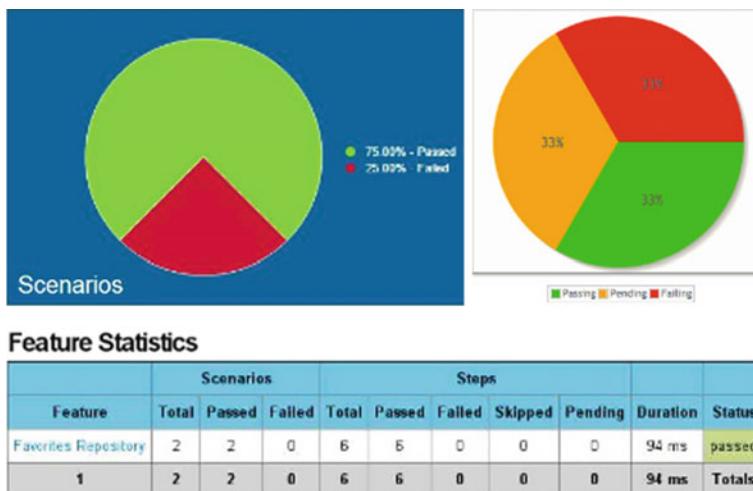


Fig. 9 Sample of Test report [21]

result to mark its success. Any variation will be recorded and email Test Report to the configured audience.

### 3.3.5 Reporting

Reporting in AutoJet will be supported using Jenkins Plugin. Reporting includes two sections: **Display Test Report** and **Email Test Report**. Display Test Report component would display test report on screen. Email Report component would email test report to an intended audience. These reports would contain pie charts and details of each passed and failed scenarios. This test report would help Test Manager to share status report with clients or other stakeholders. The report would be in the format specified in Fig. 9.

The pseudocode of email Test Report is as follows:

**Algorithm:**

**Input:** To, From, Host, Result, DateTime, Recipients

**Steps:**

- 1.Properties=getSystemProperties()
- 2.message=MimeMessage()
- 3.message.setSubject("Test Report", DateTime)
- 4.message.setTo(To)
- 5.message.setFrom(From)
- 6.message.setHost(Host)
- 7.message.setRecipients(Recipients)
- 8.Transport.send(message)

This algorithm will email report to intended audience as configured.

### 3.3.6 Logging

Logging would be performed in two ways: One is to maintain log file to capture steps performed and errors occurred while another would be capturing screenshot of the Web application where test scenario failed.

This logging helps testers to identify the reason for the failure of test scenario as well as its point of occurrence. This would further help testers to decide whether it is just test data issue or defect in the Web application.

Logging feature would be available at User Interface component of AutoJet which allows testers to view log files and screenshots captured.

## 3.4 *Adaption of Autojet in Agile Methodology*

Agile methodology promotes concurrent iterations of both software development and software testing within SDLC of a project. Smaller sprints of 2–4 weeks are generated to deliver new functionality to customers. Agility allows modification in requirements even at later stages of development. In this methodology, requirements are delivered frequently with maximum customer satisfaction in terms of quality. Regression Testing need to be performed before closure of each sprint which ensures that no new defect introduced into the system due to new functionality added in the current sprint. As the system grows in terms of functionality, the count of regression test cases increases significantly. Due to a smaller duration of sprints, i.e., 2–4 weeks, it becomes difficult for a tester to execute regression test cases manually. Therefore, agile methodology endorses the creation of automation scripts for regression test cases. The creation of automation scripts required an integrated framework along with appropriate development skills. With existing automation tools, this process becomes time-consuming and resource's skills constrained. Due to the short time span, the automation coverage of regression test cases is generally less than the required percentage.

Time and quality are two most prominent factors in agile methodology. AutoJet can ensure the quality of the software within the time span of a sprint from the creation of automation scripts to its execution. It is easily adaptable in Regression Testing phase of agile methodology as it eliminates the time and cost involved in generating the automation framework and hiring the resources as per required development skills. AutoJet enables a tester to create automation scripts of regression test cases as fast as Jet without any constraint on development skills. It allows testers to follow the testing processes from test case creation to test reporting along with logging feature to triage the root cause of failed test cases. These automation scripts executed without any human intervention which can ensure the quality of build outside working hours.

## 4 Conclusion and Future Scope

The development and specification of test automation tool is a complex process whose success depends upon its features, adopted approach, framework, and usability. Further, various challenges are being faced in the expansion of automation testing despite the availability of many Web automation tools in the market.

In order to identify these challenges faced in the domain of Web automation testing in the present era, a survey has been conducted that aims to address various critical issues, e.g., lack of integrated tools, a dearth of skilled and experienced automation testers, and scarcity of time to create automation scripts.

Furthermore, nowadays, agile methodology is highly adopted in most of the software development projects. In Agile-based projects, all testing activities, e.g., creation of test plan, test case creation, manual testing of new functionalities, regression testing of existing functionalities using automated test scripts, and test summary creation, need to be completed in a sprint of 2–3 weeks. This enforces the need of automation scripts to be developed efficiently in less time along with testing of new functionalities. AutoJet would allow testers to create automated test scripts for regression test cases within the sprint of agile methodology.

In this study, an attempt has been made to propose an innovative tool called AutoJet. The proposed tool is a milestone for automation testing domain because it provides an integrated solution for critical challenges that are addressed through the survey. The solutions fully support adaption and expansion of Web automation testing. It empowers testers to automate test cases effectively and efficiently without any extensive programming knowledge. It further provides support to manual testers in terms of User Interface component. It enriches testers to perform various testing activities using single tool AutoJet, e.g., creation of test cases in the form of Feature Files, Automatic Test Execution, View and Email Test Report. Furthermore, AutoJet tool allows testers to analyze failed Test Journeys using log files and captures screenshots.

In this paper, AutoJet tool is being proposed and the future work can be done for its implementation with different advanced features, i.e, to support parallel execution of Test Journeys which further reduces its execution time effectively. Secondly, the integration of AutoJet with Appium can be done to support automation testing of Mobile applications. In near future, research can be pursued in these directions.

## References

1. Garousi, Vahid, & Mäntylä, Mika V. (2016). When and what to automate in software testing? A multi-vocal literature review. *Information and Software Technology*, 76, 92–117.
2. Olsson, H. H., Alahyari, H., & Bosch, J. (2012). Climbing the “Stairway to Heaven”—A multiple-case study exploring barriers in the transition from agile development towards continuous deployment of software. In *2012 38th EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA)*, on (pp. 392–399). IEEE.

3. Alégroth, Emil, Feldt, Robert, & Kolström, Pirjo. (2016). Maintenance of automated test suites in industry: An empirical study on Visual GUI Testing. *Information and Software Technology*, 73, 66–80.
4. Garg, D., Singhal, A., & Bansal, A. (2015). A framework for testing web applications using action word based testing. In *2015 1st International Conference on Next Generation Computing Technologies (NGCT)*, (pp. 593–598). IEEE.
5. Wang, F., & Du, W. (2012). A test automation framework based on WEB. In *2012 IEEE/ACIS 11th International Conference on Computer and Information Science (ICIS)*, (pp. 683–687). IEEE.
6. Sharma, Monika, & Angmo, Rigzin. (2014). Web-based automation testing and tools. *International Journal of Computer Science and Information Technologies*, 5(1), 908–912.
7. Molina, A. I. et al. (2012). CIAT-GUI: A MDE-compliant environment for developing graphical user interfaces of information systems. *Advances in Engineering Software* 52, 10–29.
8. Brajnik, G., Baruzzo, A., & Fabbro, S. (2015). Model-based continuous integration testing of responsiveness of web applications. In *2015 IEEE 8th International Conference on Software Testing, Verification and Validation (ICST)* (pp. 1–2). IEEE.
9. Wandan, Z., Ningkang, J., & Xubo, Z. (2009). Design and Implementation of a Web Application Automation Testing Framework. In *Ninth International Conference on Hybrid Intelligent Systems, 2009. HIS'09* (Vol. 2, pp. 316–318). IEEE.
10. Banerjee, I., Nguyen, B., Garousi, V., & Memon, A. (2013). Graphical user interface (GUI) testing: Systematic mapping and repository. *Information and Software Technology*, 55(10), 1679–1694.
11. Capgemini, HPE, & Sogetti (2017). World Quality Report 2016–2017. <https://www.capgemini.com/thought-leadership/world-quality-report-2016-17>.
12. Bures, M., & Filipsky, M. (2016). SmartDriver: Extension of selenium WebDriver to create more efficient automated tests. In *2016 6th International Conference on IT Convergence and Security (ICITCS)* (pp. 1–4). IEEE.
13. Stocco, A., Leotta, M., Ricca, F., & Tonella, P. (2015). Why creating web page objects manually if it can be done automatically? In *Proceedings of the 10th International Workshop on Automation of Software Test* (pp. 70–74). IEEE Press.
14. Nguyen, B. N., Robbins, B., Banerjee, I., & Memon, A. (2014). GUITAR: an innovative tool for automated testing of GUI-driven software. *Automated Software Engineering*, 21(1), 65–105.
15. Chaini, H. S., & Pradhan, S. K. (2015, March). Test script execution and effective result analysis in hybrid test automation framework. In *2015 International Conference on Advances in Computer Engineering and Applications (ICACEA)*, (pp. 214–217). IEEE.
16. Garg, D., Singhal, A., & Bansal, A. (2015). A framework for testing web applications using action word based testing. In *2015 1st International Conference on Next Generation Computing Technologies (NGCT)* (pp. 593–598). IEEE.
17. Leotta, M., Clerissi, D., Ricca, F., & Spadaro, C. (2013). Comparing the maintainability of selenium webdriver test suites employing different locators: A case study. In *Proceedings of the 2013 International Workshop on Joining Academia and Industry Contributions to Testing Automation* (pp. 53–58). ACM.
18. Gojare, Satish, Joshi, Rahul, & Gaigaware, Dhanashree. (2015). Analysis and design of selenium webdriver automation testing framework. *Procedia Computer Science*, 50, 341–346.
19. Jain, C. R., & Kaluri, R. (2015). Design of automation scripts execution application for selenium webdriver and test NG framework. *ARPJN Journal of Engineering and Applied Science*, 10, 2440–2445.
20. Kumar, A., & Saxena, S. (2015). Data driven testing framework using selenium WebDriver. *International Journal of Computer Applications*, 118(18).
21. <https://sqa.stackexchange.com>.

# Prioritization of User Story Acceptance Tests in Agile Software Development Using Meta-Heuristic Techniques and Comparative Analysis



Ritu Sibal, Preeti Kaur and Chayanika Sharma

## 1 Introduction

Agile software development has gained vast popularity in the past decade as it promotes lean documentation, is flexible to changes, results in quick development and is customer-centric. The implementation of a user story begins only after successful acceptance testing of the user story. Customer-specified acceptance criteria are used for deriving user story acceptance tests in agile software development. Acceptance test-driven development ensures that the user stories are accepted by the customers before implementation. The number of user stories and their associated acceptance tests increases as the size of an application increases. Execution of all acceptance tests for user stories is time-consuming and delays delivery of a working software.

In this paper, we propose a prioritization technique to prioritise user story acceptance tests and identify the critical acceptance tests. The implementation of critical acceptance tests is sufficient to ensure the acceptance of a user story by the end user/customer. This reduces testing time and effort in agile software development and facilitates faster delivery of working software. Initially, acceptance tests for a user story are written in the Given-When-Then (GWT) template [1]. An activity diagram is then drawn for the acceptance test in the GWT formula. The GWT template explicitly lists out the observable consequences on carrying out an action in some context. UML activity diagram shows the flow of activities of an object in a

---

R. Sibal · P. Kaur (✉)  
Netaji Subhas Institute of Technology, New Delhi, India  
e-mail: preetikaur1@rediffmail.com

R. Sibal  
e-mail: ritusib@hotmail.com

C. Sharma  
ABES Institute of Technology, Ghaziabad, Uttar Pradesh, India  
e-mail: chaya29@gmail.com

system. The actions and their corresponding consequences for a user story acceptance test are represented by the activities in an activity diagram. The activity diagram is converted into a control flow graph (CFG) to show the flow of control between the various activities of the activity diagram. Each node of the CFG represents an activity, and the edges connecting nodes of CFG depict control flow of the activities. We identify the critical path/acceptance test by applying meta-heuristic technique. Three meta-heuristic techniques, namely genetic algorithm, cuckoo search algorithm, and micro-genetic algorithm have been applied independently to find the highest priority acceptance test. This is the critical test that covers the maximum acceptance criteria for a user story. This prioritization facilitates the developer in selecting the acceptance test that must be tested first and the successive tests with decreasing priority. Thereby, this technique helps effectively plan and schedule the testing process to enable quicker delivery of working software.

In this paper, a comparison of the performance of the meta-heuristic techniques used for prioritization is also drawn. The comparison shows that in our case cuckoo search technique outperforms the other two meta-heuristic techniques.

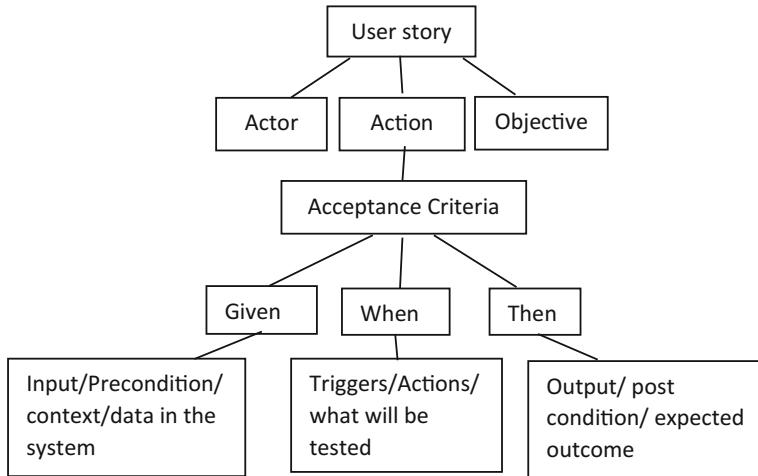
This paper is organized as follows. Section 2 gives a brief introduction of the concepts that are used in the proposed approach. Section 3 describes the proposed approach with the help of a small but realistic case study followed by a comparative analysis among the various meta-heuristic techniques for the case study. Section 4 concludes the paper along with overview of future work.

## 2 Basic Concepts

This section describes basic concepts used in the proposed approach.

### 2.1 Given-When-Then Format

Given-When-Then (GWT) format shown in Fig. 1 is a template used for writing acceptance tests for a user story in agile software development. The acceptance criteria associated with user story are tested with acceptance tests written in GWT format. The Given part in this format is the input or preconditions which triggers some action and then provides the output. It suggests preconditions and the post-conditions needed for performing the acceptance testing for a user story. The GWT communicates the purpose of the test and each scenario is checked once.



**Fig. 1** Given-When-Then format

## 2.2 Overview of Meta-Heuristic Algorithms

In this section, we briefly discuss the three meta-heuristic algorithms, namely Genetic Algorithm, Cuckoo Search Algorithm, and Micro-GA.

### 2.2.1 Genetic Algorithm

Genetic algorithm is a prominent and widely used optimization technique and search method [2]. It is a search method based on the concept of natural evolution of species in nature. GA uses the technique of natural selection of the fittest individuals of the population occurring in nature. A set of chromosomes which is a string of binary digits represent all the possible solutions to the problem being solved. Each digit of a chromosome is called a gene. The initial population can be selected randomly or can be manually created. The pseudocode of GA given by Goldberg. D.E. is shown below [2, 3]:

```

Initialize (population)
Evaluate (population)
While (stopping condition not satisfied)
{
  Selection (population)
  Crossover (population)
  Mutate (population)
  Evaluate (population)
}
  
```

The operators used in GA are selection, crossover, and mutation. New population is created by using three operators as described below:

- Selection: Selection operator selects the individuals for creating new individuals. This is done on the basis of the fitness of the individuals called the fitness function. The fitness of an individual shows the capability of an individual for survival and reproduction in an environment. This selection operation creates the new population of individuals from the old individuals. Every chromosome of the current generation is evaluated by finding its fitness value. The fitness value is used for selecting individual candidates from the existing population to produce next generation.
- Crossover or recombination: The crossover operation is applied after selection to the current generation. To bring diversity in the population, genes or sequence of bits are swapped in the string through crossover or recombination between two individuals.
- Mutation: New traits are introduced in the chromosomes through gene mutation, i.e., by changing the chromosomes in small ways. This operation is applied to bring diversity in the population.

### **2.2.2 Micro-GA**

According to Carlos et al. [4] “In micro-GA, population is of small size with reinitialization”. Krishankumar [5] implemented micro-GA on a problem having a small population of size 5 with crossover rate of 1 and mutation rate of 0.0. The next generation has only the best candidates from the previous generations. Selection of individuals is done through tournament selection strategy. The working of micro-GA is illustrated below [5]:

1. A population of size 5–7 individuals is generated randomly.
2. Individual having best fitness value is carried over to the next generation.
3. To determine parents of remaining four or six individuals, tournament selection strategy is applied.
4. If population is converged, the best individual is kept and another four or six individuals are randomly generated. If population is not converged, the process starts again from step 2.

### **2.2.3 Cuckoo Search Algorithm**

Meta-heuristic cuckoo search algorithm was developed by Yang and Deb [6] in 2009. Search space in cuckoo search algorithm is a generation that represents a set of host nests and each nest is carrying an egg or the solution [6]. Each generation has fixed number of solutions. As we know that in nature, the cuckoo bird lays her egg in the nest of some other host species and the host bird recognizes that the egg is of some other species with a probability  $pa \in [0, 1]$ . The host bird either throws the egg or

destroys the nest after finding the fact that the egg belongs to some other species. The new egg is made by modifying one solution randomly. The new and better solution replaces the existing solution. Best individuals are carried over to the next generation, and worst solutions of fraction  $p_a$  are discarded to be taken over by new and better solutions. The working of the cuckoo search algorithm is illustrated below [7]:

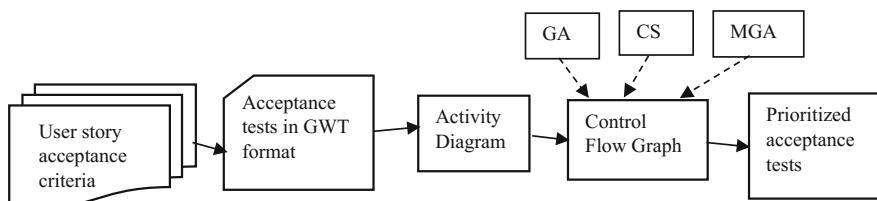
1. Create an initial population representing  $n$  host nests.
2. While stopping condition is not reached, generate an initial population randomly and evaluate its fitness function.
3. Randomly choose one nest among  $n$  host nests. Replace existing solution by the new solution, if new solution is better.
4. Destroy fraction  $P_a$  of worst nests and construct new nests.
5. Carry over the best solutions to next generation.
6. Find the best solution by ranking all the solutions.

### 3 Proposed Approach

In this section, we explain our proposed meta-heuristic-based approach for prioritization of user story acceptance tests. Initially, user story acceptance tests are written in GWT format. A UML activity diagram is drawn corresponding to an acceptance test expressed in GWT format. This activity diagram is then converted into a CFG where each node denotes an activity and the edges in the CFG depict the flow of control among the activities (Fig. 2).

Acceptance testing requires generating set of acceptance tests that will cover every path of the CFG. Finding the entire set of acceptance tests may be expensive and time-consuming due to various reasons. For example, infinite paths may exist when there are loops in CFG. We propose to solve this problem by finding the critical path using meta-heuristic algorithm and concept of information flow (IF). For effective testing, we assume that each loop is executed at most once. We apply three meta-heuristic techniques, i.e., genetic algorithm (GA), cuckoo search algorithm, and micro-GA separately to find the critical path.

The steps for identifying acceptance tests that must be tested first are outlined below:



**Fig. 2** An overview of the proposed approach

1. The given user story acceptance test is structured into GWT format.
2. UML activity diagram is drawn for user story acceptance test in GWT format.
3. The UML activity diagram for user story acceptance test is converted into an intermediate CFG. Apply basic IF model [5] to the assign weights to the nodes of CFG. According to the basic IF model, information flow metric (IF) is applied to the components/modules of a system design. In our proposed work, the node of the CFG is considered as the component of the system. Information flow is calculated for each node of CFG using Eq. (1) of basic IF model.

$$\text{IF}(A) = \text{FAN-IN}(A) * \text{FAN-OUT}(A) \quad (1)$$

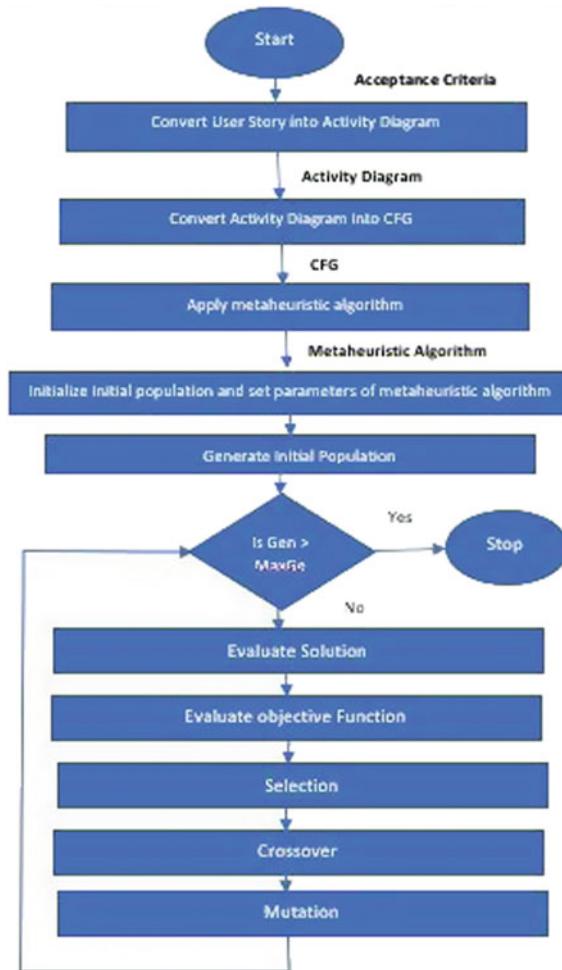
where FAN-IN (A) is a count of the number of other nodes that can call or pass control to node A and FAN-OUT (A) is a number of nodes that are called by node A. IF is calculated for each node of CFG. The weights of the nodes in a path of CFG are added to determine the weight of this path.

4. Apply meta-heuristic algorithm on CFG. The input to meta-heuristic algorithm is CFG.
  - a. Selection: the chromosomes or candidates for initial population are the possible test orders in CFG. In our approach, a gene is a node of CFG and a string of possible nodes form a test order represents a chromosome. The solution to our problem is represented using the permutation encoding technique. For a set of nodes {1, 2, 3, 4, 5, and 6}, a possible chromosome sequence is {2, 1, 4, 6, 3, and 5}. In our case, this means that chromosome {2, 1, 4, 6, 3, and 5} represents a path consisting of corresponding nodes which are not necessarily in a sequence (in order) and starts from start node in CFG. We have considered valid as well as invalid acceptance tests for our case study. Therefore, the number of tests is very large in number. In our case study, the invalid tests are ignored by meta-heuristic algorithms.
  - b. The fitness value of each chromosome (path in CFG) is computed by adding weights of edges in that path. The formula for finding fitness value of each chromosome is given in Eq. (2)

$$F = \sum_{i=1}^n w_i \quad (2)$$

where  $w_i$  is weight of  $i_{th}$  edge in CFG and  $n$  is number of nodes in current path. The weight of each edge  $(i, j)$  is determined by summing IF values of node  $i$  and node  $j$ .

Crossover and mutation are applied on the chromosome having lowest fitness value to produce new chromosome or solution. The crossover probability  $P_c$ , and mutation probability  $P_m$ , is high for bad individuals having a low fitness value and is low for good individuals having a high fitness value. Figure 3 shows the algorithm that we propose in this paper.



**Fig. 3** Algorithm presenting the proposed approach

### 3.1 Case Study: Bank Management System

In this section, we illustrate our proposed approach by means of a case study of a bank management system [1]. For purpose of illustration, we choose the user story for withdrawal of cash from the ATM. The same approach can similarly be applied to remaining user stories of the bank management system to prioritize acceptance tests.

### 3.2 User Story: Account holder withdraws cash [1]

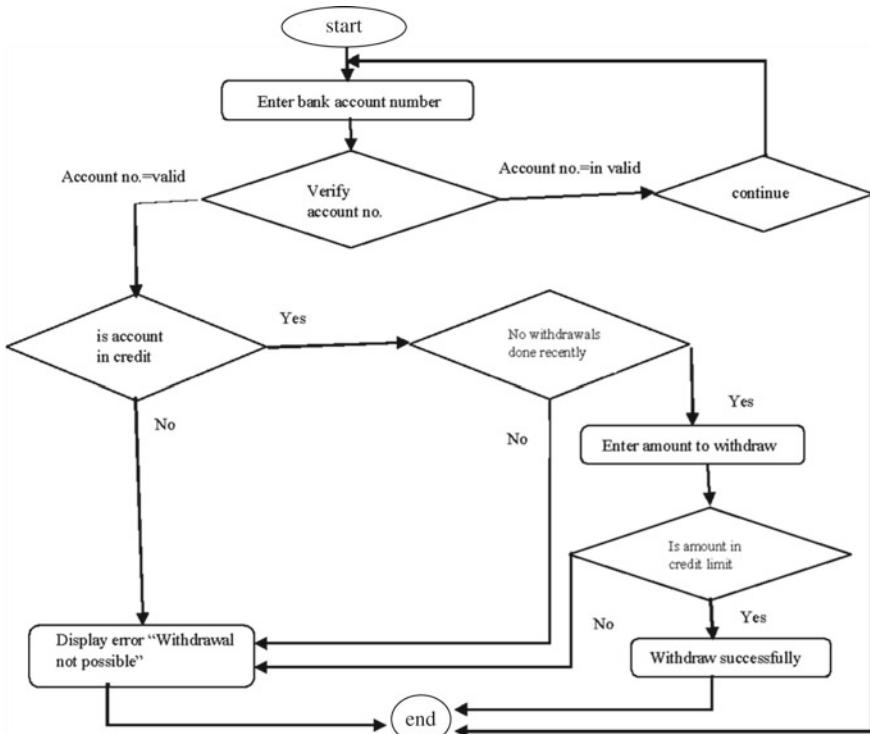
As an account holder, I want to withdraw cash from an ATM so that I can get money when the bank is closed.

### 3.3 Acceptance Criteria

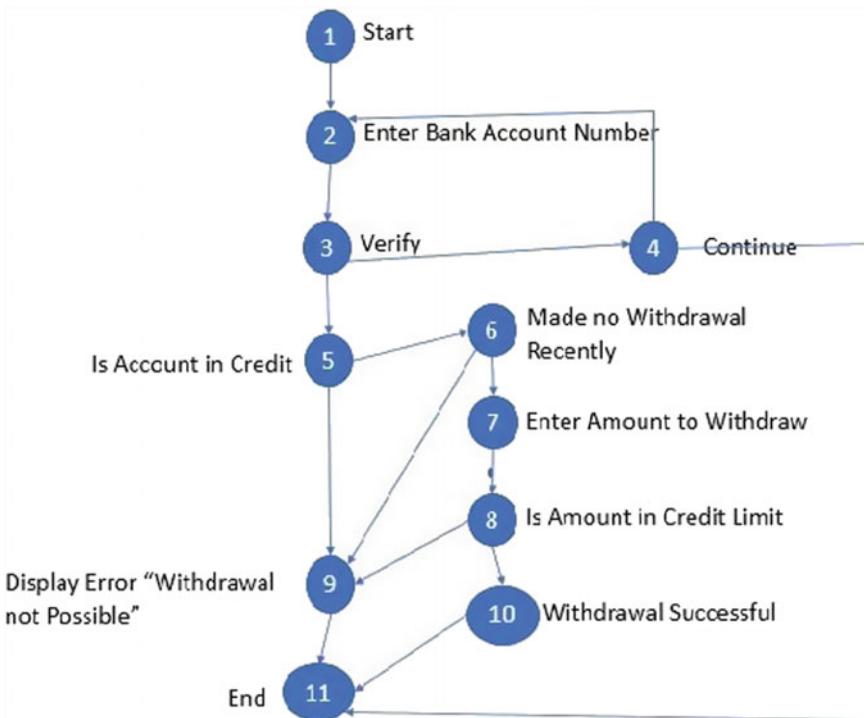
GIVEN my bank account is in credit, and I made no withdrawals recently, WHEN I attempt to withdraw an amount less than my card's limit, THEN the withdrawal should complete without errors or warnings.

An activity diagram is drawn for user story acceptance test written in the GWT format as shown in Fig. 4. The activity diagram for the user story acceptance test in GWT format is then converted into a control flow graph (Fig. 5).

The IF metric values of all the nodes of the CFG are shown in Table 1. For each edge connected by a pair of nodes in CFG, the weight is given by summation of IF



**Fig. 4** Activity diagram for user story acceptance test in GWT format

**Fig. 5** CFG of activity diagram in Fig. 4**Table 1** Information flow (IF) of nodes in CFG

Node	FAN-IN	FAN-OUT	IF=FAN-IN * FAN-OUT
1	0	1	0
2	2	1	2
3	1	2	2
4	1	2	2
5	1	2	2
6	1	2	2
7	1	1	1
8	1	2	2
9	3	1	3
10	1	1	1
11	3	0	0

values of the connecting nodes as shown in Table 2. If no edge exists between a pair of nodes, the corresponding entry in Table 2 is zero. The weight of a test path is determined by summation of the weights of edges in that path. The weight of each edge in CFG is shown in Table 2.

**Table 2** Edge weights connecting nodes in CFG

Nodes	1	2	3	4	5	6	7	8	9	10	11
1	0	2	0	0	0	0	0	0	0	0	0
2	0	0	4	0	0	0	0	0	0	0	0
3	0	0	0	4	4	0	0	0	0	0	0
4	0	4	0	0	0	0	0	0	0	0	2
5	0	0	0	0	0	4	0	0	5	0	0
6	0	0	0	0	0	0	3	0	5	0	0
7	0	0	0	0	0	0	0	3	0	0	0
8	0	0	0	0	0	0	0	0	5	3	0
9	0	0	0	0	0	0	0	0	0	0	3
10	0	0	0	0	0	0	0	0	0	0	1
11	0	0	0	0	0	0	0	0	0	0	0

Let us consider an acceptance test consisting of nodes 1-2-3-5-6-7-8-10-11. The path consisting of these nodes has a weight  $2+4+4+4+3+3+3+1 = 24$ . When applying GA, result shows that acceptance test 1-2-3-4-2-3-5-6-7-8-10-11 has highest weight, i.e., 36 among other tests/paths in CFG. By applying cuckoo search algorithm, results show that acceptance test 1-2-3-5-6-7-8-10-11 has highest weight  $2+4+4+4+3+3+3+1 = 24$ , i.e., 24 among other paths/tests in CFG. By applying micro-GA path, 1-2-3-5-6-7-8-10-11 has highest weight, i.e., 24 among other paths/tests in CFG.

Since the weight of a path is the summation of the IF values of nodes in that path, the meta-heuristics approach helps in determining the path that corresponds to maximum information flow. This is the most critical path that corresponds to the coverage of maximum user story acceptance criteria. Similarly, path having next higher fitness values is a candidate to be tested next. Execution of acceptance tests with higher priorities assures coverage of maximum user story acceptance criteria. The results are shown in Fig. 7 after applying GA, micro-GA, and cuckoo search algorithm (Fig. 6).

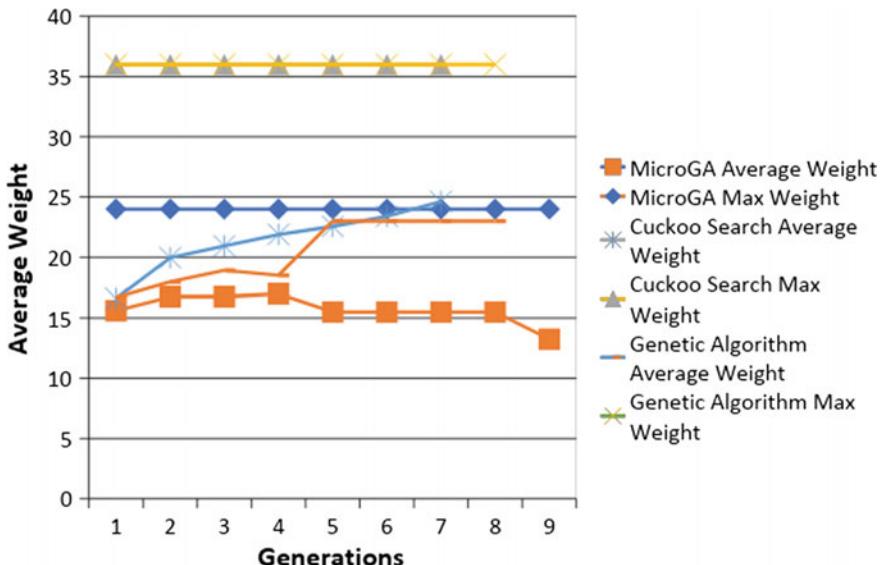
As shown in Fig. 8, cuckoo search algorithm outperforms GA and micro-GA algorithm. The average weight of population in each generation increases by applying cuckoo search algorithm, whereas in GA after four generation average weight of individuals increases sharply and after five generations no improvement in the result is seen, whereas by applying micro-GA, the average weight decreases after five generations and no improvement is seen in the results after third generation.

Parameters	GA	Micro-GA	Cuckoo Search Algorithm
<b>Initial Population Size</b>			
	100	5	100
<b>Selection</b>	Average weight = 20	Elitism	If $\text{Fitness}_{\text{new individual}} > \text{Fitness}_{\text{old individual}}$ . Replace old individual with new individual.
<b>Crossover</b>	Weight > $\text{Fitness}_{\text{Average}}$ , Crossover Probability = 40%	100% for 4 leftover individuals.	-----
	Weight < $\text{Fitness}_{\text{Average}}$ , Crossover Probability = 80%		
<b>Mutation</b>	Weight > $\text{Fitness}_{\text{Average}}$ , Mutation Probability = 15%	-----	Delete 20% worst individuals in each generation.
	Weight < $\text{Fitness}_{\text{Average}}$ , Mutation Probability = 40%		

**Fig. 6** Parameter of meta-heuristic algorithm for proposed approach

MicroGA			Cuckoo Search			Genetic Algorithm		
Gen	Average Weight	Maximum Weight	Gen	Average Weight	Maximum Weight	Gen	Average Weight	Maximum Weight
1	15.6	24	1	16.58	36	1	16.67	36
2	16.75	24	2	19.96	36	2	17.97	36
3	16.75	24	3	20.93	36	3	18.91	36
4	17	24	4	21.89	36	4	18.5	36
5	15.5	24	5	22.56	36	5	23	36
6	15.5	24	6	23.39	36	6	23	36
7	15.5	24	7	24.62	36	7	23	36
8	15.5	24				8	23	36
9	13.25	24						

**Fig. 7** Results obtained after applying GA, micro-GA, and cuckoo search algorithm on proposed approach



**Fig. 8** Comparative analysis of meta-heuristic algorithm

## 4 Conclusion and Future Work

Acceptance test prioritization approach is proposed for agile paradigm in this work. Acceptance tests corresponding to a user story tend to increase in number as the size of a software system increases. Therefore, we propose an approach to prioritize acceptance tests using meta-heuristic techniques, viz. genetic algorithm, micro-GA algorithm, and cuckoo search algorithm. The highest priority acceptance test is one which covers the maximum user acceptance criteria for a user story. We have performed a comparative analysis of meta-heuristic techniques. The results obtained show that cuckoo search outperforms GA and micro-GA. Although these are preliminary findings, nevertheless they provide enough insight regarding the application of these meta-heuristic techniques in this domain. In future, we would like to apply this approach to other real-life problems in agile paradigm to improve the quality of agile processes.

## References

- Pandit, P., & Tahliani, S. (2015). AgileUAT: A framework for user acceptance testing based on user stories and acceptance criteria. *International Journal of Computer Applications* 120(10). <https://doi.org/10.5120/21262-3533>.
- Goldberg, D.E. (1989). *Genetic Algorithms: In search, optimization and machine learning*. Addison Wesley.

3. Mitchell, M. (1996). *An introduction to Genetic Algorithms*. Cambridge: MA, MIT Press.
4. Carlos, A. C., & Pulido, G. T. (2001). A micro-genetic algorithm for multiobjective optimization. In *proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization* (pp. 126–140). [https://doi.org/10.1007/3-540-44719-9\\_9](https://doi.org/10.1007/3-540-44719-9_9).
5. Krishankumar, K. (1989). Micro-genetic algorithm for stationary and non-stationary function optimization. In *SPIE Proceedings, Intelligent Control and Adaptive Systems* (pp. 289–296).
6. Yang, X. S. & Deb, S. (2009). Cuckoo search via Levy flights. In *World Congress on Nature and Biologically Inspired Computing* (pp. 210–214). Coimbatore, India.
7. Valian, E., Tavakoli, S., Mohanna, S., & Haghi, A. (2013). Improved cuckoo search for reliability optimisation problems. *Journal of Computers and Industrial Engineering* 64(1), 459–468. <https://doi.org/10.1016/j.cie.2012.07.011>.
8. Tiwari, S., & Gupta, A. (2015). An approach of generating test requirements for agile software development. In *proceedings of ISEC'15*, Bangalore, Karnataka, India. <https://doi.org/10.1145/2723742.2723761>.
9. Booch, G., Jacobson, I., & Rumbaugh, J. (2005). *UML2 and the Unified Process* (2nd ed.).
10. Aggarwal, K. K., & Yogesh, S. (2007). *Software Engineering* (3rd ed.). New Age International Publishers.

# Software Reliability Assessment Using Deep Learning Technique



Suyash Shukla, Ranjan Kumar Behera, Sanjay Misra  
and Santanu Kumar Rath

## 1 Introduction

In the present day scenario, use of the software is increasing rapidly in every sphere of life. The reliability of software products is of utmost concern from users as well as developers perspective. For improving the reliability of the software, comprehensive use of technologies, as well as associated methodologies, is often advisable. The waterfall model is a well-known model for software development process because of its sequential phases. The requirement of reliability is initially identified at analysis phase as one of the various informal requirements. Then, after analyzing software requirements, efforts are given to obtain a full proof class diagram in such a manner that the final product turns to be a reliable one. Then in the implementation phase of any project, design aspects are transformed into a failure-free executable form. After implementation, testing is performed with the intent of finding errors and then removing those errors. It also checks that whether the software meets its acceptable requirements or not, in such a manner that its possible failure can be brought down to a minimum level as possible. From the developers angle, there are two universally accepted ways for software development,

---

S. Shukla (✉) · R. K. Behera (✉) · S. K. Rath  
Department of Computer Science and Engineering,  
National Institute of Technology Rourkela, Rourkela, Odisha, India  
e-mail: suyashshukla2811@gmail.com

R. K. Behera  
e-mail: jranjanb.19@gmail.com

S. K. Rath  
e-mail: skrath@nitrkl.ac.in

S. Misra  
Department of Electrical and Information Engineering,  
Covenant University, Ota 1023, Nigeria  
e-mail: ssopam@gmail.com; sanjay.misra@covenatuniversity.edu.ng

- Proprietary based
- Open-source based.

The main characteristics of OSS is it can be accessed and modified by a user who has the authority of accessing/modifying the product. However, while managing other quality aspects, issue of security and reliability of software emerges. The development technique of OSSs focuses on parameters of reliability, which can be characterized as the uninterrupted service given by the software for a specified duration in a given situation. Since 1970, a great number reliability growth models (SRGMs) [1, 2] have been proposed. Nonhomogeneous Poisson process (NHPP) is the most generally used among these models because of its ability to describe the phenomenon of software failure. Yamada [2] proposed the first NHPP model. Consequently different variation of the NHPP models has been proposed by several authors. Although these models can be used in wide range of applications, they impose certain restrictions and assumptions. Due to these restrictions, there is not any model which can be used for all software applications.

One possible solution to this problem is to use a neural network. Neural Networks works in a similar way as a human brain. It has the ability to organize its structural constituents called as neurons, which perform computational tasks very quickly of an input layer: where the values are presented to the network, one hidden layer: which processes these input values, and an output layer: from which the output of the network is obtained. The below-described network architecture, however, seems to prove insufficient for solving more complex problems. For complex functions, a shallow network can require a very large number of computational units and equally large amounts of time for training. Problems such as overfitting are also more likely to occur in this case. One possible solution to it can be adding more hidden layers of neurons to the network, increasing the complexity of computation the network, even with the relatively small number of neurons. It is observed that more complex problems can be represented using some extensions of such networks called as deep networks. The deep neural network is better than shallow neural network because as the complexity of task increases, the number of neurons also increases exponentially in case of the shallow network. This study aims at identifying the critical faults caused in a software, where the fault characteristics have been chosen by using deep learning-based technique.

Since input datasets are chosen by deciding the measure of attributes for the target variable in advance, it is not very much convenient to apply the conventional neural network model. But in case of deep learning-based method, the measure of attributes for the target variable is automatically decided.

## 2 Related Work

Recently, there has been a lot of speculation on OSS. The major concern of research still residing on if the behavior of OSS is alike closed source software (CSS) or not. Another major issue is to choose appropriate model or family of models for

software reliability analysis. One of such experiments was carried out by Mohamed et al., who investigated the defect discovery rate of two OSS products with his self-developed software using two SRGM [3]. Upon experiment, it has concluded that the two OSS products have dissimilar profiles of defect discovery. Similar experiments were performed by Zhou et al. for analyzing the bug tracking data of six OSS projects and inferred that along with their developmental cycle, OSS projects does not exhibit identical reliability growth pattern as of closed source projects [4]. Hence, the general Weibull distribution was proposed to model the failure occurrence pattern of OSS projects. This failure occurrence pattern was analyzed for three OSS products by applying SRGM by Rossi et al. [5], who then found out Weibull distribution to be the best model for OSS. This was later contradicted by Rahmani et al. [6], who obtained a contrasting result in an experiment by comparing the prediction and the fitting capabilities of the three models which used the failure data of five OSS projects. From their experiments, it has concluded that Schneidewind model yield best results as compared to Weibull model. Further, Zou et al. modeled the bug reports of six OSS projects using nonparametric techniques which resulted in generalized additive (GA) models and exponential smoothing methods being more suitable for reliability characterization of OSS projects [7]. The experiments performed by Li et al. finally confirms that SRGM can be utilized for reliability characterization of OSS projects Li et al. [8]

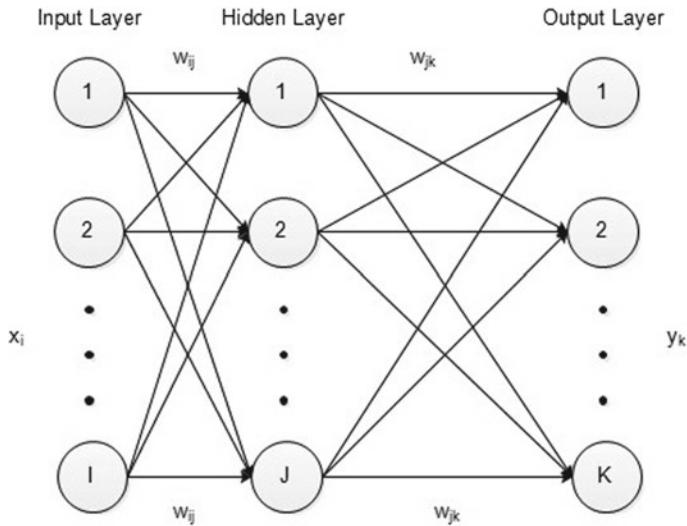
### 3 Bug Tracking System

Fault data of OSS are often that they reused for further development support. Commonly considered bug tracking systems in software industries are may be named as BugLister for proprietary software and Bugzilla for open-source software [9–11]. Among various parameters in the dataset, eight number of parameters have been considered as most important ones for identification of critical faults. These parameters are a date on which the bug is recorded, product name, component name, version, fault reporter, fault assignee, status, and operating system.

Eight types of fault level, i.e., trivial, enhancement, minor, normal, regression, blocker, major, and critical have been applied to the input parameters. Each fault level is represented by a number. Trivial, enhancement, minor, normal, regression, blocker, major, and critical are represented by 1, 2, 3, 4, 5, 6, 7, 8, respectively. In this paper, a prediction model using deep learning has been proposed for identifying the software faults in OSS.

### 4 Identification of Critical Fault Based on Neural Network

Figure 1 is representing the structure of the neural networks used in this paper. Let  $w_{ij}^1 (i = 1, 2, 3, \dots, I; j = 1, 2, 3, \dots, J)$  represents the weights from  $i$ -th neuron of input layer to  $j$ -th neuron of hidden layer, and  $w_{jk}^2 (j = 1, 2, 3, \dots, J;$



**Fig. 1** Structure of neural network

$k = 1, 2, 3, \dots, K$ ) represents the weights from  $j$ -th neuron of hidden layer to  $k$ -th neuron of output layer.

The steps for the identification are as follows:

### Step-1

Since, the data obtained from bug tracking system is the raw data, they cannot be directly provided as input to neural network. The raw data is first transformed to the fault count data which is based on the rate of occurrence. Initially, X will be taken as input matrix and Y will be taken as output matrix.

### Step-2

Then in the second step, weights and biases will be initialized with random values. Let us define:

- $H_w$  is the matrix of weights to the hidden layer
- $H_b$  is the matrix of biases to the hidden layer
- $O_w$  is the matrix of weights to the output layer
- $O_b$  is the matrix of biases to the output layer.

### Step-3

The dot product of input and weight matrix to hidden layer has been taken, and then add matrix of biases at hidden layer.

$$\text{input\_hidden} = X \cdot H_w + H_b \quad (1)$$

Then sigmoid function is used as an activation function to perform the nonlinear transformation, and the output of sigmoid function is  $\frac{1}{1+e^{-x}}$ .

$$\text{activation\_hidden} = \text{sigmoid}(\text{input\_hidden}) \quad (2)$$

**Step-4**

The dot product of activation\_hidden and weight matrix to output layer has been taken, and then add matrix of biases at output layer. Then sigmoid function has been considered as an activation function to predict the output, however depending upon the requirements other activation functions can also be used.

$$\text{input\_output} = \text{activation\_hidden} \cdot O_w + O_b \quad (3)$$

$$\text{output} = \text{sigmoid}(\text{input\_output}) \quad (4)$$

**Step-5**

For the training of input data, multilayered neural network is used with backpropagation learning algorithm. Finally, error in the network is calculated using following equation:

$$\text{Error} = \frac{1}{2} \sum_{k=1}^K (b_k - q_k)^2 \quad (5)$$

where

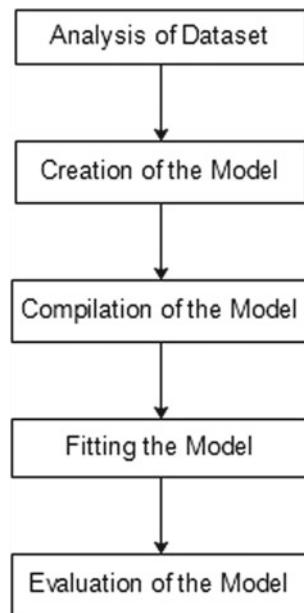
$$q_k (k = 1, 2, 3, \dots, K) \quad (6)$$

are the target input values for the output values. Since eight parameters have been assumed as input, number of neurons in input layers is eight, and only eight kinds of fault levels are considered. So, number of neurons in output layers will be eight only.

## 5 Identification of Critical Fault Based on Deep Learning

Many researchers have proposed various deep learning-based algorithms [12–17] in the past. This paper uses the deep neural network to predict the level of faults in the data obtained from bug tracking system [18]. While considering the problem of identification of faults, it may be observed that it is difficult to apply ANN because the prior knowledge is required for deciding the parameters for the target variable. But in case of based technique, such kind of prior knowledge is not required, the input characteristics for target variable is automatically decided. For developing and evaluating deep neural network model, the most powerful Python libraries are Keras. For performing numerical computation, Keras uses the other libraries such as Theano and TensorFlow. The flowchart for deep learning-based technique using Keras is shown in Fig. 2.

**Fig. 2** Keras: flowchart for deep learning



### 5.1 *Analysis of Dataset*

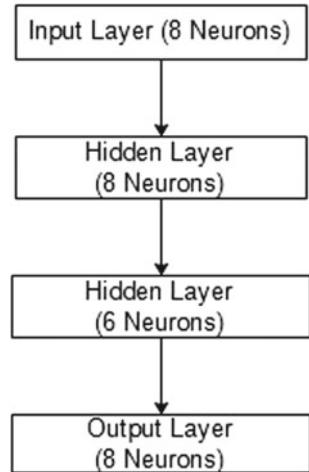
The useful parameters to look are dataset characteristics, attribute characteristics, the total number of rows, and the total number of columns, associated task (classification or regression). The structure is studied thoroughly so that sending it to classifier will become easier.

### 5.2 *Creation of the Model*

In Keras, a model is created using sequential layers. In neural networks, the large number of neurons reside inside several sequential layers. The diagram is shown representing the model that has fully connected layers, which means that all the neurons are connected from one layer to its next layer. To achieve this, dense function in Keras has been utilized.

The above architecture to build our network model is used. Input data is provided to the first hidden layer that has eight neurons. Generally, the number of neurons in the hidden layer is the mean of input neuron and output neurons. The arrangement of the neurons in the proposed model for eight fault level is shown in Fig. 3.

**Fig. 3** Deep neural network model for estimation of fault level in case of eight fault levels



### 5.3 Compilation of the Model

After the model is created, three parameters are identified for compiling the model in Keras. These parameters are loss function, optimizer, and metrics. Loss function could be *binary\_crossentropy* or *categorical\_crossentropy*. *Binary\_crossentropy* is basically used in case of binary classification, but for multiclass classification *categorical\_crossentropy* is used. In the dataset considered, the number of classes to classify is eight; so here *categorical\_crossentropy* is used. The optimizer could be Adam, RMSprop, etc. These are basically used for learning the model. The value of metrics tells the parameter on which the model is evaluated. In this model, the accuracy is taken as a metrics.

### 5.4 Fitting the Model

After the compilation of model, the next step is to fit the dataset with the model. For this, `fit()` function is used in Keras which expects five arguments such as the input training data, the output training classes, testing data used to check the performance of the network, a number of iterations, a number of instances which are evaluated before performing a weight update in the network.

### 5.5 Evaluation of the Model

After fitting the dataset to the model, the final step is evaluating the performance of a model. For this, evaluate () function is used in Keras which expects two arguments as input data and the output data, respectively.

## 6 Dataset

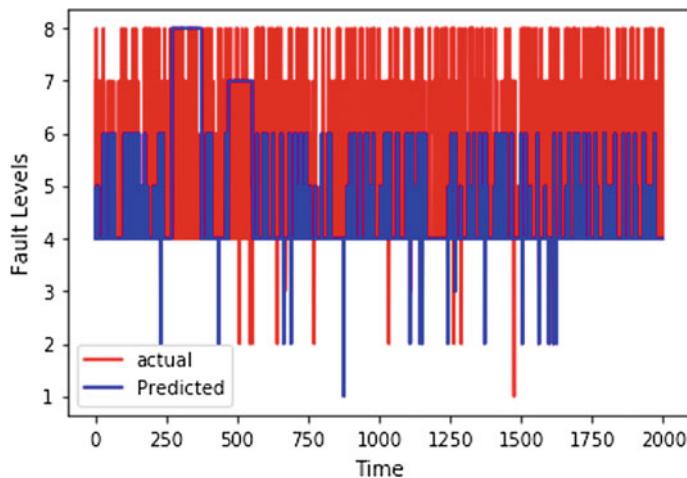
In this section, Firefox and Eclipse datasets have been considered in order to estimate the performance of the proposed technique. The official Web site of Bugzilla has been considered for collecting the bug data recorded on bug tracking system. A total of 10,000 fault datasets have been extracted, and from these collections, 80% of the bug data is considered for training and remaining 20% is used for testing the performance of proposed method.

## 7 Analysis of Result

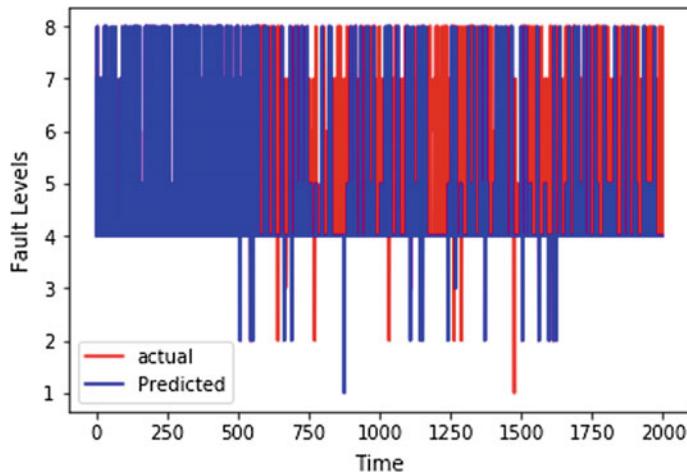
The dataset obtained from Bugzilla are analyzed, and then preprocessing of data is done because using the dataset obtained from bug tracking system directly is a difficult task.

Initially, eight fault levels are applied to the objective variable on bug tracking system. Then, with the help training data of 8000 bugs, we have estimated the fault level of testing data of size 2000 bugs using both the techniques (neural network based and deep learning based), and the results of both the techniques are shown in Fig. 4 and Fig. 5, respectively. From Figs. 4 and 5, it may be observed that the deep learning-based identification method presents a convincing results.

The recognition rate obtained by both the techniques are shown in Table 1. From Table 1, we can say that the recognition rate of deep learning-based technique is much higher than that of neural network based technique. Moreover, to check the efficiency of the proposed method, we have calculated the results in case of only two types of fault levels (critical and noncritical). The eight fault levels are categorized into two categories, where the first category contains trivial, enhancement, minor, normal, regression, blocker, and the second category contains major and critical fault levels. The first category is assigned as index 1, and the second category is assigned as index 2. Figure 6 and Fig. 7 show the estimation results of 2000 testing dataset based on neural network and deep learning by using 8000 learning dataset, respectively. Table 2 shows the comparison results for methods based on neural networks and deep learning in case of two categories. However, it may be observed that the estimation results based on deep learning present high recognition rates.



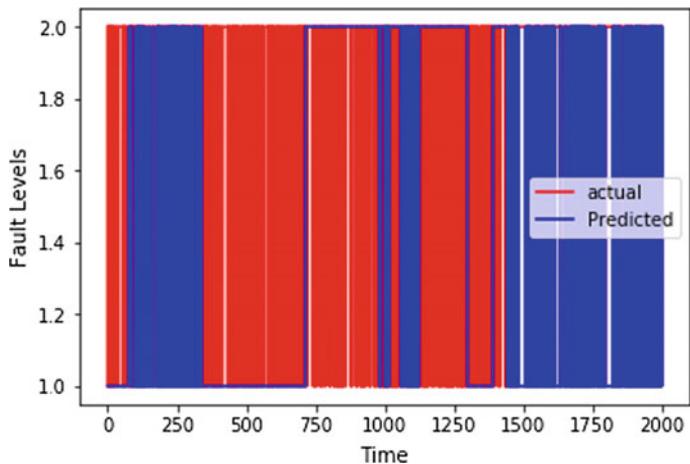
**Fig. 4** Estimation of fault level by neural network



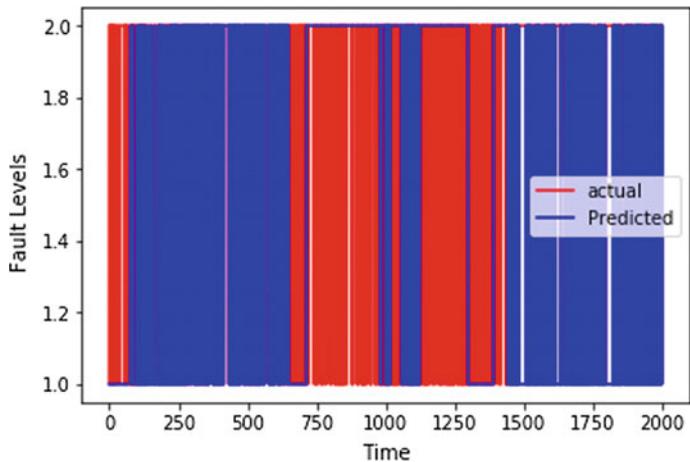
**Fig. 5** Estimation of fault level by deep learning-based technique

**Table 1** Rate of recognition using ANN and deep neural network

Method	Rate of recognition (%)
Neural networks	8.023
Deep learning	49.743



**Fig. 6** Estimation of fault level by artificial neural network in case of second category



**Fig. 7** Estimation of fault level by deep neural network in case of second category

**Table 2** Rate of recognition using ANN and deep neural network in case of second category

Method	Rate of recognition (%)
Neural networks	69.227
Deep learning	75.363

## 8 Conclusion

Currently, many bug tracking systems are available through which the bug data related to an open-source project can be easily obtained. The task of software reliability is majorly divided into two parts: the identification of critical faults and then with the help of those critical faults to assess the reliability of the software system. In this paper, a deep learning-based technique is developed for the identification of critical faults in OSS, and the result of the proposed technique is then compared with the existing technique. It is observed that the result obtained by deep learning-based technique is more convincing than other techniques used for identification of the critical fault in OSS. In the future, the results obtained based on proposed technique can be used by the software managers to assess the reliability of the software project.

## References

1. Lyu, M. R. (Ed.). (1996). *Handbook of software reliability engineering*. Los Alamitos, CA: IEEE Computer Society Press.
2. Yamada, S. (2014). *Software reliability modeling: fundamentals and applications*. Tokyo/Heidelberg: Springer Verlag.
3. Syed-Mohamad, S. M., et al. (2008). Reliability growth of open source software using defect analysis. In: *International Conference on Computer Science and Software Engineering*.
4. Zhou, Y., et al. (2005). Open source software reliability model: an empirical method. In: *ACM SIGSOFT Software Engineering Notes*.
5. Rossi, B., et al. (2010). Modelling failures occurrences of open source software with reliability growth. *Journal of Open Source Software: New Horizons*, 268–280.
6. Rahmani, C., et al. (2010). A comparative analysis of open source software reliability. *Journal of Software*, 1384–1394.
7. Zou, F., et al. (2008). Analyzing and modeling open source software bug report data. In: *19th Australian Conference on Software Engineering*.
8. Li, X., et al. (2011). Reliability analysis and optimal release-updating for open source software. *Information and Software Technology*, 53, 929–936.
9. Yamada, S., & Tamura, Y. (2016). *OSS reliability measurement and assessment*. London: Springer Verlag.
10. Schick, G. J. & Wolverton, R. W. (1978). An analysis of competing software reliability models. *IEEE Transactions on Software Engineering*, SE-4(2), 104–120.
11. Bosio, D., Littlewood, B., Strigini, L., & Newby, M. J. (2002). Advantages of open source processes for reliability: Clarifying the issues. In: *Proceedings of the Open Source Software Development Workshop*, Newcastle, pp. 25–26, 30–46.
12. Kingma, D. P., Rezende, D. J., Mohamed, S., & Welling, M. (2014). Semi-supervised learning with deep generative models. In: *Proceedings of the 27th International Conference on Neural Information Processing Systems*, Montreal, QC, pp. 3581–3859.
13. Blum, A., Lafferty, J., Rwebangira, M. R., & Reddy, R. (2004). Semi-supervised learning using randomized mincuts. In: *Proceedings of the International Conference on Machine Learning*, p. 113. New York, NY: ACM.
14. George, E. D., Dong, Y., Li, D., & Alex, A. (2012). Contextdependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on Audio, Speech and Language Processing*, 20, 30–42.

15. Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., & Manzagol, P. A. (2012). Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11, 3371–3408.
16. Martinez, H. P., Bengio, Y., & Yannakakis, G. N. (2010). Learning deep physiological models of affect. *IEEE Computational Intelligence Magazine*, 8, 20–33.
17. Hutchinson, B., Deng, L., & Yu, D. (2013). Tensor deep stacking networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35, 1944–1957.
18. Tamura, Y., Matsumoto, M., & Yamada, S. (2016). Software reliability model selection based on deep learning. In: *Proceedings of the International Conference on Industrial Engineering, Management Science and Application*, Korea, pp. 77–81. 23–26 May 2016.

# Empirical Validation of OO Metrics and Machine Learning Algorithms for Software Change Proneness Prediction



Anushree Agrawal and Rakesh Kumar Singh

## 1 Introduction

Software products are very gigantic and multifaceted nowadays. Maintenance of these software systems needs immense efforts. Software systems evolve in order to maintain their efficacy after release. Changes do not only impact the particular phase where they are introduced, but its impact propagates to other succeeding phases of SDLC as well [1–3]. This enormously increases the development cost.

Change proneness is the likelihood that a particular part of software would change. Early prediction of change prone classes is helpful in maintenance and testing phases. In this work, we establish a relationship between various object-oriented metrics and change proneness. We also compared the performance of different machine learning methods and statistical technique for successful prediction of change prone classes.

This paper is organized as follows. Section 2 summarizes related work. Section 3 summarizes the research approach including the software metrics, independent and dependent variables studied. Section 4 describes the empirical data collection method. The results of the study are presented in Sect. 5, and Sect. 6 synopsizes the conclusion of the work.

---

A. Agrawal (✉) · R. K. Singh  
Department of Information Technology, Indira Gandhi Delhi Technical University for Women, New Delhi 110006, India  
e-mail: anushreeagrawal.iet@gmail.com

R. K. Singh  
e-mail: rksingh988@gmail.com

## 2 Related Work

Software systems experience a number of changes in order to boost its functionality or to rectify shortcomings. It is very beneficial to identify change prone classes in the early phases of SDLC, and various approaches have been proposed in the literature for the same. Few approaches suggest the use of historical changes from software repositories [4–7], while others suggest the use of software metrics for the purpose [8–10].

The use of reverse engineering tools to collect code metrics is proposed by Sharafat and Tavildari [11] to determine the change probability. A change impact model is proposed by Chaumum et al. [12] to study the effects of software changes. A change impact model was defined according to which the impact of change depends on two main factors, its type and the type of link between classes. Telecommunication system was used to evaluate the impact of change was, and it was concluded that the chosen OO design metrics are decent measures of changeability. Bieman et al. [13] examined five progressing software systems in order to examine the relationship between design patterns and software change proneness. Classes were more change-prone in four out of five software systems. Larger classes were found more change prone after normalizing for class size. The combined effect of addition and modification of classes on the change proneness of OO design is examined by Tsantalis et al. [14]. They studied the stability of design over subsequent software versions, beyond which altering of design cannot be practiced. Zhu and Song [15] studied the relationship between static software metrics and change proneness. Classification methods were used to identify change-prone classes. The results are validated using open-source software system, DataCrow. It indicates that classification methods are beneficial for identifying change-prone classes. Romano and Pinzger [16] studied the effectiveness of source code metrics to predict change prone Java interfaces. The authors have used C&K metrics along with two external cohesion metrics. They have used ten open-source Java software systems to draw conclusions and found that external cohesion metrics had better performance compared to C&K metrics for classification of change prone classes. Lu et al. [17] have employed statistical meta-analysis techniques to study the effectiveness of 62 OO metrics to predict change proneness of classes and evaluated their results on 102 Java systems. They concluded that all size metrics show adequate ability in identifying change-prone classes, while coupling and cohesion metrics have a lower predictive ability. Also, inheritance metrics have a meager ability to predict change-proneness. Elish et al. derived and validated a set of evolution-based software metrics for the identification of change prone classes in object-oriented software systems [18].

Many researchers have studied the relationship between software static attributes and change proneness of classes in software systems. Software metrics have proved to be significant in the identification of change prone classes, but there is no generalized mechanism for metric selection to obtain acceptable results. In this research, we propose a set of software metrics that when used in combination is useful for the identification of change prone classes. We have validated our experiment using open-source software systems.

### 3 Approach

#### 3.1 Independent Variables

Experimental results in the literature suggest that size metrics show moderate predictive power; coupling and cohesion metrics have even lower predictive power, while inheritance measures were found to be less prognostic of change proneness [17]. Hence, we have used 15 software metrics for our experiments which include volume, complexity, and object-oriented metrics [19]. In this study, static software metrics are used as independent variables. We have selected these OO metrics because they are found effective for change proneness prediction studies in previous researches. We have used “Understand for Java” (<https://scitools.com/>) tool to visualize the source code and obtain values for these software metrics. We have described these software metrics in Table 1.

#### 3.2 Dependent Variable

In this study, we have taken change proneness as dependent variable. A class, which is changed in the next version of a system, is called change-prone and not change-prone otherwise. Change can be studied as number of lines of code added, deleted, or modified between two versions of a software system. In this paper, we analyze the relationship between change proneness and various object-oriented software metrics.

#### 3.3 Prediction Model

The statistical and machine learning methods used for prediction of the dependent variable, i.e., change proneness, are discussed here. The performance evaluation measures are also discussed in this section.

**Methods.** In this paper, we have used one statistical method and six machine learning methods for our study. These methods are applied for predicting the dependent variable from the set of independent variables. We have used the LR statistical method and Bayesian network, multilayer perceptron, k-star, bagging, random forest, and PART among the machine learning methods. Among the machine learning methods, we have chosen one method from each class of methods. We have used Bayesian network for probability-based prediction using Bayes theorem. Multilayer perceptron is function-based machine learning algorithm, k-star is an instance-based lazy learning algorithm, random forest is decision tree based, and PART is a rule-based algorithm. Thus, we have chosen machine learning methods with different learning patterns to compare their performance for change proneness prediction.

**Table 1** Software metrics

Serial no.	Metric name	Description
1	Coupling between objects (CBO)	CBO is measured only for object-oriented systems, and it is defined as the number of other classes that a class is coupled to
2	Number of children (NOC)	It is the count of number of immediate subclasses that inherit the class. This gives an idea about the influence of the class (span of control) on software design
3	Number of class method	It is the count of total number methods in a given class
4	Number of class variable	It is the count of total number of variables in a given class
5	Number of instance method (NIM)	It is the count of total number of instance method
6	Number of instance variable (NIV)	It is measured as the total number of instance variable
7	Number of local methods (NLM)	NLM is measured as the total number of local variable of a given class
8	Response for a class (RFC)	The response set (RS) of a class is a set of methods that can potentially be executed in response to a message received by an object of that class
9	Number of local private methods	It is the total number of local private methods which are not inherited by the descendent classes
10	Number of local protected methods	It is the total number of local protected methods which are not inherited by the descendent classes
11	Number of local public methods	It is the total number of local public methods which are inherited by the descendent classes
12	Lines of code (LOC)	The total number of executable lines of code excluding blank lines and comments
13	Depth of inheritance tree (DIT)	DIT is the path length from root node to the farthest leaf node of the inheritance tree. The higher value of DIT denotes a greater number of classes that it inherits, making it complex to predict the class behavior
14	Lack of cohesion in methods (LCOM)	It is the difference between method not having common attribute usage and methods having common attribute usage
15	Weighted method per class (WMC)	WMC is defined as the weighted sum of the complexities of all the methods defined in a class

**Performance Evaluation Measures.** The following performance evaluation measures are used in our study:

*Sensitivity.* It is the total number of true classified change prone classes to the total number of true change prone classes. A higher value signifies that most of the change prone classes have been identified.

*Specificity.* It measures the number of non-change prone classes classified as non-change prone by the prediction model. A high value signifies that most of the non-change prone classes have been identified.

*F-Measure.* F-measure is the harmonic mean of precision and recall (sensitivity).

*Receiver Operating Characteristic (ROC) Analysis.* ROC is a graphical plot of sensitivity on y-axis and 1-specificity on x-axis. Decent performance of classification method is signified by high values of area under curve (AUC) obtained from ROC analysis.

High values of sensitivity and specificity are desired as it indicates that most of the change prone classes are identified correctly by the prediction model [6]. Hence, we have chosen the above-mentioned parameters to evaluate the correctness of our prediction model.

### 3.4 Validation Method

We have used 10-cross-validation method to validate the prediction model. In 10-cross-validation method, we distribute the dataset into ten equal subsets. Out of these ten subsets, one subset is used for testing the data and the remaining nine subsets are used to train the model. The process is repeated for all ten subsets.

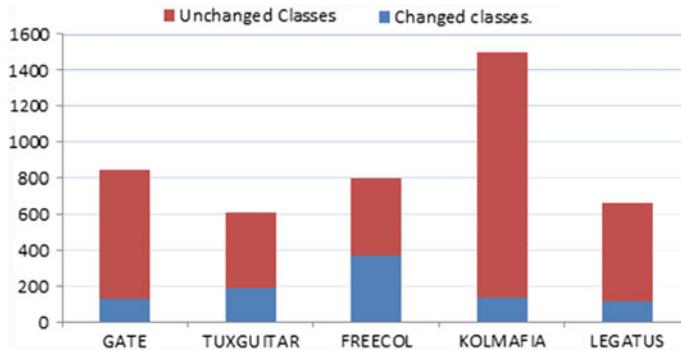
## 4 Empirical Data Collection

In this section, we explain the data collection method used for our experiments. We have arbitrarily selected five open-source software developed in Java from source-forge.net which have a large number of classes so that we have enough data points for our analysis. Table 2 summarizes the details of both the versions of the software systems under study. Figure 1 shows the distribution of change and non-change prone classes.

For the data collection process, we first obtain the software metrics mentioned in Table 1 for each software system using “Understand for Java.” We consider the software metric values of common classes in the two versions for the software systems under study. To generate a change report, we used CMS tool for Java [20]. We have developed CMS tool to generate the change report by comparing two versions of a software system and then comparing the common classes between two versions using diff command. For the comparison, CMS extracts all the common classes between the two versions and preprocesses them by removing any non-executable lines of

**Table 2** Software systems used for experiments

Dataset	Prog. lang.	V1	V2	Changed classes	Unchanged classes	Change %
GATE	Java	3.0	3.1	128	720	15.1
TuxGuitar	Java	1.0	1.1	189	419	31.1
FreeCol	Java	0.11.5	0.11.6	372	429	46.4
KoLmafia	Java	17.4	17.5	138	1358	9.2
Legatus	Java	0.2.6	0.2.7	118	544	17.8

**Fig. 1** Statistics of software systems used for experiments

code in the class. Then with the help of “diff” command, it counts the number of added, deleted, and modified lines among common classes and generates the report for the same. The change report generated by CMS tool counts the total number of changes in each class with the following assumption:

1. Any addition or deletion of an executable line of code to a class in the new version adds one to the change count.
2. Any modification of an executable line of code in a class in the new version adds two to the change count. We assume so because a modification can be viewed as a deletion followed by an addition to the source code.

Since change proneness is a binary variable, we have marked change proneness for all changed classes as “1” and non-changed classes as “0”. Now, we combine the metric and change report to construct our dataset. We repeat the process for all five software systems to construct the five datasets for our experiment.

#### 4.1 Descriptive Statistics

Here, we will describe statistics results of software systems under study. Tables 3, 4, 5, 6, and 7 shown are the descriptive results that contain mean, median, standard

**Table 3** Descriptive statistics for GATE dataset

	Mean	Median	SD	Min.	Max.	Percentiles	
						25	75
CBO	5.57	3	7.15	0	58	1	7
NOC	0.25	0	1.66	0	28	0	0
Cl. me.	0.55	0	3.27	0	59	0	0
Cl. var.	1.01	0	3.11	0	61	0	1
NIM	7.44	4	10.01	0	85	2	9
NIV	11.43	5	17.77	0	146	2	11
Loc. me.	0.21	0	0.76	0	10	0	0
RFC	0.64	0	2.68	0	32	0	0
Lo. Pri. Met.	0.59	0	2.20	0	37	0	0
Lo. Pro. Me.	5.99	3	7.80	0	56	2	7
Lo. pu. me.	120.44	42	239.96	0	2778	20.25	111
LOC	5.83	4	9.73	0	160	2	6.75
DIT	1.92	2	1.05	1	7	1	2
LCOM	41.74	48.5	38.07	0	100	0	80
WMC	19.49	7	37.93	0	509	4	19

deviation, minimum, maximum, and percentiles for each metrics of all software used. From Tables 3, 4, 5, 6, and 7, we can see that the mean value of NOC and DIT is very low, which indicates that inheritance is not much used in these datasets. LCOM metric has greater value indicating that classes are less dependent on each other in terms of attributes and variable usage.

## 5 Result Analysis

### 5.1 Univariate LR Analysis Results

Univariate analysis is performed to extract significant and insignificant metrics. SPSS tool is used for conducting univariate analysis. It examines the singular effect of each independent variable on the dependent variable. The univariate results of software GATE, TuxGuitar, FreeCol, KoLmafia, and Legatus are shown in Tables 8, 9, 10, 11, and 12.

We classify the significant and insignificant metrics based on metric sig. value. If sig. value is less than or equal to 0.01, then that metric is significant; otherwise, metric is insignificant. So from Tables 8, 9, 10, 11, and 12, we can see that CBO, RFC, LOC, LCOM, and WMC metrics are significant in all the datasets, which is also concluded by Malhotra et al. for change proneness prediction using six datasets (developed using C++ and Java) [21]. NIM, NIV, number of local private methods,

**Table 4** Descriptive statistics for TuxGuitar dataset

	Mean	Median	SD	Min.	Max.	Percentiles	
						25	75
CBO	6.67	4	8.65	0	124	1	10.00
NOC	0.42	0	5.12	0	124	0	0.00
Cl. me.	0.46	0	1.37	0	12	0	0.00
Cl. var.	1.64	1	5.11	0	73	0	1.00
NIM	8.70	5	12.06	0	128	3	9.00
NIV	3.31	2	6.68	0	120	0	4.00
Loc. me.	9.15	5	11.98	0	128	3	9.00
RFC	16.25	11	16.37	0	147	5	26.00
Lo. Pri. Met.	1.77	0	4.78	0	36	0	1.00
Lo. Pro. Me.	0.87	0	2.20	0	26	0	1.00
Lo. pu. me.	6.51	4	10.24	0	125	2	7.00
LOC	89.42	46	124.37	4	902	20	98.75
DIT	1.63	1	0.85	1	3	1	2.00
LCOM	56.41	65	29.21	0	100	42	79.00
WMC	17.19	8	28.29	0	219	4	16.00

**Table 5** Descriptive statistics for FreeCol dataset

	Mean	Median	SD	Min.	Max.	Percentiles	
						25	75
CBO	9.84	6	13.53	0	137	2	11
NOC	0.72	0	4.12	0	64	0	0
Cl. me.	1.21	0	4.25	0	45	0	1
Cl. var.	2.57	1	7.77	0	109	0	2
NIM	10.34	4	21.14	0	226	2	11
NIV	2.6	1	4.4	0	48	0	3
Loc. me.	11.55	5	21.8	0	233	2	12
RFC	40.64	26	56.11	0	415	6	37
Lo. Pri. Met.	1.29	0	3.99	0	51	0	1
Lo. Pro. Me.	0.81	0	1.89	0	22	0	1
Lo. pu. me.	9.2	4	18.86	0	218	2	9
LOC	141.08	62	286.19	1	3012	23	138
DIT	2.52	2	1.29	1	7	2	3
LCOM	49.11	50	33.1	0	100	19	79
WMC	29.42	11	66.37	0	685	4	25

**Table 6** Descriptive statistics for KoLmafia dataset

	Mean	Median	SD	Min.	Max.	Percentiles	
						25	75
CBO	7.84	4	13.32	0	228	2	8.00
NOC	0.67	0	6.87	0	206	0	0.00
Cl. me.	3.89	0	18.75	0	492	0	2.00
Cl. var.	5.60	0	69.69	0	2622	0	2.00
NIM	3.96	2	6.12	0	83	1	4.00
NIV	1.90	0	6.03	0	55	0	1.75
Loc. me.	7.84	3	19.80	0	492	2	8.00
RFC	35.73	18	44.86	0	492	8	43.75
Lo. Pri. Met.	1.34	0	4.78	0	73	0	0.00
Lo. Pro. Me.	0.14	0	0.50	0	8	0	0.00
Lo. pu. me.	6.35	3	17.85	0	478	2	7.00
LOC	235.98	57	588.60	0	9974	23	157.00
DIT	2.29	2	1.04	1	6	2	3.00
LCOM	33.20	0	38.54	0	100	0	75.00
WMC	31.37	7	108.14	0	2351	3	20.00

**Table 7** Descriptive statistics for Legatus dataset

	Mean	Median	SD	Min.	Max.	Percentiles	
						25	75
CBO	5.69	3.00	9.22	0	82	1	6
NOC	0.51	0.00	2.06	0	34	0	0
Cl. me.	0.29	0.00	1.31	0	26	0	0
Cl. var.	2.45	1.00	6.43	0	130	0	2
NIM	13.14	9.00	15.71	0	134	4	16
NIV	5.10	3.00	7.45	0	57	1	6
Loc. me.	13.43	9.00	15.75	0	134	5	16
RFC	23.18	16.00	26.72	0	192	8	29
Lo. Pri. Met.	2.61	0.00	6.47	0	60	0	2
Lo. Pro. Me.	0.48	0.00	1.76	0	24	0	0
Lo. pu. me.	10.32	7.00	11.10	0	103	4	13
LOC	151.80	73.00	228.45	3	2069	36	158
DIT	1.79	2.00	0.85	1	5	1	2
LCOM	66.97	75.50	27.84	0	100	58	87
WMC	26.18	13.00	41.44	0	402	6	27

**Table 8** Univariate LR analysis result for GATE dataset

	B	S.E.	Sig.	Exp(B)
CBO	0.080	0.012	0.000	1.084
NOC	0.065	0.044	0.140	1.067
No. of class method	0.049	0.023	0.034	1.051
No. of class variable	0.127	0.035	0.000	1.135
NIM	0.062	0.009	0.000	1.064
NIV	0.031	0.005	0.000	1.032
No. of local methods	0.206	0.099	0.037	1.229
RFC	0.104	0.029	0.000	1.110
No. of local private methods	0.094	0.037	0.011	1.099
No. of local protected methods	0.078	0.011	0.000	1.081
No. of local public methods	0.002	0.000	0.000	1.002
LOC	0.059	0.013	0.000	1.061
DIT	0.271	0.080	0.001	1.311
LCOM	0.020	0.003	0.000	1.020
WMC	0.017	0.003	0.000	1.017

**Table 9** Univariate LR analysis result for TuxGuitar dataset

	B	S.E.	Sig.	Exp(B)
CBO	0.140	0.016	0.000	1.150
NOC	0.128	0.083	2.379	1.137
No. of class method	-0.104	0.075	0.165	0.901
No. of class variable	0.063	0.025	0.011	1.065
NIM	0.091	0.012	0.000	1.096
NIV	0.110	0.021	0.000	1.116
No. of local methods	0.092	0.012	0.000	1.096
RFC	0.043	0.007	0.000	1.044
No. of local private methods	0.230	0.040	0.000	1.258
No. of local protected methods	0.124	0.043	0.004	1.132
No. of local public methods	0.065	0.013	0.000	1.067
LOC	0.009	0.001	0.000	1.009
DIT	-0.192	0.107	0.072	0.825
LCOM	0.015	0.003	0.000	1.015
WMC	0.042	0.006	0.000	1.043

**Table 10** Univariate LR analysis result for FreeCol dataset

	B	S.E.	Sig.	Exp(B)
CBO	0.022	0.006	0.000	1.022
NOC	-0.013	0.018	0.465	0.987
No. of class method	0.032	0.018	0.078	1.033
No. of class variable	0.016	0.010	0.120	1.016
NIM	0.008	0.004	0.029	1.009
NIV	-0.006	0.016	0.723	0.994
No. of local methods	0.009	0.004	0.014	1.009
RFC	0.004	0.001	0.002	1.004
No. of local private methods	0.066	0.024	0.005	1.069
No. of local protected methods	0.083	0.040	0.036	1.086
No. of local public methods	0.010	0.005	0.025	1.010
LOC	0.001	0.000	0.006	1.001
DIT	0.298	0.057	0.000	1.347
LCOM	0.013	0.002	0.000	1.013
WMC	0.004	0.001	0.004	1.004

**Table 11** Univariate LR analysis result for KoLmafia dataset

	B	S.E.	Sig.	Exp(B)
CBO	0.052	0.006	0.00	1.053
NOC	0.013	0.008	0.13	1.013
No. of class method	0.063	0.008	0.00	1.065
No. of class variable	0.047	0.007	0.00	1.049
NIM	0.038	0.01	0.00	1.039
NIV	0.030	0.01	0.003	1.030
No. of local methods	0.053	0.006	0.00	1.054
RFC	0.012	0.002	0.00	1.013
No. of local private methods	0.069	0.014	0.00	1.071
No. of local protected methods	0.299	0.127	0.019	1.348
No. of local public methods	0.072	0.008	0.00	1.075
LOC	0.001	0.00	0.00	1.001
DIT	0.298	0.057	0.000	1.347
LCOM	0.013	0.002	0.000	1.013
WMC	0.004	0.001	0.004	1.004

**Table 12** Univariate LR analysis result for Legatus dataset

	B	S.E.	Sig.	Exp(B)
CBO	0.081	0.012	0.000	1.085
NOC	0.096	0.044	0.030	1.101
No. of class method	-0.191	0.167	0.253	0.826
No. of class variable	0.089	0.022	0.000	1.094
NIM	0.055	0.008	0.000	1.056
NIV	0.114	0.016	0.000	1.121
No. of local methods	0.054	0.008	0.000	1.055
RFC	0.020	0.004	0.000	1.021
No. of local private methods	0.149	0.019	0.000	1.160
No. of local protected methods	0.200	0.059	0.001	1.222
No. of local public methods	0.047	0.009	0.000	1.048
LOC	0.004	0.001	0.000	1.004
DIT	0.327	0.113	0.004	1.386
LCOM	0.026	0.006	0.000	1.026
WMC	0.019	0.003	0.000	1.019

number of local public methods, and DIT are found significant in four out of five datasets.

Lu et al. conducted a study on 102 Java datasets using statistical meta-analysis technique to investigate the ability of 62 object-oriented metrics for change proneness prediction. The study consists of size, cohesion, coupling, and inheritance metrics [17]. We used different datasets developed in same programming language and included complexity metric (WMC) for our experiments. They found that size metrics are good predictors of change proneness prediction, and we too obtained a similar result. LOC was found significant for change proneness prediction. In our experiments, cohesion and coupling metrics (i.e., CBO, RFC and LCOM) were found very significant while Lu et al. found that cohesion and coupling metrics had lower predictive ability [17]. Inheritance metrics (NIM, NIV, and DIT) were also found significant predictors in our results, which are contrasting when compared to the literature [17].

## 5.2 Model Evaluation Using ROC Curve

We evaluate the best prediction model using ROC analysis. The model having largest area under curve (AUC) is able to identify most change prone classes in a dataset. Tables 13, 14, 15, 16, and 17 show the validation result on GATE, TuxGuitar, FreeCol, KoLmafia, and Legatus datasets, respectively. We have applied six machine learning methods and one statistical method on datasets under study. Bagging and random

**Table 13** Model evaluation result for GATE dataset

GATE	AUC	Cutoff Point	Sensitivity	Specificity	F-measure
BayesNet	0.717	0.047	76.9	80.9	78.9
MLP	0.702	0.119	84.4	96.5	81.2
K-star	0.625	0.036	80.2	91.2	78.5
Bagging	0.74	0.142	84.1	97.1	80.0
RF	0.699	0.102	84.6	96.2	81.6
PART	0.707	0.119	83.8	95.9	80.7
LR	0.722	0.138	84.3	97.6	79.9

**Table 14** Model evaluation result for TuxGuitar dataset

TuxGuitar	AUC	Cutoff Point	Sensitivity	Specificity	F-measure
BayesNet	0.756	0.207	75.5	88.5	74.2
MLP	0.75	0.244	75.5	97.1	71.0
K-star	0.793	0.224	77.3	89.0	76.3
Bagging	0.799	0.305	77.1	91.6	75.5
RF	0.786	0.329	74.5	81.6	74.5
PART	0.767	0.393	73.8	86.8	72.6
LR	0.735	0.265	76.5	96.8	72.6

**Table 15** Model evaluation result for FreeCol dataset

FreeCol	AUC	Cutoff Point	Sensitivity	Specificity	F-measure
BayesNet	0.709	0.512	64.8	62.7	64.8
MLP	0.667	0.479	64.0	68.0	64.0
K-star	0.753	0.422	69.3	74.5	69.2
Bagging	0.762	0.454	69.0	74.1	68.9
RF	0.758	0.475	69.2	66.6	69.2
PART	0.72	0.459	65.4	69.9	65.3
LR	0.65	0.461	61.4	72.5	60.8

forest models show higher AUC values than other models in three and two datasets, respectively as shown in Fig. 2. We have also applied nonparametric test on the obtained results to compare the obtained results.

### 5.3 Friedman Test Result

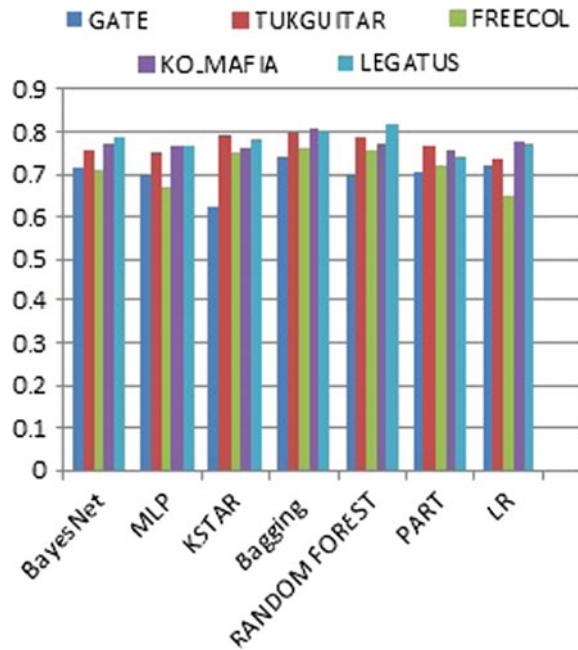
We have analyzed change prediction model for five datasets using six machine learning methods and LR technique. To analyze the statistical significance of the difference in performance of these methods, we have applied Friedman statistical test. Table 18

**Table 16** Model evaluation result for KoLmafia dataset

KoLmafia	AUC	Cutoff Point	Sensitivity	Specificity	F-measure
BayesNet	0.769	0.019	85.1	88.5	86.7
MLP	0.764	0.049	91.4	99.7	88.3
K-star	0.759	0.042	91.0	98.3	89.0
Bagging	0.809	0.054	91.1	98.7	88.8
RF	0.772	0.055	90.5	96.4	89.6
PART	0.754	0.055	91.0	99.3	87.8
LR	0.775	0.094	91.6	99.4	88.9

**Table 17** Model evaluation result for Legatus dataset

Legatus	AUC	Cutoff Point	Sensitivity	Specificity	F-measure
BayesNet	0.787	0.068	84.1	89.8	84.3
MLP	0.766	0.086	85.5	97.4	83.0
K-star	0.781	0.083	84.1	93.9	82.8
Bagging	0.804	0.125	85.5	95.4	84.0
RF	0.815	0.183	84.7	92.7	84.1
PART	0.743	0.104	85.6	94.4	84.6
LR	0.772	0.140	84.6	97.9	81.2

**Fig. 2** ROC analysis

**Table 18** Friedman test results

Algorithm	Mean rank
Bagging	1.20
Random forest	3.00
BayesNet	4.00
K-star	4.40
LR	4.60
PART	5.20
MLP	5.60

lists the mean ranks of each method obtained from Friedman test. The test is applied based on the AUC measure of the algorithms.

The results suggest that bagging is the best algorithm when applied to these five datasets with mean rank of 1.20. Three other machine learning algorithms, i.e., random forest, Bayesian network, and k-star, also perform better than LR. MLP is the worst algorithm with mean rank of 5.60. Our results contradict few studies in the literature [22] where machine learning methods outperformed LR in change proneness prediction.

The Friedman statistical value with six degrees of freedom was calculated as 14.314. A p value of 0.026 signifies that the results are accurate with the confidence interval of 95%. Thus, we reject the null hypothesis, which states that all the algorithms are the same in their prediction behavior.

## 6 Conclusion

The relationship between OO metrics and change proneness of a class is studied in this paper. We have empirically analyzed and compared the performance of LR and machine learning methods for change proneness prediction in software systems. We performed the analysis using five freely available Java software systems and took the changes in classes into account while classifying the change prone classes. We found that few machine learning methods are superior to LR in terms of predictive power for change proneness prediction.

## References

1. Han, A. R., Jeon, S., Bae, D., & Hong, J. Behavioral dependency measurement for change proneness prediction in UML 2.0 design models. In *32nd Annual IEEE International Conference on Computer Software and Applications*.
2. D'Ambros, M., Lanza, M., & Robbes, R. (2009). On the relationship between change coupling and software defects. In *16th Working Conference on Reverse Engineering* (pp. 135–144).

3. Singh, Y., Kaur, A., & Malhotra, R. (2010). Empirical validation of object-oriented metrics for predicting fault proneness. *The Software Quality Journal*, 18(1), 3–35.
4. El Emam, K., Benlarbi, S., Goel, N., & Rai, S. N. (1999). A validation of object-oriented metrics. Technical Report ERB-1063, NRC.
5. Zhou, Y., Leung, H., & Xu, B. (2009). Examining the potentially confounding effect of class size on the associations between object oriented metrics and change proneness. *IEEE Transactions on Software Engineering*, 35(5), 607–623.
6. Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society, Series A*, 36, 111–114.
7. Briand, L., Wust, J., Daly, J., & Porter, D. V. (2000). Exploring the relationships between design measures and software quality in object-oriented systems. *Journal of Systems and Software*, 51(3), 245–273.
8. Erlikh, L. (2000). Leveraging legacy system dollars for e-business. *IT Professional*, 2(3), 17–23.
9. Cartwright, M., & Shepperd, M. (2000). An empirical investigation of an object-oriented software system. *IEEE Transactions on Software Engineering*, 26(8), 786–796.
10. Elish, M., & Al-Khiaty, M. (2013). A suite of metrics for quantifying historical changes to predict future change-prone classes in object-oriented software. *The Journal of Software: Evolution and Process*, 25(5), 407–437.
11. Sharafat, A. R., & Tavildari, L. (2007). Change prediction in object oriented software systems: A probabilistic approach. In *11th European Conference on Software Maintenance and Reengineering*.
12. Chaumum, M. A., Kabaili, H., Keller, R. K., & Lustman, F. (1999). A change impact model for changeability assessment in object oriented software systems. In *Third European Conference on Software Maintenance and Reengineering* (p. 130).
13. Bieman, J., Straw, G., Wang, H., Munger, P. W., & Alexander, R. T. (2003). Design patterns and change proneness: An examination of five evolving systems. In *The Proceeding of 9th International Software Metrics Symposium*.
14. Tsantalis, N., Chatzigeorgiou, A., & Stephanides, G. (2005). Predicting the probability of change in object oriented systems. *IEEE Transactions on Software Engineering*, 31(7), 601–614.
15. Zhu, X., & Song, Q. (2013). Automated identification of change-prone classes in open source software projects. *Journal of Software*, 8(2).
16. Romano, D., & Pinzger, M. (2011). Using source code metrics to predict change prone java interfaces. In *27th IEEE International Conference on Software Maintenance*.
17. Lu, H., Zhou, Y., Xu, B., Leung, H., & Chen, L. (2012). The ability of object-oriented metrics to predict change-proneness: a meta-analysis. *Empirical Software Engineering*, 17(3).
18. Elish, M. O., & Al-Khiaty, M. A.-R. A suite of metrics for quantifying historical changes to predict future change-prone classes in object-oriented software.
19. Chidamber, S. R., Darcy, D. P., & Kemerer, C. F. (1998). Managerial use of metrics for object-oriented software: An exploratory analysis. *IEEE Transactions on Software Engineering*, 24(8), 629–639.
20. Malhotra, R., & Agrawal, A. (2014). CMS tool. *ACM SIGSOFT Software Engineering Notes*, 39(1), 1–5.
21. Malhotra, R., & Khanna, M. (2104). Analyzing software change in open source projects using artificial immune system algorithms. In *International Conference on Advances in Computing, Communications and Informatics (ICACCI)*.
22. Tripathi, A., & Sharma, K. (2015). Improving software quality based on relationship among the change proneness and object oriented metrics. In *2nd IEEE International Conference on Computing for Sustainable Global Development (INDIACoM)*.

## **Part II**

# **Data Management**

**Dr. Anand Gupta Section Editor**

### **Editorial**

With improvements in data acquisition technology and data storage capacity, data management faces the challenge of rapid expansion of data and the concomitant improvement in terms of adaptability and extensibility. This section presents a compendium of research works that address the challenges of data storage, data processing for health care, data routing and secure transmission of data.

Agarwal and Nalavade propose a non-volatile cache (NVC) system that utilizes a solid-state device (SSD) layer to retain active data set within the memory and SSD in a combined manner. This fulfils the objective of storing warm data that might later be deemed as hot, in such a manner that the throughput, which earlier depended on disk read/write latencies, is now predominantly dependent on SSD latencies. The authors show that the proposed NVC design uses minimal SSD capacity to fit an active data set and its throughput performs close to all SSD solutions.

Yadav et al. showcase a cloud-based telemedicine application that enables real time updation of patients' medical records and remote analysis of health data. This provides a widely sharable platform that is accessible by doctors and other professionals to provide medical aid to people living in remote areas who are otherwise unable to get good medical facilities.

Sadasivam et al. propose a honeynet architecture for a network that is prone to distributed brute force attacks. They give a methodology to detect individual botnets from a set of password-guessing attacks and analyse the detailed behaviour of such attacks on Secure Shell services. This paves the way for understanding the nature of such attacks at a deeper level so as to counter them effectively.

Gupta et al. present a survey of various existing data routing techniques for wireless sensor networks (WSNs) for terrestrial and underwater applications. They identify their advantages and disadvantages along with different parameters including QoS, security, energy balancing, energy consumption and mobility. The analysis helps researchers identify the shortcomings of existing WSN data routing techniques and refine them according to application-specific requirements.

Kumar et al. present an interesting analysis and graphical depiction of the spread of epidemics and rumours over networks that model complex real-world systems. They study the relationship between various factors which affect epidemic and rumour spread and their arrest. The authors calculate the particular recovery rate for a given infection rate so as to arrest the spread of epidemic or rumour in a timely and effective manner.

Kumar Abhishek et al. gather data on infections from physical–cyber–social systems to analyse the patterns of infection according to location and generate a heuristic-based opinion. This helps in generating timely alerts on the level of infection in different locations. This makes evident the correlations between different events that probably caused the disease to spread.

### **Section Reviewers:**

Anand Gupta

Anil Goel

Anjali Thukral

Bharti Suri

Chittaranjan Hota

Deepika Prakash

Dilum Bandara

Indu Singh

Gagandeep Kaur

Gaurav Saxena

Geeta Rani

Goldie Gabrani

Hardeo Thakur

Krishana Kumar

Kuldeep Kumar

Mahendra

Mala Saraswat

Manju

Manpreet Kaur

Manusheel Gupta

Monadhika Sharma

Mukta Goel

Namita Gupta

Naveen Kumar

Payal Khurana

Priti Bansal

Raghu

Rolly Bansal

Ruchi Sharma

Sangeeta Srivastva

Shelly Sachdeva

Shikha Mehta

Shikha Gupta  
Shivani Batra  
S. K. Jain  
Sunny Rai  
Sanjay H. A.  
Srishti  
Vandana Bhattacharya  
Vidhi Khanduja

# Extending Database Cache Using SSDs



Prateek Agarwal and Vaibhav Nalawade

## 1 Introduction

OLTP databases store data on disk in form of pages and use caching techniques to retain important pages in memory. Locality of reference in typical database workloads makes sure access is limited to a data set much smaller than entire database. As a consequence, cache size has to be dynamically adjusted to match active data set size. For workloads with larger active data set, it could be a limitation as memory is expensive and adding more RAM to the host system is not always easy. Our non-volatile cache (NVC) design introduces solid-state device (SSD) layer to retain active data set within memory and SSD combined. NVC is second-level cache between main memory cache and hard disk, and it uses SSD for its storage.

Buffer cache retains currently accessed and recently accessed pages usually referred to as hot pages in memory. Thus, when active data set grows large, some warm pages have to be removed from memory. These pages could become hot again due to change in access patterns. Throughput of such operations depends on HDD read latencies. Pages changed by transactions have to be persisted, so throughput of these operations depends on HDD write latencies. NVC caches both types of pages in SSDs so the throughput now is determined by SSD latencies.

An alternative solution is to replace HDDs with SSDs entirely, to speed up I/Os. The drawback here is higher costs due to SSD prices. This approach could be tuned to use SSDs only for hot tables. Administrator should determine which tables are currently hot and move them from HDDs to SSDs. This is not an easy task to administer as it needs constant monitoring, and also bulk data movement would be time-

---

P. Agarwal (✉) · V. Nalawade  
SAP Labs India, Pune, India  
e-mail: prateek.agarwal@sap.com

V. Nalawade  
e-mail: vaibhav.nalawade@sap.com

consuming. Table-level hotness can be misleading because only a small segment of a table might be hot during a time period.

NVC design uses minimal SSD capacity to the active data set, so it is less costly than all SSD solutions. Transactions write to SSDs only and read mostly from SSDs; hence, NVC throughput comes out very close to all SSD solutions. Since the caching algorithms take care of keeping warm data in SSDs, no user intervention is required to move data when access patterns change. However, administrator can always tune NVC size and thresholds to suit their needs. NVC feature is available in SAP ASE 16.0 SP02 [1] and later releases.

Our experiments conducted performance benchmark runs on following three scenarios in order to demonstrate efficacy of NVC. In later sections, we would compare details of benchmark and analyze the results on these scenarios:

**HDD store:** Traditional database setup with HDD-based page store.

**SSD store:** Same as HDD store but SSDs being used instead of HDDs.

**NVC:** Same as HDD store but with a NVC much smaller than SSD store.

## 2 Configuring NV Cache

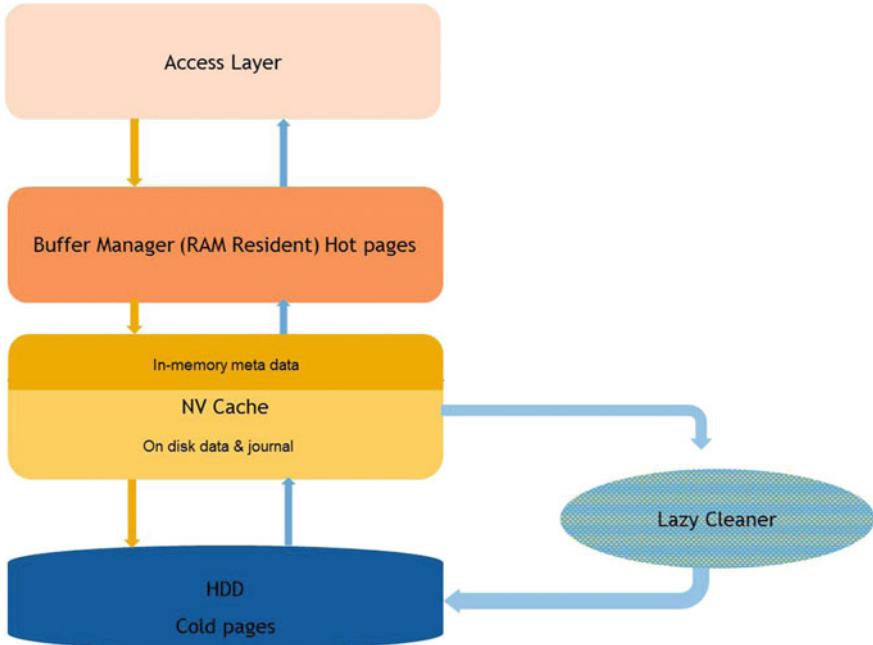
An administrator can configure ASE to create and configure NVC without change in client-side application. NVC can be dynamically added, removed, or tuned. Host machine should have adequate sized SSD and relatively small amount of memory for NVC's meta-data management.

## 3 NV Cache Design

Figure 1 shows block diagram of important components from SAP ASE's architecture from the perspective of NVC. The access layer requests pages from buffer cache to operate on user queries/transactions. In this section, we will briefly describe SAP ASE's buffer cache layer and then explain internals of NV cache component and the rationale behind key design decisions.

### 3.1 Buffer Cache Description

Buffer cache layer stores pages in memory and is responsible for fetching pages from disk. It uses LRU algorithm to retain hot data in memory. User transactions mark modified pages as dirty but do not necessarily push pages to disk. Page writes are handled internally based on cache tuning parameters. Only exception to this is commit or checkpoint which requires log or data persistence, respectively. Buffer cache layer



**Fig. 1** Block diagram

interacts with NVC during page read, write, and replacement, as described in later sections.

### 3.2 Data Layout

NVC stores page data in SSD, and meta-data is stored in memory as data structures and in SSD as meta-data pages. During runtime, NVC writes meta-data pages, and during reboot, they are read to recreate the data structures and re-store NVC before database recovery is performed. Meta-pages are stored in SSD together with data pages and occupy less than 1% space. NVC uses SSD persistence to ensure data durability upon server crash or restart.

Following data structures are used to manage NVC:

1. Hash map: To locate page on SSD file based on logical page number
2. LRU queue: Queue with page ordered by NVC read/write time
3. Dirty status: Page level flag to denote if a page has to be written to HDD

```

1  getpage(LPAGENO)
2      BUF = buf_search(LPAGENO)
3      if BUF != NULL
4          return BUF
5      BUF = grab(LPAGENO)
6      NVCBUF = nvc_search(LPAGENO)
7      if NVCBUF != NULL
8          read(NVCBUF.file, NVCBUF.offset, BUF)
9          return BUF
10     read(BUF.file, BUF.offset, BUF)
11     return BUF

```

**Fig. 2** Page search in presence of NV cache

### 3.3 Page Search

Page requests from access layer are served by first searching RAM-based buffer cache. If page is found, then the cached copy is read. If page was not present in buffer cache but present in NVC, then it is read from SSD into buffer cache. If the page was not present in NVC, then it is read from HDD into buffer cache.

Figure 2 shows this as `getpage()` which takes `LPAGENO` as input and returns `BUF`. `BUF` is in-memory structure which has page contents and its location on disk. A similar structure `NVCBUF` is used for NVC which contains only location on SSD. `buf_search()` searches `BUF` for the given `LPAGENO` in buffer cache. Similarly, `nvc_search()` searches for `NVCBUF` in NVC. `grab()` selects a page to be replaced with contents of the page being read. When page is read from SSD, it is served to access layer in buffer cache's memory.

Ideal scenario is that all warm page requests are served from SSD; hence, NVC design should make it more likely for future page requests to be found in NVC. Subsequent sections will show how this is achieved during page writes and replacement from buffer cache.

### 3.4 Page Writes

When NVC is in use, then user operations write pages only to SSD (Fig. 3). On each write, NVC is searched (`nvc_search()`) to determine if the page is already present. If page is found, then `NVCBUF` contains the physical location on SSD where new data is to be written. If page was not present in SSD, then either a free page is used to write the data or a clean page is reassigned for the page being written (via `nvc_grab()`).

Pages with latest data on HDD are considered clean pages. They can be replaced in SSD without any data loss. Pages being written to SSD are marked as dirty (`NVCBUF.IS_DIRTY`) so that they will be written to HDD in future. A least recently used (LRU) mechanism is used to the oldest clean page for replacement. This ensures pages which

```

1 pagewrite(BUF)
2         NVCBUF = nvc_search(BUF.LPAGENO)
3         if NVCBUF == NULL
4             NVCBUF = nvc_grab(LPAGENO)
5             write(NVCBUF.file, NVCBUF.offset, BUF)
6             NVCBUF.IS_DIRTY = true
7             NVC_dirty_count = NVC_dirty_count + 1

```

**Fig. 3** Page writes to NV cache

have not been accessed frequently are considered cold and they can be replaced. Such pages will be read from HDD in future if required.

The decision to write only to SSD is a very important one as it lays emphasis on making system's throughput depending on SSD write latency instead of HDD.

Since pages are never written directly from memory to HDD, NVC design imposes a strict ordering of a page modification across devices (Memory ! NVC ! HDD). This is beneficial for page search because access layer requires only latest modified copy of a page. Hence, page search can safely stop at whichever device it finds the page in.

### 3.5 Lazy Cleaner Task

A consequence of NVC write operation is that transactions will cache with dirty pages. These have to be cleaned up by writing them to HDD to make space for new pages being added in NVC.

Page cleaning requires two I/Os—a read from SSD and a write to HDD. A special background system task called NVC lazy cleaner has been introduced in the database server to bring down dirty pages in NVC to a particular configurable threshold called ‘dirty threshold.’ Whenever dirty page count crosses threshold, lazy cleaner moves pages from HDD to SSD in batches as shown in Fig. 4. After its work is done, the task puts itself to sleep. In order to maintain LRU property in NVC, nvc\_get\_lru\_buf() ensures oldest page le gets replaced.

```

1 while(NVC_dirty_count < NVC_dirty_threshold)
2     while i = 1 .. BATCHSIZE
3         NVCBUF[i] = nvc_get_lru_buf() read(NVCBUF[i].file,
4             NVCBUF[i].offset, BUF[i]) write(BUF[i].file,
5                 BUF[i].offset, BUF[i]) NVCBUF[i].IS_DIRTY = false
6
7     NVC_dirty_count = NVC_dirty_count - 1

```

**Fig. 4** Lazy cleaner

```

1  page_evict(BUF)
2      unhash(BUF)
3      if BUF.access_freq < NVC_selectivity
4          return
5      else
6          pagewrite(BUF)

```

**Fig. 5** Page writes to NV cache on buffer cache replacement

### 3.6 Page Eviction

Buffer cache replaces unchanged warm pages with hot pages (using grab() routine, Fig. 2). Pages being replaced could become hot again in future. Such pages are written to NVC in the hope that future request of the page can be served by NVC instead of HDD. If page is already present in NVC, then it is not written as its contents are unchanged.

Caching all evicted pages is counterproductive for pages which are never accessed again. Hence, pages with access frequency higher than a configurable threshold (NVC\_selectivity) are cached in NVC. This gives an advantage over external caching mechanisms provided by le systems or third-party vendors to avoid caching unnecessary pages. Since database servers manage their own storage, hence such access information helps in intelligent data management.

In absence of NVC, when a page is replaced and later brought back into memory, the whole sequence requires one I/O which is an HDD read. In presence of NVC, the same sequence may require maximum two I/Os that is one SSD write during eviction and one SSD read. Despite NVC adding an I/O, the SSD speeds make the whole sequence faster than HDD.3 (Fig. 5).

## 4 Performance Results

In this section, we will discuss performance experiments and results. We used same benchmark against three scenarios (HDD store, SSD store, NVC). Experiments were run on Intel(R) Xeon(R) CPU E7-8880 v3 @ 2.30 GHz processor with 8 sockets and 18 cores per socket. We used Fujitsu's FTS PRAID EP420i hard disks and Intel DC P3700 Series SSDs. The workload used 50 concurrent users, and SAP ASE was configured with 16 threads. Test database size was 1000 MB with 33% space free at the start of run.

Transactional throughput is measured in transactions per minute (tpms). ‘Thread utilization’ was sampled from the running server. It gives percentage of time spent by threads in performing I/Os vs time spent in CPU bound activities.

These experiments demonstrate that NVC throughput is very close to SSD store, and this could be achieved with significantly less SSD space.

**Table 1** tpm—transaction per minute

HDD store	SSD store	NVC-200 GB	NVC-100 GB
71,463	325,717	296,174	235,118

**Table 2** Resource allocation in GBs

Scenario	Memory HDD SSD		
	HDD store	1000	0
SSD store	20	0	1000
NVC-200 GB	20	1000	200
NVC-100 GB	20	1000	100

## 4.1 Benchmark

We used a performance with mixed workload consisting of selects, inserts, updates, deletes. Concurrent users perform transactions on a predefined set of tables. The workload and server were configured so that the active data set does not fit in memory. As a consequence, thread activity became I/O bound (HDD store: 83.5%, SSD store: 0.4%, NVC-200 GB: 0.6%, NVC-100 GB: 2.1%). Once this baseline was established on HDD store, the same configuration was used on SSD store and NVC scenarios (Table 1). NVC scenario was run twice: 200 GB (NVC-200 GB) and 100 GB (NVC-100 GB). The system was minimally tuned as the focus was to compare performance across device storage and not achieve highest tpm.

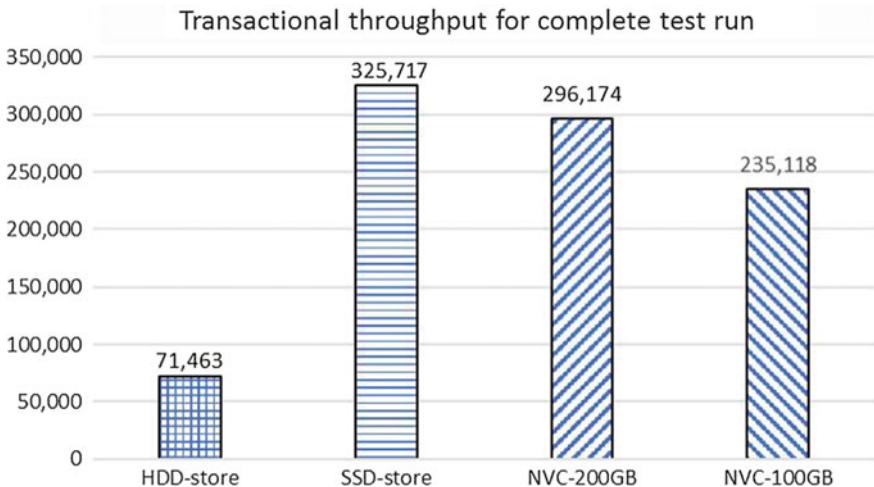
## 4.2 Results

Table 2 shows transactional throughput for all the runs for comparison. The same data is shown inside chart in Fig. 6. SSD store throughput was 4.5 times that of HDD store. This was expected as test setup makes throughput dependent on I/O latency.

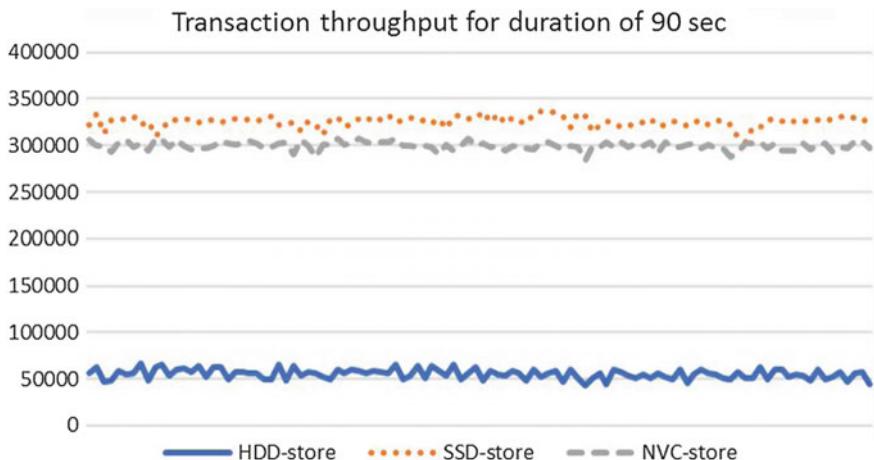
Even though the test setup makes the system performance dependent on the underlying storage, we observe differences between NVC-200 GB and SSD store runs. NVC-200 GB used 1/5th of SSD compared to SSD store but gave only 9% drop in throughput. Foremost reason for this benefit is that entire database is not accessed frequently, so its read and write could be restricted to RAM and SSD. This demonstrates that intelligent placement of data can reduce resource usage footprint.

The comparison between NVC-100 GB and NVC-200 GB shows one can trade off capacity with throughput. Compared to NVC-200 GB, NVC-100 GB used half of SSD capacity with only 20% throughput.

Figure 7 shows tpms measured periodically during a sampling period of 90 s, this is view of system when all caches are warmed up, and system has reached a stable state, and throughput is not fluctuating.



**Fig. 6** Transaction throughput for complete run



**Fig. 7** Transaction throughput for duration of 90 s

## 5 Related Work

Intelligent data placement to store hot data in faster device is not a new idea. Performance and price offered by SSDs make them suitable for tiered data management solution.

Oracle 11g release two introduced ‘Smart Flash Cache’ feature [2] which uses SSDs as secondary cache. It is aimed to benefit read-mostly or read-only workloads as the cache only stores clean pages. This is different from our approach as we store

dirty pages which are pushed to HDD by a dedicated task. ‘Buffer Pool Extension’ feature in SQL Server 2014 [3] also extends memory-based buffer cache using SSD. This implementation also stores clean pages in SSD.

Canim et al. [4] investigate using SSD as a write through secondary cache which stores clean pages. Page writes have to be written immediately to HDD. Caching policy named temperature-aware caching (TAC) based at granularity level of disk region is a set of physically contiguous pages. Pages are cached in SSD based on region temperature. In addition to written pages being cached in SSD, even pages read from HDD can be immediately written to SSDs based on their temperature.

Lazy cleaning technique has been explored by Do et al. [5] to have a write back SSD cache. A background task flushes pages from SSD to HDD when number of dirty pages in SSD crosses a threshold. Additionally, during checkpoint all pages are flushed from SSD to HDD. Our design does not necessitate this as persistent meta-data maintained inside SSD ensures dirty pages are recovered.

## 6 Enhancements and Future Work

A database has different types of pages; they have distinctive access pattern. For example, database log pages are typically written sequentially during runtime for purpose of crash recovery in future. These pages are written more than read during runtime. Current NVC design caches all types of pages without any distinction. A workload which generates a lot of log records can fill up NVC with log pages not all of which might be read in near future. An enhancement in NVC design could be to age out log pages much faster than data pages.

Traditional stable storage media has always been block based; thus, application is required to marshal in-memory data structures in blocks for persistence on stable storage. SSDs follow same block-level interface as HDDs. Non-volatile memory (NVM) is a new class of stable storage media which is accessed as a linear byte-oriented address space much like a DRAM. This opens up opportunities to have durable data structures, meaning they can be recovered even after system crash. The work done for SSD-based cache can be extended to use NVMs as a replacement of SSDs. Currently, the in-memory data structures (refer Sect. 3.2) used by NVC are packaged into meta-page. This creates a contention between tasks writing on same meta-page even though their changes are in different data pages. NVM can eliminate the need for meta-pages as in-memory data structures could be persisted without serialization into pages.

## 7 Conclusion

In this paper, we discussed how SSDs can extend memory-based buffer cache. We outlined key design decisions to retain active data set within memory and SSDs combined. As a consequence, throughput does not depend on HDD latencies despite

a large portion of the storage residing in HDD. This approach uses caching algorithms to retain hot data in memory, warm data in SSD, and cold data in HDD. Tiered storage architecture keeps hardware costs low by using minimal SSDs.

Caching algorithms offer ease of administration as they move data automatically between storage classes based on page-level statistics instead of an administrator having to move tables on different media. Non-volatile cache offers configurations which can be dynamically adjusted to accommodate change in access patterns. SSD as a cache approach also has an advantage that the data movement between media takes place at page-level granularity instead of having to move entire database or its objects between HDD and SSD.

Based on ideas described here, an application for patent has been led in USA with application number: 15/070,355 [6].

**Acknowledgements** We would like to acknowledge SAP ASE development and server performance engineering teams who have provided valuable feedback.

## References

1. What's New in SAP Adaptive Server Enterprise 16 SP02. <https://www.sap.com/documents/2016/06/02a21e18-767c-0010-82c7-eda71af511fa.html>.
2. Packer, A., Jernigan, K., & Ramanujam, S. Oracle database smart flash cache. <http://www.oracle.com/technetwork/articles/systems-hardware-architecture/oracle-db-smart-ash-cache-175588.pdf>.
3. Buffer Pool Extension. <https://docs.microsoft.com/en-us/sql/database-engine/configure-windows/buffer-pool-extension>.
4. Canim, M., Mihaila, G. A., Bhattacharjee, B., Ross, K. A., & Lang, C. A. (2010). SSD bufferpool extensions for database systems. In *VLDB*.
5. Do, J., Zhang, D., Patel, J. M., DeWitt, D. J., Naughton, J. F., & Halverson, A. (2011). Turbo charging dbms buffer pool using SSDs. In *Proceedings of the SIGMOD International Conference on Management of Data*.
6. Prateek, A., & Vaibhav, N. Database caching in a database system. Application number 15/070355. <http://appft.uspto.gov>.

# Cloud-Based Healthcare Monitoring System Using Storm and Kafka



N. Sudhakar Yadav, B. Eswara Reddy and K. G. Srinivasa

## 1 Introduction

Medical care is one of the key requirements in today's fast development of the industrial life. The traditional method and process of medicine will saturate and cannot satisfy the requirement. In this regard, telemedicine a new concept has emerged. In the telemedicine field, we can cover patients at any time and place. Due to the rapid development of mobile computing, the population of people using mobiles is increasing [1]. With these mobile devices, the border between the telemedicine and traditional medicine is slowly disappearing. Cloud computing with its unique and key features such as dynamic, scalable and reliability has provided healthcare as a service via Internet. Due to the advances and capabilities of processing and storage, cloud computing has led the way for telemedicine concept.

In the telemedicine field, it is important to provide efficient automated analysis observing the healthcare records. In this regard, it is required to achieve reliable communication and high throughput. Cloud computing offers horizontal scalability geographically as well in a distributed manner [2]. Big data growth is one of the driving factors in healthcare industry. It is difficult to manage the data as the amount of data digital information increases. The main drawback of the data related to healthcare industry is some of the data are not digital transcript. This accounts for

---

N. Sudhakar Yadav (✉)

CSE Department, Jawaharlal Nehru Technological University, Ananthapur, Andhra Pradesh, India  
e-mail: sudhakaryadav.mtech@gmail.com

B. Eswara Reddy

CSE Department, JNTUA College of Engineering, Kalikiri, Chittoor, Andhra Pradesh, India  
e-mail: eswarcsejntua@gmail.com

K. G. Srinivasa

Department of Information Technology, Ch. Brahm Prakash Government Engineering College,  
New Delhi, India  
e-mail: srinivasa.kg@gmail.com

future clinical advances and sometimes inaccessible to the researchers. Cloud computing can enable data sharing and integration at large scale. In [3], different solutions for healthcare and Big data are provided stressing the point that virtualization has provided healthcare to move to cloud-based solutions.

In recent years, cloud computing is used for a variety of application areas such as online retail, marketing, and health care. It provides virtualization through virtual machines (VMs) and is accessible via the network. The services offered by the cloud are dynamic, scalable, and accessible via Internet. Pay-as-you-go model enables the cloud computing to provide computing services that are scalable and on-demand resources. Thus, cloud computing is a key technology that can be used in the areas of healthcare and monitoring systems. In the field of healthcare, continuous monitoring and attention of patients are to be cost-effective along with highly quality of service. It has been suggested that cloud computing is one of the key technologies that can be used for healthcare systems [2, 3].

Using cloud computing technologies in healthcare field reduces the cost of maintenance, license fees and reduces response time [2]. In addition to these several innovations, initiatives and number of applications are already under process to adopt cloud technologies. With help of advanced cloud infrastructure technology, the opportunities in the field of health care technology have been improved a lot [3]. In this paper, the proposed system introduces a new framework for cloud-based service for healthcare. It uses Hadoop as the underlying infrastructure with Apache Storm and Apache Kafka as the integrating components. The main aim of the proposed system is to provide a cloud-enabled service for health care with low response time and high throughput.

## 2 Related Work

In the field of healthcare systems, it is challenging to keep up the technical considerations such as reliability and scalability for data management. There exists a huge challenge to provide a quick and easy general data center that offers a reliable service with huge infrastructure. However, cloud computing is a promising solution to overcome these challenges and offers a good healthcare monitoring system. In this section, brief descriptions of the existing telemedicine systems are discussed. One of the key features for the adoption of cloud-based healthcare systems is it is cost-effective. In [4], a cloud-based telemedicine service was enabled and proved to be cost-effective. In [5], a genomic analysis was carried out using a cloud-based service and proved to be a viable and cheaper technology. A comparison of computational and economic characteristics of the service offered through cloud was carried out and proved to be cost-effective.

Grid computing was used as a technology enabler to monitor health issues related to cardiovascular diseases [6]. In this system, personal digital assistant (PDA) was used as the portals for monitoring the health. This provides the application as software as a service (SaaS) basis. In the proposed system, this work is enhanced with the

integration of Apache Kafka and Apache Storm by reducing the response time for the monitoring. A sensor-based platform solution proposed in [7] implants the sensors into the bodies of the patients to get vital data and assist in monitoring. Since it is sensor based, nodes are not scalable and “on demand” to facilitate the monitoring and analysis. In the proposed system, patient health records are simulated rather than the use of sensors in the patient bodies for analysis.

Cloud computing-based framework for healthcare monitoring is proposed in [8, 9] with the use of Microsoft technologies and existing toolkits for cancer patients. An HTML-based Web solution is also provided since it is empowered with cloud. This shows that cloud computing remains to be the paradigm for healthcare monitoring and services. The current position of cloud computing in healthcare field is explored in [10, 11]. The concept of e-health cloud monitoring introduced in [9] proposes a system for building an environment that facilitates automated health monitoring. In [11], integration of the formal healthcare systems with cloud-based services was experimented and achieved success in facilitating automated monitoring. In [13], the different mechanisms for handling electronic health records are analyzed with the support of both local and centralized access. Thus, the proposed system integrates the concept of telemedicine and cloud services to use a platform for supporting the data integration and data analysis.

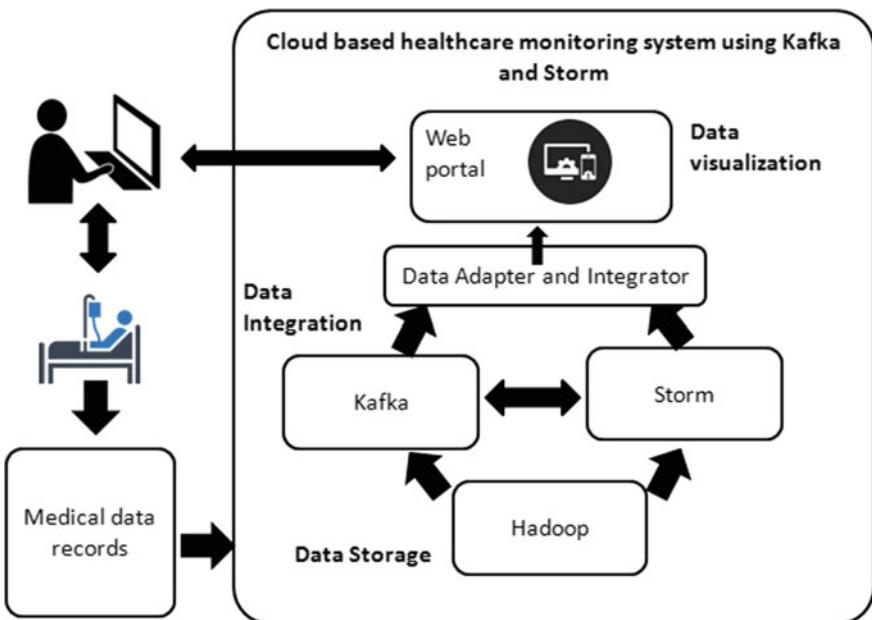
Electronic Health Record (EHR) for patients maintained in the cloud proposed in [5] is a novel are in healthcare field. This environment gives access to the doctors to view the patient’s electronic health records at any time and from anywhere with the permission from patients. So that patients can take the doctor’s advice for further health improvement. Using advanced mobile devices technologies can manage the medical data in cloud and access them using our own mobile proposed in [6]. Collecting the patient’s data with help of sensors and storing them in the cloud environment for analysis discussed in [7]. The healthcare monitoring of patient for ECG data has been proposed in [8]. In the same way, EHR system with the help of smart card has been introduced in [9]. The only drawback of these systems is the use of basic cloud infrastructure rather than the advanced cloud-based solutions for automated healthcare monitoring. In this paper, the proposed system uses Apache Kafka and Apache Storm for automated telemedicine and monitoring.

### 3 Proposed System

Cloud computing can be used as one of the solutions for analysis of healthcare signals or records in near real time. It can be accomplished with software as a service (SaaS) model of cloud computing. With SaaS as a platform, historical data of patients can be diagnosed and analyzed through various services available in the cloud. Cloud infrastructure utility minimizes the time for analysis and computation in a distributed computing environment. SaaS takes the data in all kinds of forms (graphical and numerical) and forwards the analyzed data to the patient’s mobile. In the case of

emergency situations, SaaS can be configured to alert ambulances and doctors in the hospital for decision making based on the results of analysis [7].

Figure 1 illustrates the working of the proposed framework of healthcare monitoring system using cloud with Apache Kafka and Apache Storm. In the proposed system design, wireless health monitoring devices are used for collecting the user data and later communicating telemedicine center (TMC) via global network. It contains components such as data collection/request unit, TMC unit for analysis and monitoring. A layered approach is followed for healthcare analysis and monitoring. Our approach represents that the top-down order contains the cloud service models like software as a service (SaaS), platform as a service (PaaS), and the infrastructure as a service (IaaS), respectively. The main aim of implementing the proposed system as SaaS is to facilitate the client side functionality and hiding the complexities and underlying functions of the cloud application. The IaaS layer involves Hadoop as the core file system for handling storage of medical records. In PaaS layer, Apache Kafka and Apache Storm are used for data integration and analysis in order to visualize in the Apache.



**Fig. 1** Framework for cloud-based healthcare monitoring

### ***3.1 Web Portal***

A Web portal is the opportunity for the user to directly interact with the cloud system. In the Web portal, we can deploy the software for analyzing the patient's health parameters data like heartbeat, temperature, glucose level, ECG. We use the service in the cloud called SaaS. The electronic health record (EHR) of the patient would be analyzed with the help of SaaS cloud service. The EHR contains the historical and present data. The access is provided to the data in the cloud for authenticated users. Patients can also access the data in cloud specifying the real-time analysis and diagnosis and can in turn consult the doctor through teleconferencing service.

### ***3.2 Data Adapter and Integrator***

The data adapter and integrator modules are acts as a mediator between data visualization and data storage modules. The aim of these is to perform the job of SaaS in the architecture. This layer gets the different formats of data by different patient's health devices and then converts it into a desired format required to the architecture. After that, combine the latest data to the electronic health record (EHR) for analyzation purposes. This layer continuously monitors the HER's data, and if in case any abnormal results are presented, alert the patient or caretaker.

### ***3.3 Apache Kafka***

This is an open-source stream processing manifesto refined by the Apache Software Foundation which is written in Scala and Java. Its storage layer is intrinsically a “ponderously adaptable pub/sub message queue architected as a circulated transaction log,” making it eminently treasured for enterprise frameworks to outgrow the streaming data.

Kafka mainly has four application programming interfaces (APIs) which are presented as follows. These APIs are used to interconnect with Apache Storm for healthcare monitoring.

1. Producer APIs: Grants an application to promulgate a stream of records to one or more Kafka topics.
2. Consumer APIs: Grants an application to endorse to one or more topics and outgrows the stream of records composed to them.
3. Stream APIs: Grants an application to execute as a stream processor, engrossing an input stream of records from one or more fields which in return produces an output for one or more output fields.

4. Connector APIs: Grants erected and functioning metamorphic producers or consumers that affix Kafka fields to current applications or data systems. For instance, a damp to a relational database might ensnare every switch to a table.

### 3.4 Storm

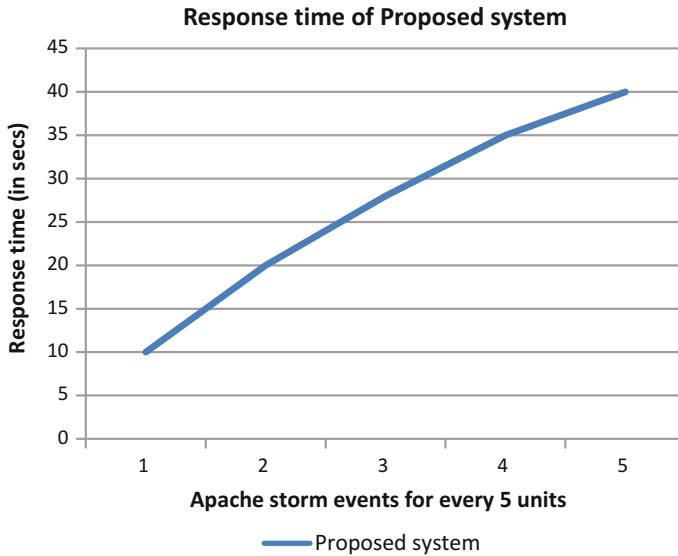
Apache Storm is a real-time infrastructure to smoothly deliberate real-time analytical need. The storm application is delineated in the configuration of a directed acyclic graph (DAG) along with spouts and bolts that ventures as vertices. Edges of the graph are appellate as streams unmediated data from one mode to another mode. In amalgamation, the topology performs as a data transformation pipeline. At the peripheral degree, in general topology structure is homogeneous to Map-Reduce activity where, the predominant difference is that the data is processed in real time as antithetical to in single batches. The storm is a distributed, foible resilience real-time computation; it runs in a uniformed manner until wrecked, whereas MapReduce task DAG must ultimately culminate.

Kafka and Storm spontaneously augment each other and their persuasive collaboration facilitates real-time streaming analytics for fast-stirring big data. The Kafka and Storm integration is in practice to make it more convenient for the developers to devour and circulate the input the streams from storm topologies and perform analytics in memory for no loss of data and important accessing of obtained available patient data. In the next section, the proposed framework is tested out and the response time is recorded.

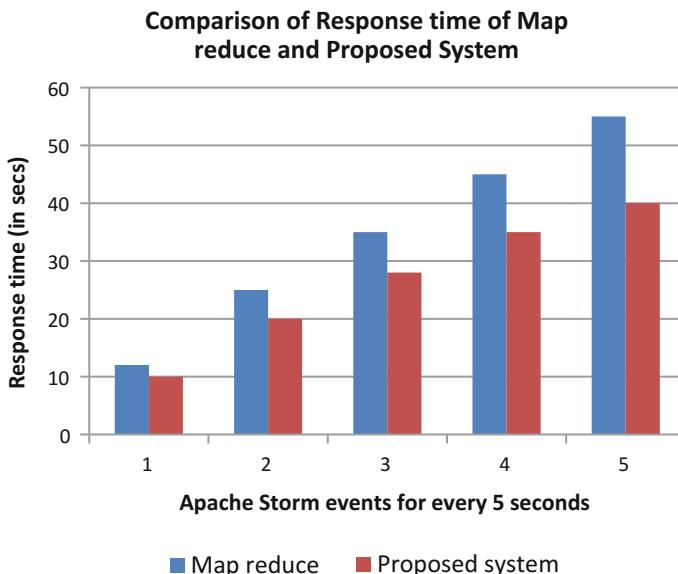
## 4 Experiment and Results

The experimental arrangement of multi-node cluster of Kafka, Storm, and Zookeeper runs in such a way that the Kafka spout is the source of consuming data into it and produces it into storm bolts for processing, and then the bolts process the tuples and generate new bolts with processed tuples. Kafka spout is the blue print and the primary source. The source sends the data to the parser and splits the process to many like; storm events, location sink, storm events, storm-events-sink, stats, stats-sink, etc. The novelty of this paper is the integration of cloud service with Kafka and Storm services to collect the real-time data and reduce the response time. Apache Storm was used to collect the real-time patient data and was integrated with Apache Kafka to create a workflow for monitoring system. The patient data was simulated with randomness and varied with increase of five units into the Apache Storm. The response time recorded with variation in the data is as shown in Fig. 2.

A MapReduce program was run with the existing simulation dataset that was fed into the Apache Storm. The comparison of results of the MapReduce-based program and the integration into Apache Storm and Kafka are as shown in Fig. 3. We can



**Fig. 2** Response time of the healthcare monitoring system using cloud



**Fig. 3** Comparison of MapReduce and Apache Storm

observe from Fig. 3 that the use of Apache Storm for monitoring events of patients such as ECG and heartbeat is more responsive than the MapReduce program.

## 5 Conclusion

The proposed system reduces the response time so that it helps tracking the patients actively. In short, our framework delivers an automated telemedicine system that contains the starting from the collection of data to producing the relevant information to the stakeholder. This telemedicine service contains a number of advantages like collecting on time data so that it eliminates the manual process of collection of data. Because of technological advancements in the field of computer science, the process of deployment and providing services to healthcare stakeholders is made easy through PaaS and SaaS cloud models. As in future work, we are planning to use this in real world with enhancements of some security services to get the benefits to the healthcare stakeholders.

## References

1. Zhijie, W., & Qing, G. (2014). The situation of abroad medical tourism research and its teaching. *Chinese Journal of Health Policy*, 11, 59–63.
2. Dudley, J. T., Pouliot, Y., Chen, J. R., Morgan, A. A., & Butte, A. J. (2010). Translational bioinformatics in the cloud: An affordable alternative. *Genome Medicine*, 2(51).
3. Hwang, K., Dongarra, J., & Fox, G. C. (2013). *Distributed and cloud computing: From parallel processing to the Internet of Things*. Morgan Kaufmann.
4. Cimler, R., Matyska, J., & Sobeslav, V. (2014). Cloud based solution for mobile healthcare application. In *Proceedings of the 18th International Database Engineering and Applied Symposium* (pp. 298–301).
5. Greene, C. S., Tan, J., Ung, M., Moore, J. H., & Cheng, C. (2014). Big data bioinformatics. *Journal of Cell Physiology*, 229(12), 1896–1900.
6. Ma, Y. J., Zhang, Y., Dung, O. M., Li, R., & Zhang, D.-Q. (2015). Health Internet of Things: Recent applications and outlook. *Journal of Internet Technology*, 16(2), 351–362.
7. Chen, M., Ma, Y., Li, Y., Wu, D., Zhang, Y., & Youn, C. H. (2017). Wearable 2.0: Enabling human-cloud integration in next generation healthcare systems. *IEEE Communications Magazine*, 55(1), 54–61.
8. Abawajy, J. H., & Hassan, M. M. (2017). Federated Internet of Things and cloud computing pervasive patient health monitoring system. *IEEE Communications Magazine*, 55(1), 48–53.
9. Chu, X., Nadiminti, K., Jin, C., Venugopal, S., & Buyya, R. (2007). Aneka: Next generation enterprise grid platform for e-Science and e-Business applications. In *Proceedings of the 3rd IEEE International Conference on e-Science and Grid Computing* (pp. 10–13). Los Alamitos, CA, USA: IEEE CS Press.
10. Pandey, S., Voorsluys, W., Niu, S., doker, A., & Buyya, R. (2012). An autonomic cloud environment for hosting ECG data analysis services. *Future Generation Computer Systems*, 28, 147–154.
11. Costs, D. U. (2007). The growing crisis of chronic disease in the United States. *Partnership to Fight Chronic Disease*.
12. Shammugasundaram, G., Thiagarajan, P., & Janaki, A. (2017). *A survey of cloud based healthcare monitoring system for hospital management*. Springer Nature.

# Honeynet Data Analysis and Distributed SSH Brute-Force Attacks



**Gokul Kannan Sadasivam, Chittaranjan Hota  
and Bhojan Anand**

## 1 Introduction

Network attacks are increasing in both frequency and intensity in recent years. Hackers tend to build automated malware tools that search for vulnerable systems on the Internet. Major attacks like DDoS attacks, ransomware attacks are made possible by infecting a vulnerable system. Some of the attacks are targeted on a particular organisation causing substantial financial loss. In general, the attacks are performed to bring down the servers, to corrupt database records and to steal confidential information.

In a password cracking attack, the malicious entity tries to figure out the right username and password for a system. In most of the business organisations, usernames and passwords are used to login to a system. The primary method of authentication in bank Web sites, emails, social network sites is password-based. Hence, cracking a username–password has become a lucrative business for hackers. In 2015, one-quarter of the attacks are brute-force password-guessing attacks [1, 2].

In password-guessing attacks, a single machine can try several usernames and passwords until the server resets the connection. However, attack with several attempts in a relatively small duration of time can be easily detected. There are several off-the-shelf tools (e.g., iptables, denyhosts, blockhosts, fail2ban) to detect them. In a stealthy password-guessing attack, an attacker tries to evade detection by trying only a couple of username–passwords in a given time. After a predetermined

---

G. K. Sadasivam (✉) · C. Hota

Department of Computer Science, BITS, Pilani, Hyderabad Campus, Hyderabad, Telangana, India  
e-mail: gokul@hyderabad.bits-pilani.ac.in

C. Hota

e-mail: hota@hyderabad.bits-pilani.ac.in

B. Anand

School of Computing, National University of Singapore, Computing 1, 13 Computing Drive,  
Singapore, Singapore  
e-mail: dcsab@nus.edu.sg

period, the attack machine tries another pair of a username and password. A stealthy attack is potent than an ordinary password-guessing attack. However, stealthy attacks take a very long time to succeed. In a distributed stealthy attack, a hacker builds a botnet that commands the individual bots to guess different username–password combination on a particular server. Depending on the size (number of bots) of the botnet, the time to find the correct username–password is relatively faster than a single-source stealthy attack.

We have built a honeynet architecture that is not affected by network attacks. Data analysis is done to investigate the types of attacks captured by this system. The investigation proves that our honeynet system captures distributed brute-force attacks. This work is an extended work [3] of the authors, and their earlier publication was in ‘Emerging Information Technology and Engineering Solutions (EITES 2015)’ conference held in Pune, India.

The rest of the paper is as follows. Section 2 presents the related works. Section 3 briefly explains the experimental setup used for malicious data collection. Malicious traffic data analysis is done in Sect. 4. Section 5 analyses the SSH attacks. Section 6 summarises our work and mentions the future scope.

## 2 Related Work

There are several resources (one of them is [4]) that give an overview of honeypot systems. The book by Spitzner [4] describes the history and the value of a honeypot, the honeypot tools (BackOfficer Friendly, Specter, Honeyd), honeynets and so on.

Abdou et al. [5] analysed the attacker’s source IP address and the IP allocated (by IANA) to different countries. Statistical analysis of the password lengths and the password characters (alphabets, numbers, and symbols) provided significant details about attacker’s strategies. Their work provided a heatmap that displays the percentage of passwords shared between any two subnets. Also, timing analysis was done to plot the average number of attempts per day for different scenarios. Owens and Matthews [6] reported the common usernames and passwords tried on an SSH server and noted the sharing of dictionaries among multiple source machines. Their network capture observed a slow-motion brute-force attack and a distributed brute-force attack.

Rabadia and Valli [7] worked on finding the word list used by brute-force attack tools to guess the correct passwords. Their findings suggest that most of the attackers use Kali Linux distribution. The work by Sokol and Kopčová [8] discusses the correlation of the World Bank data with the number of login attacks. There are 130 different fields in the World Bank data, and 13 of them have a strong/moderate correlation. Pearson’s coefficient R is used to find the correlation as strong, moderate, and weak.

Javed and Paxson [9] have proposed a method to detect distributed brute-force attacks on SSH server and have formulated a method to detect the participants in an attack. The attack participants are grouped based on a set of local machines or a

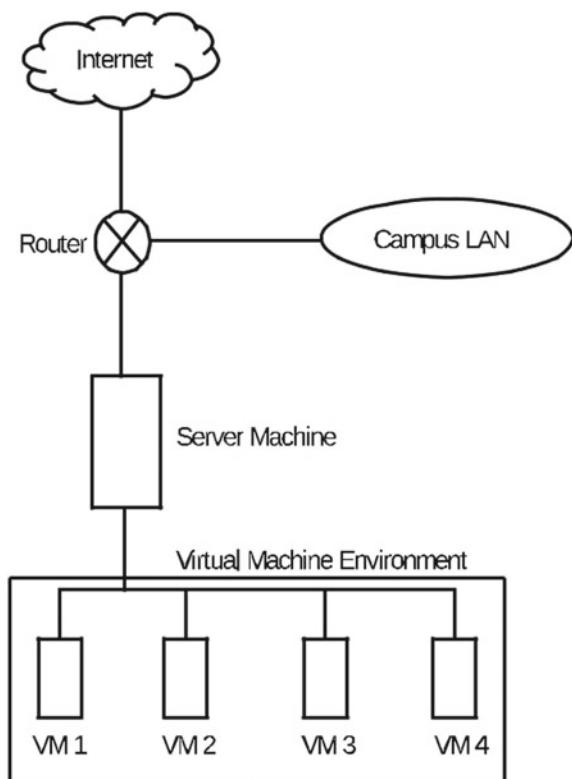
common username (e.g., ‘root’). Malecot et al. [10] have proposed a visual technique to detect coordinated brute-force attacks on SSH server. Both the works consider logs from several server machines to detect distributed brute-force attacks.

Saito et al. [11] mention about a new type of distributed brute-force attack on RDP service. Here several source IPs work together in attacking a single machine. The attacking machine is disciplined such that it keeps attempting login over a time interval. The login trials of all source machines have the similar total number of logins, average number of logins, and standard deviation of logins. They mention that these cannot be detected by conventional security tools and propose a model to detect and mitigate it.

### 3 Honeynet Architecture

Figure 1 shows the complete experimental setup built in BITS, Pilani—Hyderabad campus. All traffic from the ISP enters the university campus through a Cyberoam Firewall. Cyberoam redirects the traffic to the honeynet via a Cisco Core Switch.

**Fig. 1** Honeynet architecture



**Table 1** Server machine configuration

System information	
Processor	Intel Xeon processor
Speed	2.67 GHz
RAM	4 GB
Disc space	600 GB
OS	Ubuntu 16.04

A honeynet system was deployed on the server machine (DELL PowerEdge Blade Server) which had the machine configuration shown in Table 1. Virtual machine manager (VMM) was installed on the server. On top of the hypervisor, four virtual machines were installed. All the virtual machines were running mini-Ubuntu 14.04 LTS operating system. Each one of the virtual machines executed a different honeypot. The VMM operated in NAT mode.

The hostname of the server machine was ‘BITS-OS-PC’. The machine had a NIC card configured with a public IP address. Since there was a need to create multiple virtual hosts, a NAT tool (iptables) was installed to map a public IP address to several private IP addresses.

The honeynet system ran for two different time periods. First, it ran for a total of 25 consecutive days from 22 July 2014 to 16 August 2014. Second, it ran from 11 April 2017 to 29 April 2017 (total 19 days). During these time periods, the honeynet ran day and night incessantly. Here in after, ‘Dataset 1’ refers to network traffic captured during 2014 and ‘Dataset 2’ refers to traffic captured during 2017. The amount of network traffic collected in pcap format is approximately 86 MB (Dataset 1) and 893 MB (Dataset 2). There were nearly 669,831 packets in Dataset 1 and 3,567,125 packets in Dataset 2.

## 4 General Characteristics

In a TCP flow pcap file, the packets are between two systems over a single conversation (one client port number and one server port number). Different types of TCP flow pcap files were observed (as shown in Table 2).

**Table 2** TCP flow types

TCP flow types	Dataset 1 (total flows)	Dataset 1 (%)	Dataset 2 (total flows)	Dataset 2 (%)
No SYN packets	4	0.01	70	0.07
One SYN packet	2287	8.50	26,154	24.83
Two SYN packets	23,228	86.31	49,623	47.12
n-SYN packets	1392	5.18	29,475	27.98

‘No SYN packets’ refer to all TCP flow pcap files having no SYN packets (the SYN flag is zero in the TCP header). There were two types of flows in this category: port scanning flow that started with an acknowledgement packet (the ACK flag is one in the TCP header) and finished with an RST packet, NAT error that captured the latter part of a TCP flow. ‘One SYN packet’ is TCP flows that have only one SYN packet in it. It refers to the flows like SYN packet followed by an RST packet. ‘Two SYN packets’ are the normal flows which start with an SYN packet and then go to SYN-ACK packet and so on. ‘n-SYN packets’ refer to TCP flows that have more than two SYN packets in it.

There is the possibility of a flow having more than two packets with SYN flag set. Firstly, if the attacker machine does not reply to SYN-ACK packet, then the server keeps sending SYN-ACK for a given interval. On Linux machines, the time between successive SYN-ACK packets is exponential. Secondly, an attacker machine may use the same source port number again. The above conditions could lead to many SYN packets (new TCP connections) in a single TCP flow.

In the following analysis, the flow types considered are ‘No SYN packets’, ‘One SYN packet’ and ‘Two SYN packets’.

## **4.1 Source of the Attacks**

It is essential to know the source of the network attacks. The source refers to the country location, the characteristics of the attack tool, and so on. The location can be obtained using the source IP address. The source IP address might point to the location of an attacker or that of the victims or the proxy server which an attacker uses to hide his location.

In Dataset 1, approximately 3050 unique IP addresses belong to 77 different countries, and in Dataset 2, nearly 26,070 unique IP addresses belong to 153 different countries. Most of the IP addresses belonged to China accounting for 61.25% of the traffic (TCP flows) in Dataset 1 and 49.49% in Dataset 2. Besides China, 18.97% of the attackers in Dataset 1 belonged to the USA (11.76% in Dataset 2). In general, more than 65% of the attackers were from Asia (both datasets). The Website [maxmind.com](http://maxmind.com) [12] provided the geolocation for all the IP addresses.

Some of the attackers tend to be busy on the honeynet system (i.e. the maximum number of TCP flows). These attacker’s countries are China, Germany, Turkey (top-3 attacker’s countries) for Dataset 1 and China, France for Dataset 2.

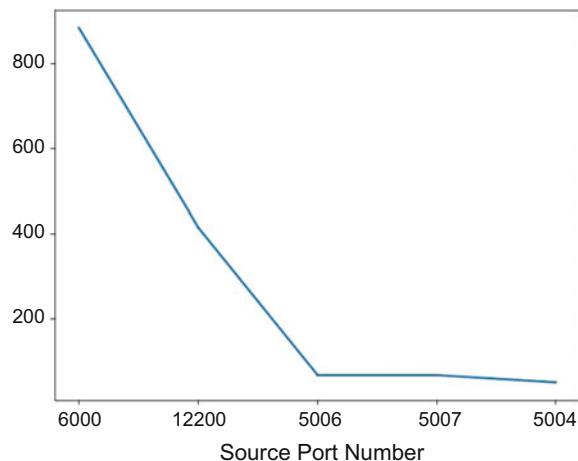
Abdou et al. [5] analysed the number of login attempts (in SSH TCP flows) on different country’s IP blocks. Their work assumed classful addressing (sub-blocks/8, /16, /24).

However, the IP blocks use classless addressing. The authors have considered all the TCP flows (irrespective of the service) and calculated the distinct subnets. In Table 3, the list of top subnets (classless addressing) is provided.

All the subnets in Dataset 1 and Dataset 2 of Table 3 belong to China. It is notable that Abdou et al. [5] got China as the first country.

**Table 3** Busy IP blocks

Dataset no.	IP blocks	Total flows	Percentage
1	116.8.0.0/14	1833	6.81
1	61.160.0.0/11	1732	6.44
1	60.160.0.0/11	1282	4.76
2	116.16.0.0/12	14,441	13.71
2	58.192.0.0/11	12,444	11.82
2	59.32.0.0/11	9675	9.19

**Fig. 2** Source port distribution for Dataset 1

The source port number (in TCP header) is chosen by the operating system and is an ephemeral number. The range of this number depends on the operating system (OS) used. As per IANA standard, a dynamic port number should be in the range 49,152 through 65,535 [13]. Many operating systems conform to the Internet Assigned Numbers Authority (IANA) standard for assigning source port number. For some Linux-based systems like Ubuntu 16.04, the range is from 32,768 to 60,999. For Windows 10, it is from 16,384 to 49,152.

On observing the source port number of the network traffic, the count of each port number was not uniformly distributed (as shown in Fig. 2). Hence, most of the attacks are originating from a couple of source port numbers. The source port 6000 occurred most of the time in Dataset 1 (884 TCP flows, as shown in Fig. 2) and in Dataset 2 (167 TCP flows). The source port 6000 mostly attacked port numbers 1433 (345 TCP flows), 3306 (166 TCP flows) and 22 (44 TCP flows to SSH).

It leads to the conclusion that there are hacking tools which have a hardcoded source port number.

**Table 4** Most used SSH client library

Library version	Total count	Percentage
SSH-2.0-PUTTY	29,302	47.15
SSH-2.0-Ganymed Build 210	4476	7.2
SSH-2.0-libssh-0.1	3745	6.03
SSH-2.0-libssh2 1.4.2	3506	5.64

## 5 Secure Shell (SSH) Traffic Analysis

Kippo honeypot emulated the SSH service. The log files of the honeypot were dissected to gather meaningful information. Since the application payload of a packet was encrypted using a standard cryptographic algorithm, the TCP flow pcap files cannot be used as such.

Kippo log files were processed to gain useful information about the attackers and their attacking methods. Kippo report [14] written by Kamil Koltys provided the format of messages in the kippo log files. Using the log files, Java programmes were written to extract SSH logins (usernames and passwords), SSH version, attacker's IP address, and so on.

There were approximately 10,500 login attempts in Dataset 1 and 107,782 login attempts in Dataset 2. In a single TCP flow, an attacker can try 21 failed login attempts before termination of the connection. There were 1,131 unique usernames and 8,141 unique passwords in Dataset 1, whereas Dataset 2 had 723 unique usernames and 20,111 unique passwords. Most often tried username is 'root'. Several works [5, 6, 15] have described the most attempted usernames and passwords on an SSH server.

There were several login attempts to the kippo server. Many of them were failed attempts. Some of them were successful connections.

On minute observation, two categories emerged out of the successful login connections. The first category is, after a successful login, the attacker immediately terminated the connection. In the second category, after a successful login, the attacker tried several shell commands on the server. Hofstede et al. [16] made the same observation while studying different attack tools.

Most of the successful login connections did not execute any shell commands. The purpose was only to break into the server. Once the system logs in an attacker, the username–password would be shared with other attackers. These attackers would probably do the execution of commands to infect the system.

The SSH library name is exchanged during initial SSH protocol handshake. The SSH client library gives information about the attack tool. There were too many different library types and versions found during the investigation. Table 4 depicts the most often used client SSH library for combined dataset (both Dataset 1 and Dataset 2).

By default, the maximum number of login attempts in a TCP flow is six for an OpenSSH server, after which the TCP connection is reset. Moreover, the fail2ban

**Table 5** Number of login attempts (less than 6) in a TCP flow

No. of attempts in a TCP flow	No. of TCP flows (D1)	Percentage (D1)	No. of TCP flows (D2)	Percentage (D2)
1	5677	53.8	10,886	23.69
2	448	4.25	2328	5.07
3	127	1.2	21,806	47.46
4	62	0.59	108	0.24
5	62	0.59	419	0.91
6	41	0.39	32	0.07

tool has the default maximum retries (i.e. the number of failed login attempts) as three occurring over a duration of 10 min.

Table 5 shows the count and the percentage of TCP flows that had different login attempts from one to six. In Dataset 1, 6417 TCP flows (60%) had less than or equal to six login attempts per flow. In Dataset 2, 35,579 flows (77%) had less than seven login attempts per flow. Hence, most of the malicious traffic is designed to have less number of login attempts per flow.

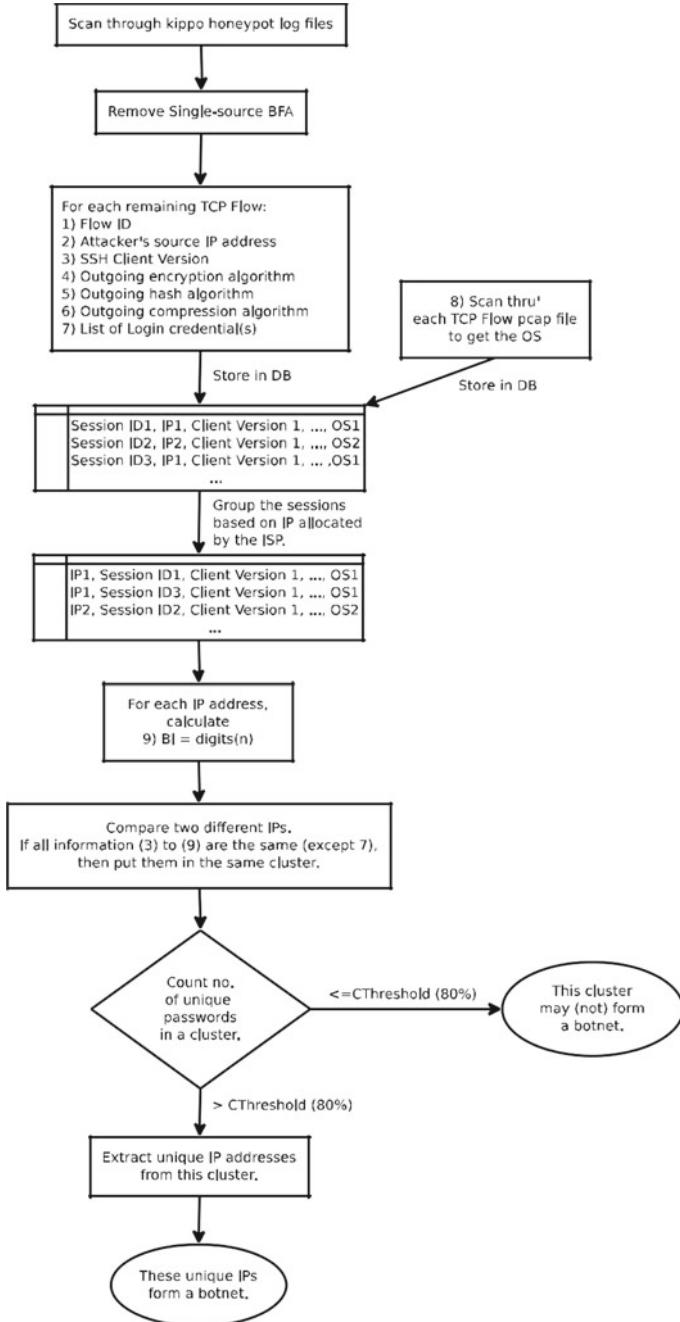
Hansteen has studied the distributed brute-force attack on SSH servers and coined the botnet as ‘hail-mary cloud’ [17]. Each bot tries a unique combination of user-name–password on a victim server. The number of login attempts for a given user-name was quite low, and the username changed after a number of retries.

The works [5, 6] discuss the coordination activity in a distributed brute-force attack. As per their works, the coordination is from a set of computer machines in a single subnet. The authors of this work claim that the attack sources can be from different geo-locations.

The works [9, 10] use information from more than one SSH server to determine the remote attackers. The authors of this work are doing the detection from a single server machine.

In a distributed brute-force attack, the attack source is an operating system executing a bot programme. All bot programmes in a botnet are written in the same language. SSH client version depends on the library file used to build the SSH client application. There are several libraries, and some of them are mentioned in Table 4. In an SSH botnet (a botnet that attacks SSH server), all bot programmes use the same SSH client library. Hence, the version information can be used to detect a set of coordinating attack hosts (i.e. an individual botnet).

In Fig. 3, a flow chart is shown that segregates individual botnets. In this approach, initially, all the single-source attacks are segregated. In a single-source attack, an attacker machine (with an IP address) keeps making several TCP connections to the server. In a TCP connection (or a flow), there could be one or more login attempts. If there are no login attempts for 10 min continuously, then the other TCP connections before and after this duration are considered as two different sessions. A session can have multiple TCP flows from a single attacker. All sessions that meet three login



**Fig. 3** Flow chart for segregating DBFA botnets

**Table 6** DBFA detection

Common string	No. of distinct IPs (D1)	Total no. of attempts	Percentage uniqueness
Windows XP, SSH-2.0-libssh-0.4.6, aes256-ctr, hmac-sha1, none, 1	1	2	1.00
Linux, SSH-2.0- libssh2-1.4.1, aes128-ctr, hmac-sha1, none, 1	1	2	1.00
Linux, SSH-2.0- libssh2-1.4.1, aes128-ctr, hmac-sha1, none, 3	2	1024	0.83

attempts in an average duration of 10 min (default rule of fail2ban) are considered brute-forcers.

The SSH client version and cryptographic algorithms (outgoing encryption algorithm, outgoing hash algorithm, outgoing compression algorithm) and the OS are known from the kippo log files and pcap files, respectively. The OS is determined using the p0f tool (passive fingerprinting tool).

Behavioural index (BI) tells the nature of login attempts from an attacker. It is the number of digits in an integer. For example, the BI for 2 is 1 and the BI for 850 is 3. For each set of sessions belonging to a particular IP address, the behavioural index is calculated. In Fig. 3, n is the total number of login attempts from an IP address and digits (n) returns the number of digits in n.

With the above parameters, the sessions are segregated into groups. Each of the unique login credential (username and password) is inspected in a group. If each of the login credentials is different, then with high probability, it can be inferred that this set of IPs form a botnet.

In Dataset 1, there were three different groups (or botnets) as shown in Table 6. All different variants of Linux are combined into the group ‘Linux’. The first column is the common string used to group a set of sessions. In the last row, two IP addresses were involved. The first IP address tried several login attempts (850) over a 20 days period, where the inter-arrival time between each attempt was nearly 30 min. The second IP address tried 174 login attempts over a 4 days period, where the inter-arrival time between attempts was nearly 30 min. In both the sessions, the number of login attempts per TCP flow was mostly one. A similar trend is observed in Dataset 2.

## 6 Conclusion

In this paper, authors have done a detailed analysis to understand network attacks, particularly distributed brute-force attacks. Detailed analysis was carried out to prove that the honeynet design is versatile enough to capture malicious attacks. A methodology was proposed to segregate individual botnets from network traffic. This methodology will help in understanding more about distributed brute-force attacks, which will provide a way for detecting these attacks. In future, the authors would like to verify and prove the methodology using clustering approaches.

## References

1. Calyptix. (2015). Top 7 network attack types in 2015. <https://www.calyptix.com/top-threats/top-7-network-attack-types-in-2015-so-far>.
2. McAfee. (2015). McAfee labs threats report. <https://www.mcafee.com/us/resources/reports/rp-quarterly-threat-q1-2015.pdf>.
3. Sadasivam, G., & Hota, C. (2015). Scalable honeypot architecture for identifying malicious network activities. In *2015 International Conference on Emerging Information Technology and Engineering Solutions (EITES)* (pp. 27–31). <https://doi.org/10.1109/eites.2015.15>.
4. Spitzner, L. (2002). *Honeypots: Tracking hackers*. Addison-Wesley Longman Publishing Co.
5. Abdou, A., Barrera, D., & van Oorschot, P. C. (2016). *What lies beneath? Analyzing automated SSH bruteforce attacks* (pp. 72–91). Cham: Springer International Publishing. [https://doi.org/10.1007/978-3-319-29938-9\\_6](https://doi.org/10.1007/978-3-319-29938-9_6).
6. Owens, J., & Matthews, J. (2008). A study of passwords and methods used in brute-force SSH attacks. Technical Report. <http://people.clarkson.edu/~jmatthew/publications/leet08.pdf>.
7. Rabadia, P., & Valli, C. (2014). Finding evidence of wordlists being deployed against SSH honeypots—Implications and impacts. In *12th Australian Digital Forensics Conference* (pp. 114–121). <http://ro.ecu.edu.au/adf/141>.
8. Sokol, P., & Kopčová, V. (2016). Lessons learned from correlation of honeypots' data and spatial data. In *2016 8th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)* (pp. 1–8). <https://doi.org/10.1109/ecai.2016.7861111>.
9. Javed, M., & Paxson, V. (2013). Detecting stealthy, distributed SSH brute-forcing. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security* (pp. 85–96). ACM.
10. Malecot, E. L., Hori, Y., Sakurai, K., Ryou, J. C., & Lee, H. (2008). (Visually) Tracking distributed SSH brute force attacks? In *Proceedings of the 3rd International Joint Workshop on Information Security and Its Applications (IJWISA 2008)* (pp. 1–8).
11. Saito, S., Maruhashi, K., Takenaka, M., & Torii, S. (2016). Topase: Detection and prevention of brute force attacks with disciplined IPs from IDs logs. *Journal of Information Processing*, 24(2), 217–226. <https://doi.org/10.2197/ipsjjip.24.217>.
12. Geolocation utilities (2017). <https://www.maxmind.com>.
13. Iana source port (2017). <https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml>.
14. Kippo. <https://www.cert.pl/en/news/single/in-depth-look-at-kippo-an-integration-perspective/>.

15. Seifert, C. (2006). Analyzing malicious SSH login attempts. <https://www.symantec.com/connect/articles/analyzing-malicious-ssh-login-attempts>.
16. Hofstede, R., Hendriks, L., Sperotto, A., & Pras, A. (2014). SSH compromise detection using NetFlow/IPFIX. *SIGCOMM Computer Communication Review*, 44(5), 20–26. <https://doi.org/10.1145/2677046.2677050>. <http://doi.acm.org/10.1145/2677046.2677050>.
17. Hansteen, P. N. M. (2013). The Hail Mary Cloud data. <http://bsdly.blogspot.in/2013/10/the-hail-mary-cloud-and-lessons-learned.html>.

# Efficient Data Transmission in WSN: Techniques and Future Challenges



Nishi Gupta, Shikha Gupta and Satbir Jain

## 1 Introduction

Wireless sensor networks (WSNs) are the collection of small sensor nodes and an infrastructural sink. The sensor nodes are equipped with equipments which can sense changes in surrounding areas. Such networks are also deployable in harsh conditions where human intervention is difficult or impossible. Sensor networks have been used in various monitoring systems ranging from military surveillance, area monitoring, animal tracking, underwater monitoring to home security, smart buildings, smart cities, medical applications and many more. These networks require collected data to reach the sink node via an efficient and reliable route.

Sensor nodes are deployed in the area of interest to sense events and record their data. When an event is recorded, nodes collect data, process it, identifies suitable multi-hop route and forwards data to the sink. Identification of an efficient route and data transmission to the sink is called routing. It is expected that the selected route enables data to be delivered without loss and in time. Routing consumes more energy and requires more computation than the sensing of data. Various routing algorithms have been proposed for different environments of WSNs [1].

Routing algorithm can be broadly classified as flat or hierarchical based on the duties performed by each node, and as query-based, multi-path-based, negotiation-based or QoS-based, depending on the operation of the protocol [2]. Clustering is a technique used to aggregate data in hierarchical routing and is an important param-

---

N. Gupta (✉) · S. Gupta · S. Jain  
Netaji Subhas Institute of Technology, Dwarka, New Delhi, India  
e-mail: nishigupta99@gmail.com

S. Gupta  
e-mail: shikha.gpt1@gmail.com

S. Jain  
e-mail: jain\_satbir@yahoo.com

eter for evaluation of routing protocol performance. Various clustering techniques have been analyzed [3], and benchmarks have been identified [4] for the clustering technique. Once the clusters are made, then the routes are decided to forward the data. Clustering and routing are two processes which require high energy and need to be optimized for better performance of the network system.

In flat routing, the routes are chosen based on pre-defined parameters while forwarding the data. Various swarm intelligence techniques, fuzzy logics and stochastic processes have been applied to identify optimal path. After the path is decided, data is forwarded on it with sink as the destination. In hierarchical routing, data is collected by nodes and sent to the elected cluster heads. The cluster head now forwards the data to the sink by selecting an optimal path based on various optimization techniques.

In this paper, we have studied various existing routing techniques and identified their drawbacks and advantages. Different protocols work better in different environments and with different goals of the network. Classification and advantages of particular technique will help the researchers to develop more efficient protocols with improved results. This paper also states the disadvantages which help in understanding the future scope and improvements in WSN.

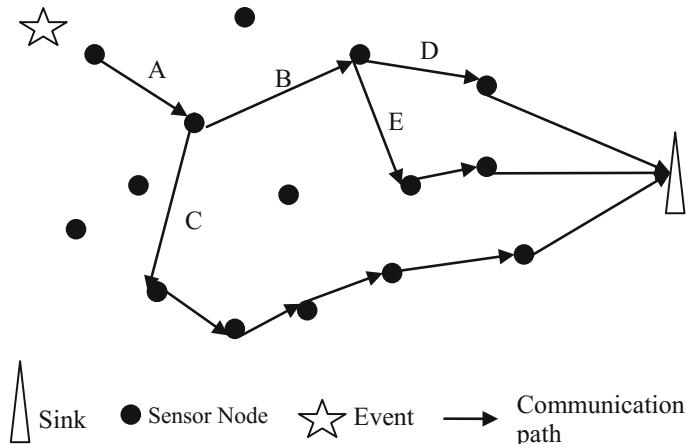
The remainder of the paper is organized as follows: Sect. 2 describes the routing in wireless sensor networks. In Sect. 3, different routing techniques have been explained, compared and analyzed. Section 4 identifies the existing drawbacks in routing techniques and future challenges. Section 5 concludes the paper.

## 2 Routing in WSN

In wireless sensor networks, the nodes collect data and transmit it in the network with sink as the destination. The nodes in WSN have the capability to directly send the data to the sink, but this communication requires high transmitting power energy, and energy is a major issue in WSNs; therefore, it is required to transmit data with energy efficiency. To transmit the data more efficiently in terms of energy, a multi-hop path is selected to forward data instead of direct communication.

In addition to energy, other parameters like QoS, security, delay and load balancing are also considered while selecting an optimal path for the data transmission. Routing is used to create a path from the source sensor node to the sink with the help of other sensor nodes of the network as intermediate nodes. This multi-hop path is based on trust of the neighbors. Node sends the data to the identified next hop, and it forwards data further until it reaches the sink.

The source node may select the next hop statically or dynamically, that is, with the help of pre-formed routing tables or using ad hoc forwarding. This next hop forwards the data with sink as the destination. After a path is chosen, the data is forwarded to ultimately reach the sink with maximum reliability or minimum delay or maximum security. This depends upon the application which is using the WSN and the requirement of the system.



**Fig. 1** Routing in a wireless sensor network

Figure 1 shows a scenario of WSN where an event is recorded by a sensor node. The node forwards the packet to the next hop via communication path A. This node has a choice to either forward the packet on path B or C. This decision will be made on the pre-defined parameters like QoS, shortest distance, maximum reliability, security and available bandwidth. On path B, the next hop again chooses from path D and E based on the parameters defined. The packet ultimately reached the destination, sink.

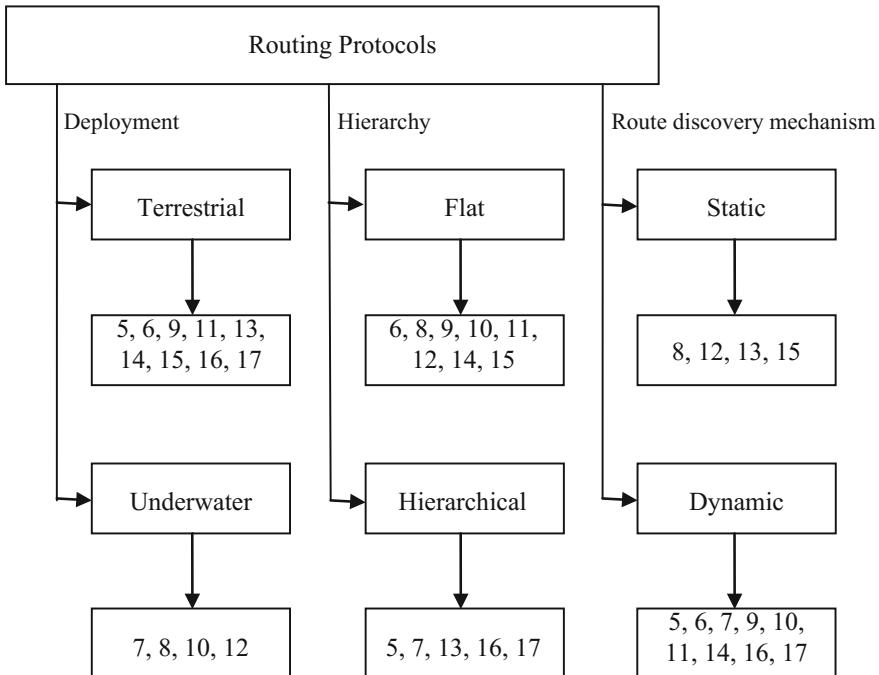
## 2.1 Classification

Figure 2 shows the classification of routing protocols in WSNs based on certain classification criteria.

Based on the deployment environment, WSNs can be classified as terrestrial and underwater WSNs. Due to inherent differences in the properties of these WSNs, different routing protocols are required to achieve optimal results.

**Terrestrial Routing:** These WSNs are deployed on land in two dimensions and work with radio waves. Any mobility in the network is introduced during deployment and according to the requirement. Global positioning system (GPS) can be integrated in such systems, and positions of nodes and sink are conveniently available with the system.

**Underwater Routing:** WSNs deployed underwater work with the acoustic signals and the network is three dimensional. They suffer from various limitations like low bandwidth, high error rate, noise, low signal strength and mobility of nodes due to water currents. GPS system is ineffective under water. Routing protocols in such



**Fig. 2** Classification of routing protocols in a wireless sensor network

lossy environment work with multiple sinks which finally send data to base station installed on land. These protocols also have to consider the depth of the nodes.

Based on the hierarchy of nodes, routing in WSNs can be classified as flat and hierarchical routing.

**Flat Routing:** In flat routing, every node in the network has equal resources and same work to do. It is a multi-hop routing, where nodes collaborate with each other to forward packets of data from sensor nodes to the sink. It is mainly implemented in networks with large number of nodes and is a data-centric routing.

**Hierarchical Routing:** Hierarchical or clustered routing divides the network in clusters. The member nodes sense data and forward it to their respective cluster heads for further delivery to the sink. Cluster heads do more computational work and hence, more energy is consumed. These protocols are more energy efficient when balancing is taken into consideration.

Routing can be classified as static and dynamic depending upon the route discovery procedure at every data packet generated.

**Static Routing:** On generation of a data packet at a node, the node discovers the optimal route to the sink and saves it. Every data packet is forwarded on that route only. This leads to energy imbalance but saves computational power required for route generation and optimization steps.

**Dynamic Routing:** In such routing, node searches for the best path every time and forwards the data. This saves time and energy in sending data on optimal paths. Quality of service becomes better, but energy consumption for path discovery becomes high. It is also beneficial in case of mobile network where nodes move from their place and new paths are created and destroyed with time.

## 2.2 *Advantages*

Routing deals with transmitting data from source to the sink. In addition to data transfer, it also plays an important role in the performance of the network. An efficient routing technique can provide several advantages to the network.

**QoS:** A QoS-aware routing emphasizes on quality of service. The delay in the network is reduced. Such routing scheme is necessary for real-time systems. Bandwidth is taken into account while choosing the routes. Such routing mechanisms use the resources in the system more efficiently, keeping in mind that the resources are the main limitations in WSNs.

**Security:** Routing is a trust-based mechanism where data is transferred hop-by-hop. Nodes forward the packets with the assumption that intermediate nodes are trustworthy and will forward the data to the destination. A secure routing procedure handles malicious nodes and keeps the network secure from external attacks also. It also ensures that malicious nodes do not waste the network resources.

**Energy balance:** Energy is a limited commodity in WSNs. It is needed that energy consumption is balanced in the network. A good routing procedure will identify overuse of energy on a path and will assign path with more available energy for the next data transfer.

**Energy efficiency:** Not only balanced energy consumption, but also efficient and minimum energy expense is important in WSNs. This improves the performance of the network by maintaining better connectivity and enhancing the lifetime of the network. A good routing algorithm expends minimum energy by reducing any overheads in the system.

## 3 Routing Techniques

Various routing algorithms have been found in the literature to efficiently transmit data from sensor node to the sink. Jose and Sadashivappa [5] present a basic hierarchical scheme with mobile sink which selects the cluster head (CH) on the basis of relative distance from member nodes, received signal strength and residual energy. CHs send data to one of the mobile sinks which is aggregate and forward it to BS. ACO-TR [6] adds trust value of neighbor as part of fitness function for the applica-

tion of ant colony optimization. Routing selects an optimal path based on this fitness function to forward data packets. TCA [7] obtains the initial topology creation by using an edge-constructed model. This distributed algorithm works in underwater WSNs and extracts a double-clustering structure from initial topology as part of routing. E-CARP [8] is an energy-efficient underwater protocol which selects best quality link to forward data to the relay (next hop). Data is stored in the cache of sensor nodes for later reference. A threshold for change in sensed data is defined to reduce traffic. Selection of a new relay node is not considered if environment does not significantly change. EDAL [9] works in two phases. A route is selected in the route construction phase using ACO, and route optimization phase uses tabu search as the optimization technique. Data is transmitted by using geographical forwarding scheme after applying compressive sensing on data. MobiSink [10] is an underwater localization free cooperative routing protocol. Nodes forward data to their neighbor that is closer to the sink and it acts as relay for data transmission. If the sink is in range, the node directly sends the data. Sink finally forwards data to the base station. ACO with ECPSO is proposed in [11], where ACO selects the most efficient and feasible path. Later, ECPSO is used to transmit data from node to sink and it also helps in recovery from failure of paths. EVA-DBR [12] is a depth-based and stateless routing protocol for underwater WSNs which avoids the network voids while routing. Void detection timer is used by a node to identify itself as void. This facilitates in selecting next-hop candidate node. TSSRM [13] selects a node as CH with high initial trust degree. Routes are constructed using trust packets and obtain optimal path. Routes are maintained to maintain connectivity by updating the values of trust degrees. In REAQ [14], the sink determines an optimal path to a node based on topology and number of times the node was used previously. Route is determined from sink to node, and then the data is sent on the reverse route. CCOR [15] identifies degree of node congestion and bottlenecks in network by using a queuing model. Optimal routes are selected on the basis of both the energy efficiency and congestion by using a link gradient. Traffic is distributed on different paths using a radius function to achieve load balancing. MRRCE [16] selects CHs using Steiner points as input to k-means algorithm. Nodes send data to CH, which chooses another CH closer to BS with maximum residual energy as next hop to transmit data to the sink. P-SEP [17] selects CH on the basis of location and distance information provided by the fog structure. CH is selected randomly for better performance, and minimum path is chosen between CH and fog node.

Table 1 shows the comparative analysis of the existing routing algorithms in wireless sensor networks. It also marks the advantages and disadvantages in the studied algorithms. The identified drawbacks also help in understanding the improvements which can be made to develop better protocols for the system.

**Table 1** Comparative analysis of routing techniques in wireless sensor networks

Algorithm	Advantages	Disadvantages	Environment	Approach	QoS	Security	Mobility	Energy balance
[5] Routing with mobile sink (RMS)	Energy efficient	No security No QoS	Terrestrial	Hierarchy	No	No	Yes	Yes
[6] ACO-based trustful routing (ACO-TR)	Energy efficient Secure transmission	No energy balancing	Terrestrial	Flat	Yes	Yes	No	No
[7] Topology control algorithm (TCA)	Energy efficient Reduced delay Better connectivity Better coverage	More cost No security No energy balancing	Underwater	Hierarchy	Yes	No	No	No
[8] Energy-efficient channel-aware routing (E-CARP)	Reduced traffic	No security No energy balancing	Underwater	Flat	Yes	No	No	No
[9]	Compressive sensing Reduced traffic	High overhead No security No energy balancing	Terrestrial	Flat	Yes	No	No	No
[10] MobiSink	Better throughput	High transmission loss High complexity No security	Underwater	Flat	Yes	No	Yes	Yes
[11] ACO with Endocrine cooperative PSO (ECPSOA)	Works better in dynamic environment	High complexity No security No energy balancing	Terrestrial	Flat	Yes	No	Yes	No

(continued)

**Table 1** (continued)

Algorithm	Advantages	Disadvantages	Environment	Approach	QoS	Security	Mobility	Energy balance
[12] Energy-efficient and void avoidance depth-based routing (EVA-DBR)	Energy efficient Reduced cost More reliable	Collisions No security No energy balancing	Underwater	Flat	Yes	No	No	No
[13] Trust sensing-based secure routing mechanism (TSSRM)	Trust-based Enhanced security More stable clusters	Mobility not considered	Terrestrial	Hierarchy	Yes	Yes	No	Yes
[14]	Energy efficient Reduced overhead More reliable Longer lifetime	No security	Terrestrial	Flat	Yes	No	No	Yes
[15] Optimizing routing based on congestion control (CCOR)	Energy efficient Low packet loss rate Reduced congestion	High delay No security No energy balancing	Terrestrial	Flat	Yes	No	No	No
[16] Multi-hop routing reducing consumed energy (MRRCE)	Energy efficient	No QoS No security No energy balancing	Terrestrial	Hierarchy	No	No	No	No
[17] Prolong stable election protocol (P-SEP)	Balanced energy Fairness Robust More stable	No QoS No security No energy balancing	Terrestrial	Hierarchy	No	No	No	Yes

## 4 Future Scope

Routing is an essential phase of wireless sensor networks to deliver the required results efficiently. Resources and network parameters affect the efficiency and performance of routing protocols. Certain limitations have been identified which will help in improving the routing by addressing and alleviating them.

**Mobility:** With mobile nodes, the routes determined for one data transfer become obsolete till the next transfer is required. The routing algorithm needs to compute a new route for every new communication. This results in increased delay and more consumption of computational power.

**Cost:** The cost in routing algorithms is in the terms of bandwidth, delay, jitter, packet drop ratio and other performance metrics. Based on the requirement of the system, a trade-off has to be done among the cost and quality of service and the resources. Routing table maintenance also leads to higher costs.

**Delay:** Delay is an important factor in real-time surveillance system and need to be minimized. It is difficult to always reduce the delay due to environmental factors, mobility of nodes and disconnections in the network. Also, formation of new routes for new communication also increases the delay.

**Bandwidth:** Bandwidth is a limited resource in WSNs. Routes should be selected to judiciously use the available bandwidth while maintaining the performance of the system. Paths with higher overall bandwidth may be chosen over the paths with higher bandwidth available only at the next hop.

**Energy:** Energy is the most crucial and limited resource in the WSN. Several trade-offs have to be done to maintain the energy of the system. Energy consumption and balanced dissipation of energy affect the lifetime, connectivity, converge and hence the performance of the complete system.

**Security:** WSNs work on trust among the nodes. Data is forwarded via intermediate nodes. Thus, security becomes a main concern while forwarding packets. Routing protocols need to identify threats and provide data and transmission security. It is an important parameter in crucial systems, such as a military surveillance system.

## 5 Conclusion

Routing is used in WSNs to identify an efficient multi-hop path from the node to the sink and forward data packets on it to deliver them successfully. Routing is a crucial process to deliver the data in time for real-time systems, securely for sensitive systems, with less computation for energy-constrained systems and optimizing resources for limited-resource systems. This paper has studied various available routing techniques in WSNs and analyzed them comparatively. We have identified the advantages that a good routing algorithm provide to the system and also various limitations suffered by them. Where energy balancing and energy efficiency is achieved by careful routing, it also requires trade-offs among delay, cost, bandwidth, energy and other

resources of the system. Identification of these challenges will allow researchers to develop more efficient algorithm for routing in WSNs which will better balance these parameters to improve the performance and scope of WSNs.

## References

1. Pant, Y., & Bhaduria, H. S. (2016). Performance study of routing protocols in wireless sensor network. In *2016 8th International Conference on Computational Intelligence and Communication Networks (CICN)*. IEEE.
2. Anisi, M. H., et al. (2017). Energy harvesting and battery power based routing in wireless sensor networks. *Wireless Networks*, 23(1), 249–266.
3. Gupta, N., et al. (2017). Clustering in WSN: Techniques and future challenges. In *2017 4th International Conference on Computing, Communication, Control and Automation (ICCUBEA)*. IEEE.
4. Gupta, N., et al. (2017). Benchmarks for evaluation of wireless sensor network clustering. In *2017 2nd International Conference on Smart Trends for Information Technology and Computer Communications (SmartCom)*. Springer CCIS.
5. Jose, D. V., & Sadashivappa, G. (2015). A novel scheme for energy enhancement in wireless sensor networks. In *2015 International Conference on Computation of Power, Energy Information and Communication (ICCP-EIC)*. IEEE.
6. Luo, Z., et al. (2015). An ant colony optimization-based trustful routing algorithm for wireless sensor networks. In *2015 4th International Conference on Computer Science and Network Technology (ICCSNT)* (Vol. 1). IEEE.
7. Rajalakshmi, P., & Logeshwaran, R. (2015). Performance analysis of cluster head selection routing protocol in underwater acoustic wireless sensor network. In *2015 2nd International Conference on Electronics and Communication Systems (ICECS)*. IEEE.
8. Yao, B., et al. (2015). An energy efficient routing protocol for underwater WSNs. In *2015 IEEE 12th International Conference on Ubiquitous Intelligence and Computing and 2015 IEEE 12th International Conference on Autonomic and Trusted Computing and 2015 IEEE 15th International Conference on Scalable Computing and Communications and Its Associated Workshops (UIC-ATC-ScalCom)*. IEEE.
9. Yao, Y., Cao, Q., & Vasilakos, A. V. (2015). EDAL: An energy-efficient, delay-aware, and lifetime-balancing data collection protocol for heterogeneous wireless sensor networks. *IEEE/ACM Transactions on Networking (TON)*, 23(3), 810–823.
10. Shah, P. M., et al. (2016). MobiSink: Cooperative routing protocol for underwater sensor networks with sink mobility. In *2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA)*. IEEE.
11. Kumar, J., Tripathi, S., & Tiwari, R. K. (2016). Routing protocol for wireless sensor networks using swarm intelligence-ACO with ECPSOA. In *2016 International Conference on Information Technology (ICIT)*. IEEE.
12. Ghoreyshi, S. M., Shahrabi, A., & Boutaleb, T. (2017). An underwater routing protocol with void detection and bypassing capability. In *2017 IEEE 31st International Conference on Advanced Information Networking and Applications (AINA)*. IEEE.
13. Qin, D., et al. (2017). Research on trust sensing based secure routing mechanism for wireless sensor network. *IEEE Access*.
14. Omar, M., Yahiaoui, S., & Bouabdallah, A. (2016). Reliable and energy aware query-driven routing protocol for wireless sensor networks. *Annals of Telecommunications*, 71(1–2), 73–85.
15. Ding, W., Tang, L., & Ji, S. (2016). Optimizing routing based on congestion control for wireless sensor networks. *Wireless Networks*, 22(3), 915–925.

16. Rezaei, E., Baradaran, A. A., & Heydariyan, A. (2016). Multi-hop routing algorithm using Steiner points for reducing energy consumption in wireless sensor networks. *Wireless Personal Communications*, 86(3), 1557–1570.
17. Naranjo, P. G. V., et al. (2017). P-SEP: A prolong stable election routing algorithm for energy-limited heterogeneous fog-supported wireless sensor networks. *The Journal of Supercomputing*, 73(2), 733–755.

# A Study of Epidemic Spreading and Rumor Spreading over Complex Networks



Prem Kumar, Puneet Verma and Anurag Singh

## 1 Introduction

Our world is saturated with the systems that are heavily complicated. Consider for example a democratic country that requires cooperation among millions of people, or the Internet that integrate billions of cell phones, computers, IoT devices, and satellites. It requires the mutual activity of billions of neurons in our brain for us to reason, comprehend, and being curious. Our genetic existence is connected with interactions between hundreds of species and metabolites within their cells. These systems are known as complex systems [1, 5], incurring the fact that it is difficult to find their behavior as a group from knowledge of the single components of the system.

Two well-known and widely used complex networks are:

- (1) Random networks
- (2) Scale-free network.

---

P. Kumar (✉) · P. Verma (✉) · A. Singh (✉)  
Department of Computer Science and Engineering,  
National Institute of Technology Delhi, New Delhi, Delhi, India  
e-mail: 151210022@nitdelhi.ac.in

P. Verma  
e-mail: 151210023@nitdelhi.ac.in

A. Singh  
e-mail: anuragsg@nitdelhi.ac.in

## 1.1 Random Networks

There exist mainly two definitions for random networks:

- (1)  $G(N, L)$  Model: Using  $L$  randomly placed edges, we connect  $N$  labeled nodes.
- (2)  $G(N, p)$  Model: A model introduced by Gilbert [4] in which each pair of  $N$  labeled nodes has a probability  $p$  of forming a link/edge between them.

In the  $G(N, L)$  model, the total number of links  $L$  is fixed while in the  $G(N, p)$  model, the probability  $p$  that whether the two nodes are connected is fixed. In the  $G(N, p)$  model, most network characteristics are easier to calculate but calculating average degree in  $G(N, L)$  model is simple,  $\langle k \rangle = 2L/N$ . The degree distribution of random networks follows the binomial distribution. For a network of  $N$  nodes

$$P(k) = \binom{N-1}{k} p^k (1-p)^{(N-1-k)} \quad (1)$$

## 1.2 Scale-free Network

A network whose degree distribution follows a power law, at least asymptotically, is called scale-free network [1, 2]. Which means that for large values of  $k$ , the relation between the fraction of nodes having  $k$  edges  $P(k)$  and  $k$  is

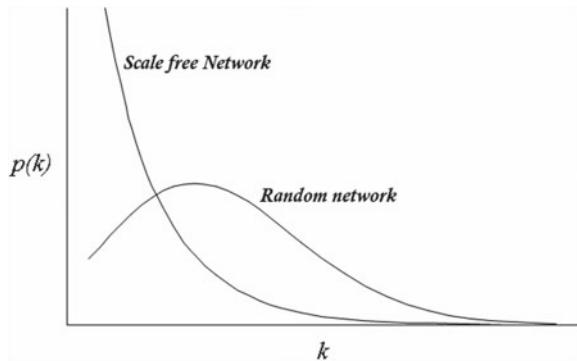
$$P(k) \sim k^{-\gamma} \quad (2)$$

where  $\gamma$  is in the range  $2 < \gamma < 3$ , but sometimes its value may be outside these limits. An important characteristic mostly found in every scale-free network is the presence of small number of vertices with a degree that exceeds the average in large extent which are also called as hubs. The hubs serve specific and important purposes in the network they are present, however, it varies with the domain. Clustering coefficient distribution of scale free network also follows power law and decreases as the node degree increases. The low-degree nodes in scale-free network are in dense sub-graphs, and hubs connect those sub-graphs to each other.

## 1.3 Properties

**Clustering Coefficient** Relationship among a node neighbors cannot be determined by its degree. Clustering coefficient provides this information that how densely the nodes in the graph cluster together. Local clustering coefficient  $C_i$ , measures the intensity of connectivity among node  $i$ 's immediate neighbors;  $C_i = 0$  shows that there is no link/edge among the node  $i$ 's immediate neighbors;  $C_i = 1$  shows that

**Fig. 1** Degree distribution of random and scale-free networks



every immediate neighbor of node  $i$  is connected to every other immediate neighbor of  $i$ .

To calculate  $C_i$ , we can simply take the ratio of number of edges between the immediate neighbors of  $i$  and the total number of edges possible between the immediate neighbors of  $i$ . In a random network, the clustering coefficient is equal to the probability  $p$ , the probability to have a edge between two randomly selected nodes. As there are  $\frac{k_i(k_i-1)}{2}$  possible links between the  $k_i$  neighbors of node  $i$ , the expected value of  $L_i$  is

$$\langle L_i \rangle = p \frac{k_i(k_i - 1)}{2} \quad C_i = \frac{2\langle L_i \rangle}{k_i(k_i - 1)} = p = \frac{\langle k \rangle}{N} \quad (3)$$

**Degree Distribution** The probability distribution of the degree of every node over the whole network is called degree distribution. Degree distribution of random and scale-free networks is shown in Fig. 1.

**Small World** The small world phenomenon, also known in public as six degrees of separation, has fascinated the world. It states that if we choose two random individuals on Earth, then the shortest path of acquaintances between them will contain at most six acquaintances. This result was obtained by the Stanley Milgram's experiments. In the terminology of network science, the small world phenomenon states that the distance between two randomly chosen nodes in a network is short.

#### 1.4 Characteristics of Some Real Network Data Available and Widely Used

See Table 1.

**Table 1** Characteristics of some networks, N = Number of nodes, L = number of links (edges),  $\langle k \rangle$  = average number of nodes connected per node,  $\langle d \rangle$  = average distance between two nodes,  $d_{max}$  = maximum distance between two nodes

Network	N	L	$\langle k \rangle$	$\langle d \rangle$	$d_{max}$
Internet	199,244	609,066	6.34	6.98	26
WWW	326,729	1,497,134	4.60	11.27	93
Power grid	4941	6594	2.67	18.99	46
Mobile_phone calls	36,595	91,826	2.51	11.72	39
Email	57,194	103,731	1.81	5.88	18
Science collaboration	23,133	93,437	8.08	5.35	15
Actor network	702,388	29,397,908	83.71	3.91	14
Citation network	449,673	4,707,958	10.43	11.21	42
<i>E. Coli</i> metabolism	1,039	5,802	5.58	2.98	8
Protein interaction	2,018	2,930	2.90	5.61	14

## 2 Analysis of Epidemic

An epidemic is the rapid spread of infectious disease or trend to a large number of people in a given population within a short span of time [11]. Analyzing an epidemic requires well-stated and practically feasible models.

Widely used models of epidemic spreading are:

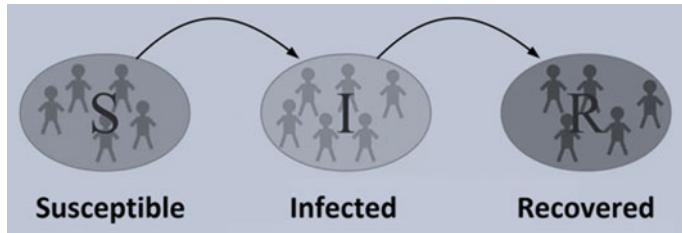
- (1) SIR model
- (2) SIS model.

### 2.1 SIR Model

In SIR model, there are three types of nodes:

- (1) Susceptible: The nodes which are vulnerable to infection.
- (2) Infected: The nodes which carry the infection and can infect susceptible nodes.
- (3) Recovered: The nodes from which infection has been removed and are no further prone to infection (Fig. 2).

Before the spreading of infection (before simulation), there exist only two type of nodes, mostly a few number of infected and large number of susceptible. If there exists a direct connection between a susceptible and infected, we do probabilistic



**Fig. 2** Illustration of SIR model

calculations to determine whether the node which is susceptible will get infected or not in a timestamp. An infected node can get converted into the recovered node in a timestamp, here the same probabilistic approach is taken into account which determines that the node is going to remain infected or will get recovered. After a node becomes recovered, it gets out of simulation and only gets importance while calculating the final result.

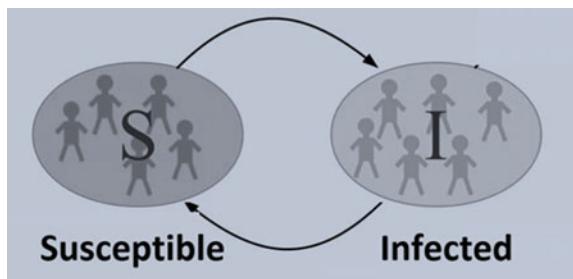
## 2.2 SIS Model

In SIS model [8], there are two types of nodes:

- (1) Susceptible: The nodes which have not been infected yet or has been recovered from infection.
- (2) Infected: The nodes which can infect susceptible nodes and can get converted into susceptible (Fig. 3).

This model comes into picture because some infections do not confer any long-lasting immunity. Such infections do not give immunization upon recovery from infection, and nodes become susceptible again.

**Fig. 3** Illustration of SIS model



### 3 Analysis of Epidemic Using SIR Model

Dataset Used:

We designed a simulation which generated the number of nodes susceptible, infected, and recovered after each timestamp. Initially, we infected random 70 nodes and defined a probability of infection:  $\lambda$  (probability of a Susceptible node to get infected if it is in contact with infected in a timestamp) and probability of recovery:  $\beta$  (probability of an infected node getting recovered in a timestamp) (Table 2).

We did the following steps in every timestamp.

Step 1. Filter the infected nodes.<sup>1</sup>

Step 2. For each node (see footnote 1).

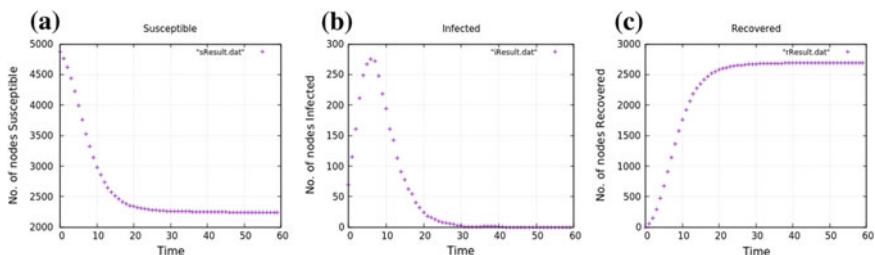
- (I) Filter the susceptible nodes<sup>2</sup> directly connected to it (see footnote 1).
- (II) For each node (see footnote 2).
  - (i) Generate a random number between 0.00 and 1.00.
  - (ii) If the random number generated is less than the  $\lambda$  then infect it (see footnote 2).
- (III) Generate a random number between 0.00 and 1.00.
- (IV) If the number generated is less than  $\beta$  then make it (see footnote 1) recovered.

We ended the program when there was no infected node left.

Finally, we took the average of 20 results and generated a file containing the data of variation of the number of different types of nodes with every timestamp. For this simulation, we took  $\lambda = 0.60$  and  $\beta = 0.85$ .

**Table 2** Dataset used

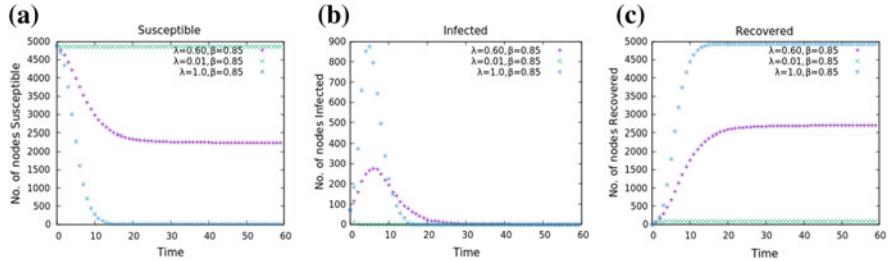
Network	N	L	$\langle k \rangle$	$\langle d \rangle$	$d_{max}$	$\ln N / \ln \langle k \rangle$
Power grid	4941	6594	2.67	18.99	46	8.66



**Fig. 4** Variation of number of **a** susceptible nodes, **b** infected nodes, and **c** recovered nodes at different time instants

<sup>1</sup>The selected infected node

<sup>2</sup>The selected susceptible node



**Fig. 5** Variation of number of nodes **a** susceptible versus timestamp, **b** infected versus timestamp and **c** recovered versus timestamp for  $\lambda = 0.01, 0.60$  and  $1.00$

We got the following plots using the generated data:

Further, we varied the rate of infection ( $\lambda$ ), kept the rate of recovery ( $\beta$ ) same, and observed the same graphs plotted in Fig. 4 (Fig. 5).

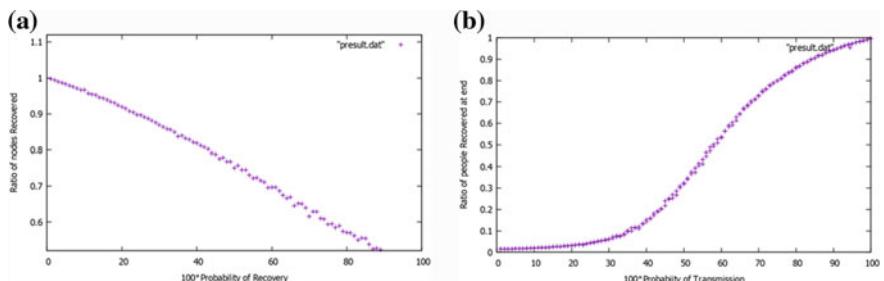
When  $\lambda$  is 0.01, the epidemic does not get a breakthrough as the number of susceptible nodes remain same.

But when  $\lambda$  is 1.00, all (approximately) nodes get infected.

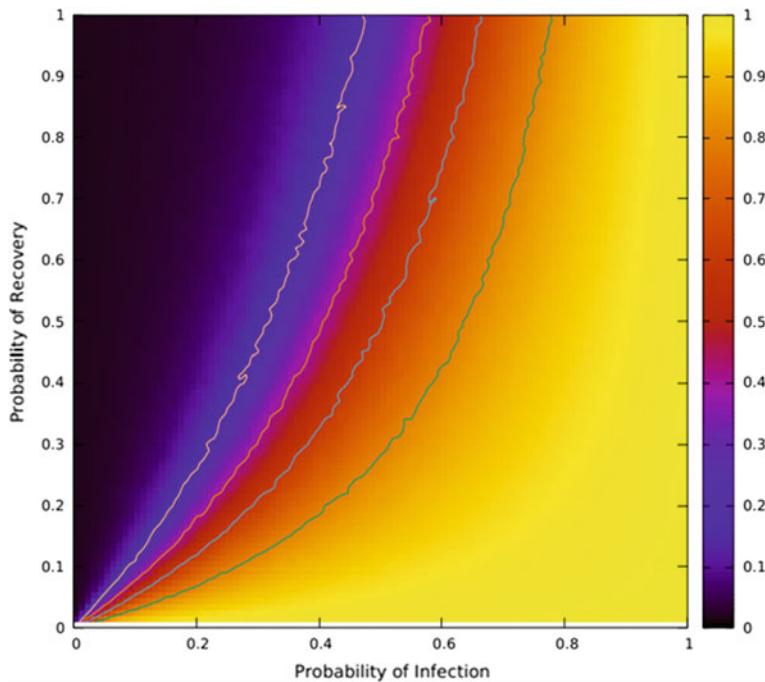
### 3.1 Impact of Epidemic: Curves for the Variations of Final Number of People Recovered Versus $\lambda$ (Rate of Infection)/ $\beta$ (Rate of Recovery)

We have used ratio of people recovered as an attribute to measure the impact of epidemic because it includes the total number of people who got infected, as an infected node will eventually get recovered at or before the end of the simulation. This is the best measure to check effectiveness and success of an epidemic (Fig. 6).

**Contour Plots** A contour plot is two-dimensional mapping of constant slices of a three-dimensional surface, where for a given z-axis value, a curve is plotted connecting the corresponding (x, y) coordinates where that z-axis value occurs. The plot given below is a colored contour plot for the variation of probability of infection,



**Fig. 6** Variation of ratio of people **a** recovered versus  $\beta * 100$  and **b** recovered versus  $\lambda * 100$



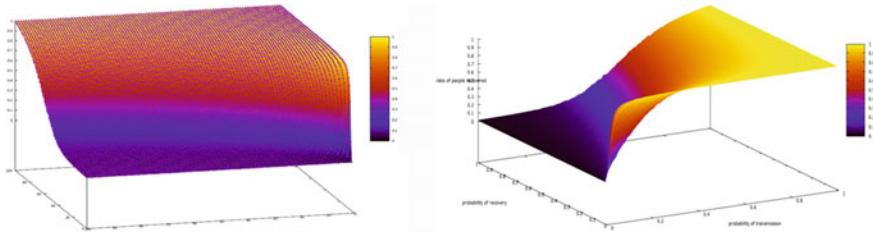
**Fig. 7 Impact of Epidemic:** Variation of ratio of people recovered versus  $\lambda$  (probability of infection) and  $\beta$  (probability of recovery)

probability of recovery versus ratio of total number of nodes recovered at the end. The colored scale shows the ratio of total nodes recovered. As the graph suggests when we increase the probability of infection and decrease the probability of recovery, the epidemic has larger impact. This plot (Fig. 7) also suggests us that how much should be the recovery rate for a given infection rate to stop an infection being an epidemic.

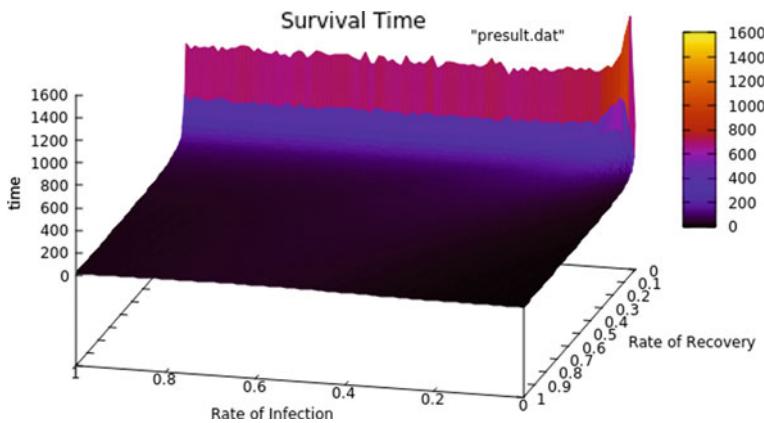
**3D Plots** These two plots (same plot in different angles, Fig. 8) are the 3D representations of the contour plot above or say that the contour plot given above is extracted from this plot.

### 3.2 Curves for the Variations of Total Timestamps Survived Versus Variations Other Properties

Number of timestamps survived is another important property. It tells that after how much time the epidemic will die. It is very interesting and important to observe how the survival time of epidemic varies with the probability of infection and probability of recovery.



**Fig. 8** The plots used to get Fig. 7. These plots are not given importance as 3D curves are not well presentable in 2D texts, but their 2D representation (Fig. 7) are useful



**Fig. 9** Variation of survival time versus  $\lambda$  and  $\beta$

Figure 9 shows that when the chances of recovery is very low, the epidemic lasts very long whether the rate of infection is high or low. The increase in chances of recovery decreases the life span of epidemic rapidly.

## 4 Analysis of Rumor Spreading

Rumor is an important form of social communications [9, 10, 12], and we can view rumor spreading as a stochastic process in social networks in which a data in any form is rapidly spread among the neighbors.

One of the standard models for the rumor spreading phenomenon was devised by Daley and Kendall, known as DK model [3, 6, 7] in which nodes in the network are categorized into three groups: ignorants, spreaders, and stiflers, which are denoted as S, I, and R, respectively:

1. S: People who are ignorant of the rumor, called ignorants.
2. I: People who actively spread the rumor, called spreaders.

3. R: People who have heard the rumor but are no longer interested in spreading it, called stiflers.

Mostly, it is similar to SIR model of epidemic. In this also, an ignorant connected to a spreader has a probability to become spreader, and there is a probability for a spreader to become a stifler, but it also has two other production rules.

1. If two spreaders meet, one of them becomes stifler



2. If a spreader and a stifler meet, both of them become stiflers.



We always have the conservation of nodes:

$$N = I + S + R \quad (6)$$

Dataset Used: Same data (power grid) used in epidemic spreading.

Here also we designed a simulation which generated the number of nodes susceptible/ignorant, infected/spreader, and recovered/stifler after each timestamp. Initially, we infected 70 random nodes, defined a probability of infection:  $\lambda$  (probability of a susceptible/ignorant node to get infected (become spreader) if it is in contact with infected/spreader in a timestamp) and the probability of recovery:  $\beta$  (probability of an infected/spreader node getting recovered in a timestamp/probability of a spreader node too become stifler when it meets a spreader/stifler) and run our simulation.

We did the following steps in every timestamp.

Step 1. Filter the spreader nodes<sup>1</sup>.

Step 2. For each node<sup>1</sup>.

(i) Filter the nodes<sup>2</sup> directly connected to it<sup>1</sup>.

(ii) For each ignorant node<sup>2</sup>.

(I) Generate a random number between 0.00 to 1.00.

(II) If the random number generated is less than  $\lambda$  then infect it<sup>2</sup>.

(iii) For each spreader node<sup>2</sup>.

(I) Generate a random number between 0.00 to 1.00.

(II) If the random number generated is less than  $\beta$  make the node<sup>2</sup> Stifler.

(iv) For each stifler node<sup>2</sup>.

(I) Generate a random number between 0.00 to 1.00.

(II) If the random number generated is less than  $\beta$  make the node<sup>1</sup> Stifler.

(v) Generate a random number between 0.00 to 1.00.

(vi) If the number generated is less than  $\beta$  then make it Stifler<sup>1</sup> (if it remained spreader after step (iv)).

We ended the program when there was no spreader node left.

For this simulation, we took  $\lambda = 0.60$  and  $\beta = 0.85$ . Finally, we took the average of 20 results and generated a file containing the data of variation of number of different types of nodes with every timestamp.

We got the following plots using the generated data.

The curves seem to be same as of SIR model of epidemic but when we observe closely, we can see that **the nodes get recovered faster, total number of nodes Recovered are low, and the simulation ends early** as we have more production rules which lead us to recovery.

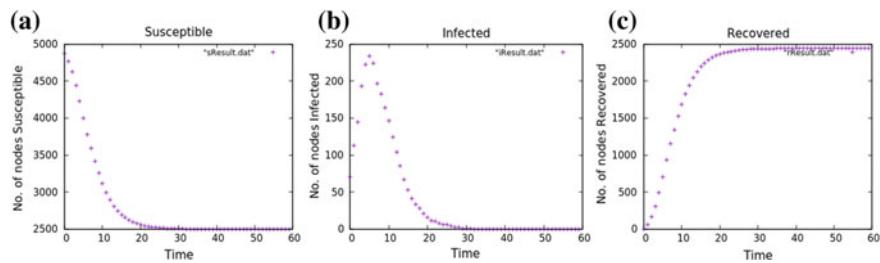
## 5 Comparison of Plots for the Random Networks and Scale-free Network

**Explanation of variation in curves.** The curves shown above are results of simulation of random network—own generated data and scale-free Network—<https://snap.stanford.edu/data/egonets-Facebook.html> having nearly equal average distance and average degree, but still the scale-free network rises much rapidly as compared to random network. This happens because there are hubs in the scale-free network, (hubs are the nodes which have a large degree). When a hub gets infected, it creates the breakthrough of epidemic due to its connectivity with a large number of nodes. But in the case of random network, maximum nodes have average degree, and there are a very few having a large degree.

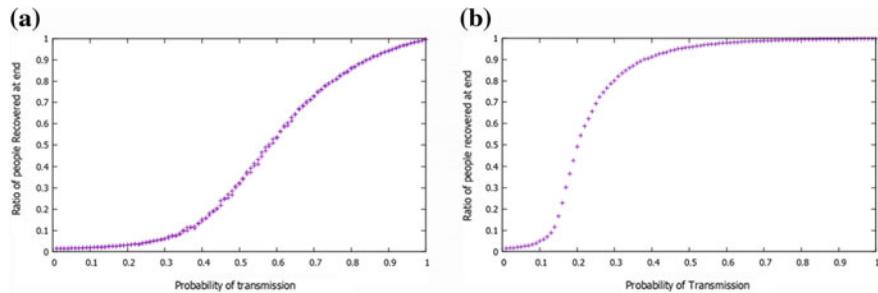
The degree distribution of the random network used is also shown in Fig. 12.

## 6 Conclusion

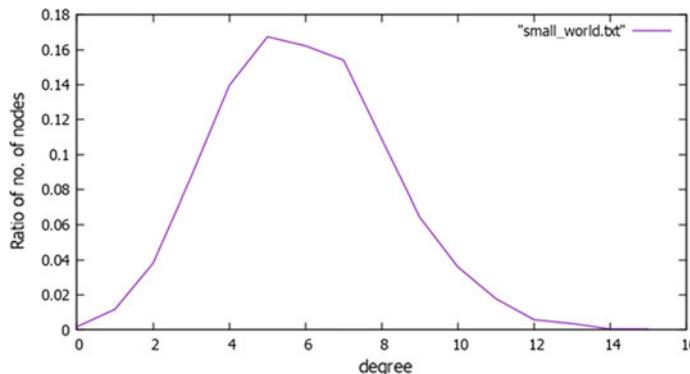
1. By looking at Fig. 4b, we can see that the number of nodes infected reaches at peak at a certain point and then decreases. In Fig. 4, we can see that **at the time number of infected nodes reaches at maximum the rate of decrease in number of nodes susceptible and the rate of increase in number of nodes recovered are maximum.** (Most negative slope in Fig. 4a at that point and most positive slope in Fig. 4c at that point.)
2. We can infer using Fig. 5b that whether the rate of infection ( $\lambda$ ) is 0.60 or 1.00 **the maximum number of infected nodes are present in the network at same time, independent of  $\lambda$ .**
3. In Fig. 7, the contour lines present can help us predict the appropriate pair of rate of recovery and rate of infection for a given impact of epidemic or say that it can help us predict the change required in rate of recovery for a given change in rate of infection if we want to keep the impact of epidemic same.



**Fig. 10** Variation of number of **a** ignorant/susceptible nodes, **b** spreader/infected nodes and **c** ignorant/recovered with timestamp



**Fig. 11** Variation of ratio of people recovered versus  $\lambda$  for **a** random network and **b** scale-free network



**Fig. 12** Degree distribution of the random network used in the simulation

4. Fig. 10 deduces that in the rumor spreading model **the nodes get recovered faster, total number of nodes recovered are low, and the simulation ends early** as compared to epidemic spreading model as we have more production rules which lead us to recovery.
5. The early breakthrough of rumor/epidemic in scale-free network as compared to random network (Fig. 11) describes why the information on social networks becomes viral so rapidly. Social networks have hubs like celebrities and highly followed pages which act as spreaders.

## References

1. Barabási, A.-L. (2016). *Network science*. Cambridge university press.
2. Barabási, A.-L., & Bonabeau, E. (2003). Scale-free networks. *Scientific American*, 288(5), 60–69.
3. Daley, D. J., & Kendall, D. G. (1965). Stochastic rumours. *IMA Journal of Applied Mathematics*, 1(1), 42–55.
4. Gilbert, E. N. (1961). Random plane networks. *Journal of the Society for Industrial and Applied Mathematics*, 9(4), 533–543.
5. Kim, J., & Wilhelm, T. (2008). What is a complex graph? *Physica A: Statistical Mechanics and its Applications*, 387(11), 2637–2652.
6. Maki, D., & Thompson, M. (2018, March). Mathematical models and applications: With emphasis on the social, life, and management sciences/Daniel P. Maki, Maynard Thompson.
7. Pittel, B. (1990). On a Daley-Kendall model of random rumours. *Journal of Applied Probability*, 27(1), 14–27.
8. Qu, B., & Wang, H. (2017). SIS epidemic spreading with heterogeneous infection rates. *IEEE Transactions on Network Science and Engineering*, 4(3), 177–186.
9. Singh, A., & Singh, Y. N. (2012). Nonlinear spread of rumor and inoculation strategies in the nodes with degree dependent tie strength in complex networks. [arXiv:1208.6063](https://arxiv.org/abs/1208.6063).
10. Singh, A., & Singh, Y. N. Rumor dynamics and inoculation of nodes in weighted scale free networks with degree-degree correlation. In *2013 International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, pp. 514–520. IEEE.
11. Vega-Oliveros, D. A., Berton, L., Vazquez, F., & Rodrigues, F. A. (2017). The impact of social curiosity on information spreading on networks. [arXiv:1706.07972](https://arxiv.org/abs/1706.07972).
12. Zhang, Y., Xiong, M., Xu, Y., Guan, J., Zhou, S. (2017). Role of individual activity in rumor spreading in scale-free networks.

# Medical Alert System Using Social Data



Kumar Abhishek, M. P. Singh, Prakhar Shrivastav  
and Suraj Thakre

## 1 Introduction

Social media is one of the most significant revolutions of the modern times. The awareness among people, which is the need of the hour, is achieved by it, convenient yet a controlled mechanism with its ever so complex evolution all over the world. All kinds of issues are shared and spread over the social web, including the medical awareness about health issues and diseases. However, as the number of diseases increased and the diagnosis procedure became more and more complex, the necessity of awareness among people has evidently become more important and crucial.

The social web provides awareness among people through posts and tweets related to the disease and health issues faced by people. Few such tweets do not have much impact on a person which is spread among other tweets. This paper is an approach presented to study and represent disease-related tweets, which has information about the disease being infected in a region or location, to compute a collective result in the form of region specific alert of a particular disease prominent in that region.

---

K. Abhishek (✉) · M. P. Singh · P. Shrivastav · S. Thakre

Department of CSE, National Institute of Technology Patna, Patna 800005, India  
e-mail: kumar.abhishek@nitp.ac.in

M. P. Singh  
e-mail: mps@nitp.ac.in

P. Shrivastav  
e-mail: prakhar134811@nitp.ac.in

S. Thakre  
e-mail: suraj134817@nitp.ac.in

## 2 Related Works

This paper builds upon previous research work done in the field of traffic analytics in “Traffic analytics using probabilistic graphical models enhanced with knowledge bases” [1]. The authors [1] have dealt with the lack of certainty, incompleteness, and dynamism in the domain knowledge of traffic data with Probabilistic Graphical Models. In order to build knowledge base supported from the ground up, a “top-down” approach to leverage available knowledge has been proposed which results in better medical predictions.

In “City notifications as a data source for traffic management” [1], the authors have talked about utilizing data from authorized, city-initiated data sources to implement an alert system that will be helpful in informing citizens to make informed travel in places lacking instrumentation for traffic monitoring. The push notification system is built using SMS updates being sent by Delhi Traffic Department using Mass SMS alert service SMS Gupshup. A major challenge here is of information extraction.

A major breakthrough in the extraction of information from Physical Cyber Social systems existing in a citywide infrastructure has been proposed in “Extracting City Traffic events from social streams” [1]. It talks about accessing instance-level domain knowledge by annotating it through a trained sequence labeling program that can extract event from the text.

Since the updates can vary in terms of structure and content to a large extent, preparing such sequence labeling tool that can take into consideration such large and varied data set is a challenging task.

India a developing nation faces an acute problem of medical health on its existing alert system. Lack of a proper tracking and monitoring system aggravates this issue and necessitates the need for a framework that can track real-time medical updates to inform commuters in making their precaution decisions based on latest movement updates sourced from physical cyber social systems.

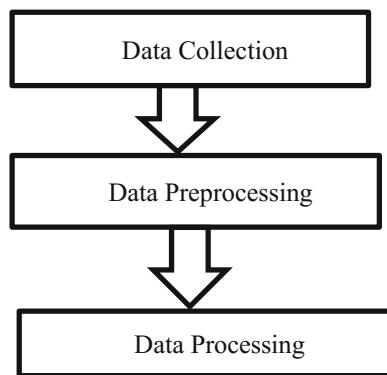
This paper aims at utilizing physical cyber social systems for obtaining information about the spread of disease in the metropolitan region; this obtained information will be analyzed to determine the precautions to be taken to help the common man tackle the disease in a better and easy way.

New Delhi, the capital city of India, has come up with a unique solution of posting updates on social channels regarding disease spread movement and latest updates providing a notification service to the general public. This system takes into consideration some official and unofficial sources of medical information, in this case, to alert commuters regarding medications.

## 3 System Implementation

The proposed work consists of three main stages as shown in the Fig. 1.

**Fig. 1** Stages of proposed work



Data collection stage focuses on the collection of raw tweeter tweets whereas data preprocessing steps deal with cleaning raw tweets to structure tweets with the help of Slang Removal and POS Tagging, and finally data processing steps deal with alert generation for infected areas. The detailed description of the stages follows next.

### 3.1 *Data Collection*

There are a lot of sources for gathering the information as previously mentioned in Sect. 1. Twitter is a social networking site where all the official and unofficial data are present. Twitter also provides the class of each tweet like here we are using the medical class so it helps us in this. So from Twitter, we get the huge amount of public data and use this data by the APIs given by the Twitter. To get the required tweet from all the tweets, these following steps will be followed.

#### 3.1.1 *Collect Raw Tweets*

LOKLAK server [2] is used to fetch the data available on social networking site like Twitter. If we try to retrieve tweets directly from Twitter, authorization is required, which makes us completely dependent on Twitter for raw data. LOKLAK server is a distributed peer-to-peer network of scrapers who act as clients to the Twitter (Fig. 2).

LOKLAK is a query-based server, which takes simple queries as an input and returns the result in JSON form. The data set specification is as follows.

The tweets posted nearby New Delhi Capital Region (NCR) are only collected.

The tweets that we grab are from Sep 01, 2016 to Oct 01, 2016.

The output in JSON format contains other details about the respective tweets. The JSON file is further processed, and only required details are filtered using Python.

```
{
  "timestamp": "2016-12-08T00:59:43.817Z",
  "created_at": "2016-12-07T22:30:09.000Z",
  "screen_name": "dengu_sengul",
  "text": "afilizken @kavist Erkek devlet yerine, fasist-katil devlet şiddeti denmeliydi. Lakin tüm tecavüzcüler katil tayyibe güveniyor.",
  "link": "https://twitter.com/dengu_sengul/status/806626879998480397",
  "id_str": "806626879998480397",
  "canonical_id": "",
  "parent": "",
  "source_type": "TWITTER",
  "provider_type": "SCRAPED",
  "retweet_count": 0,
  "favourites_count": 0,
  "place_name": "Lakin",
  "place_id": "",
  "text_length": 128,
  "place_context": "ABOUT",
  "place_country": "United States",
  "place_country_code": "US",
  "place_country_center": [
    -83.27110293007973,
    35.6452903550329
  ],
  "location_point": [
    -101.25489063619399,
    37.94057835728887
  ],
  "location_radius": 0,
  "location_mark": [
    -101.25461239990507,
    37.941561479545804
]
```

**Fig. 2** Output from LOKLAK server

Tweets are filtered on the basis of information obtained at the output such as the language of the tweet, which can be used to filter English tweets from the tweets of other languages and language probability determines the approximate probability of the tweet had a proper grammatical structure as per the grammar of the particular language. The important details in the output are tweet message, the time stamp, the location of the user, etc.

### 3.2 Data Processing

The tweets collected consist of slangs and other stops words. The raw tweets need to be preprocessed to that it can be used extract alert signals.

There is a trend of using slangs on social networking sites. Since Twitter restricts the user to write only 140 characters, so most of the tweets have slangs to express that is needed as fewer words as possible.

These slangs are not useful to us so here slang is converted into proper English words.

For example,

Sentence with Slang: “idk who u r?”

What does it mean: “I don’t know who you are?”

This can be achieved using any free slang dictionary available [3] available on the Internet and performing a lookup followed by replacing the slang with the corresponding word.

### 3.2.1 Part-of-Speech(POS) Tagging

After formatting the slangs, we are left with proper English sentences.

Each sentence is divided into tokens which are classified based on the corresponding part of speech.

The following algorithm [4] can be used as Part-of-Speech Tagger to classify the tokens.

1. Taking tokenized sentence as input, we traverse through tokens sequentially in the same order as that of the sentence.
2. Let  $h$  be the history of the tags assigned to the previous tags.
3. Let  $t$  be the tag assigned to the current token.
4. Many features  $f_i(h, t)$  are defined, like if token ends with -ing then  $f_i(h, t) = 1$ , otherwise  $f_i(h, t) = 0$ .
5.  $v$  is a vector which contains weight of the features and has dimension equal to number of features.
6. Define  $p(t_i|t_{i-1}, t_{i-2}, w_1, \dots, w_n) = \frac{\exp(\sum_i v_i f_i(h, t))}{\sum_{t'} \exp(\sum_i v_i f_i(h, t'))}$  which is the multinomial logistic regression, where  $w_i$  is the token in the sentence.
7. Calculate  $p(t_1 \dots t_n|w_1 \dots w_n) = \prod_{i=1}^n p(t_i|t_{i-1}, t_{i-2}, w_1, \dots, w_n)$  which is the probability that the sentence  $w_1, \dots, w_n$  has tags  $t_1, \dots, t_n$ .
8. Viterbi Algorithm is used to find tags  $t_1, \dots, t_n$  that maximizes the value of  $(t_1 \dots t_n|w_1 \dots w_n)$ . The tags thus obtained are the final tags assigned to the tokens.

Since all the location names and disease names are a noun, the tokens that are tagged as a noun are filtered as strings into a list. Also, many places, as well as diseases, have more than one word in the name which results in adjacent nouns. These adjacent nouns are clubbed into a single string which is verified in the next step.

For example, in Fig. 3, “Chandni” and “Chowk” are two adjacent nouns which are combined into a string “Chandni Chowk” which will be verified in the next step.

### 3.2.2 Extracting Location and Disease Names

From the list of strings obtained, we look up each noun in a disease and a location dictionary to verify whether the string is a disease or location.

**Fig. 3** Removing slangs

Translate text slang, internet slang, & acronyms

29 ppl died in Anand Vihar due to Dengue

Slang Free Translation

29 people died in Anand Vihar due to Dengue

The location dictionary can be obtained from the list of pin codes provided by Indian Postal Services [5]. Since in the scope of our discussion, we have gathered tweets only from Delhi. So we filter the pin code and corresponding location of Delhi Region, which will be used as location dictionary.

In a similar way, the disease dictionary is derived from Unified Medical Language System (UMLS). UMLS is a collection of many medical vocabularies which is viewed as a thesaurus and ontology of biomedical concepts [6]. It consists of a database and a set of software tools.

There are hundreds of thousands of terms about the disease and their description in UMLS. Since the proposed work is about disease alert system, we would consider only contagious diseases which spread through the air, water, insects, etc., thus reducing the size of the dictionary as well speed up the process.

Since disease, as well as location, is necessary for further processing, only those tweets which contain both location and disease were filtered.

If either of them is missing, then the tweet has insufficient data and thus discarded.

In Fig. 4, the first tweet has location as well as disease, whereas the second tweet has the only disease and has no location, thus discarded due to insufficient data (Fig. 5).

### 3.3 Structured Tweets

After performing the above steps, we are left out with tweets that are grammatically and semantically proper. These tweets are the final input that will be used further (Fig. 6).

```
(C:\Users\Suraj\Anaconda2) C:\Users\Suraj\Desktop\project>python nltk-test.py
21 patients infected by Chikunguniya in Chandni Chowk
[('21', 'CD'), ('patients', 'NNS'), ('infected', 'VBN'), ('by', 'IN'), ('Chikunguniya',
'NNP'), ('in', 'IN'), ('Chandni', 'NNP'), ('Chowk', 'NNP')]
```

**Fig. 4** Tagging the sentence

```
(C:\Users\Suraj\Anaconda2) C:\Users\Suraj\Desktop\project>python final.py
7 deceased in Siraspur due to Dengue Fever
Pass
2 died due to Malaria in this week
Fail
```

**Fig. 5** Filtering the sentence

```
21 patients infected by Chikunguniya in Chandni Chowk,2016-09-04T14:38:59z
2 died in Anand Vihar due to Malaria,2016-09-03T09:09:18z
13 patients infected by Chikunguniya in Chandni Chowk,2016-09-05T03:48:42z
7 deceased in Siraspur due to Dengue Fever,2016-09-02T03:08:07z
27 cases of Typhoid reported in Jafarpur,2016-09-06T03:20:39z
7 cases of Dengue Fever reported in Moti Bagh,2016-09-02T01:02:57z
6 patients infected by Chikunguniya in Chandni Chowk,2016-09-03T12:41:17z
Teenager dies of Malaria in Badusarai,2016-09-03T21:41:31z
9 cases of Tuberculosis reported in Paschim Vihar,2016-09-01T16:49:00z
12 cases of Typhoid reported in Quazipur,2016-09-06T09:01:17z
5 cases of Dengue Fever reported in Nasirpur,2016-09-02T14:43:12z
2 cases of Dengue Fever reported in Patparganj,2016-09-03T09:00:00z
Man deceased by Dengue Fever in Jafarpur,2016-09-07T19:02:18z
4 patients infected by Malaria in Chandni Chowk,2016-09-02T13:33:01z
8 cases of Malaria reported in Rampura,2016-09-06T01:09:54z
16 patients infected by Malaria in Chandni Chowk,2016-09-03T03:23:51z
```

**Fig. 6** Final input file

### **3.4 Data Processing**

This section describes step-by-step procedure to process the tweets. From the previous section, we have an input file having tweets which have proper grammatical structure. The input file is processed further. The steps are as follows.

#### **3.4.1 Storing Location and Disease**

In the previous section, location and disease dictionaries were used to determine whether a given word is a disease or a location or neither of them. This step can be improved by saving the location and disease names in variables to be used later.

#### **3.4.2 Other Information**

Any event related to disease is of two kinds, one is the infection of disease and another is the death of the infected person. This information can be retrieved by creating synsets of the nouns and verbs, other than the disease and location name, checking if it contains the word “die” or “infect” or other such synonyms.

For example, today 26 out of 97 patients are reported infected with Tuberculosis in AIIMS Delhi.

Information: infected

### 3.4.3 Level of Infection

In this section, we will discuss how to incorporate the different stages of any disease into a single level.

For death and infection, both from the same disease are treated like the different level of infection which is true. But if there are less number of death and high number of infection, it might be possible that level of infection is too high.

So, to convert the level of different events, infection, and death, into a single form, we consider death as equivalent to a certain number of infections.

As mentioned in “Calculation of infection Rates” at Utah Department of Health [7], the number of deaths and infection is merged in a single indicator as

$$\text{Level} = \text{Death} * 3.4 + \text{infection}$$

This level variable is further used to determine the alert level for the disease.

### 3.4.4 Determining the Alert Levels

The alert level variable calculated in the previous section is the indicator of the intensity of the disease in the particular region. Higher the intensity of the disease, higher is the rate at which the disease spreads. So, depending on the level variable we set some alert levels. Higher alert level signifies the need to alert more people in the surrounding regions.

There are three alert levels depending upon the value of the level variable (Table 1):

### 3.4.5 Pin Code Concept

The level of alert is known from the previous section; now the pin code of the location can be used to determine the areas that need to be alerted about the disease.

The location dictionary, discussed in Sect. 3.2.2, can be used to determine the pin code of the location.

Considering the fact that adjacent areas have consecutive pin codes, we have proposed the following concept of determining the region to be alerted based on the alert level.

If the alert level is low, the proposed infected areas to be alerted will have pin code with a difference of 1 from the location in the tweet.

**Table 1** Alert levels corresponding to the value of level variable

Variable value	Alert level
Less than 8	Low
Between 8 and 25	Moderate
Greater than 25	High

```
(C:\Users\Sura\Anaconda2) C:\Users\Sura\Desktop\project>python final.py
Timestamp: 2016-09-12T03:48:42z
Disease: Chikungunlya
Alert Level: High
Location: Chhattarpur
Precaution for Chikungunlya should be taken in the following areas
['Chhattarpur', 'Dhurunda Kalan', 'Dhurni', 'Gali 100', 'Ghuman Hora', 'Jasapur', 'Khera Dabur', 'Malik Pur', 'Mundela Kalan', 'Quazipur', 'Raota', 'Ujwa', 'C.S.K.M. School', 'Chandanpura', 'Chattarpur', 'Dera', 'Fatehpur Beri', 'Sanjay Colony Bhati Mines', 'Satbari', 'Sawan Public School', 'Aberhal', 'District Court Complex Dwarika', 'Dwarika Sec-6', 'Ali', 'Madanpur Khadar', 'Sarita Vihar']
Precautions:
Apply insect repellent.
Protect yourself day and night, because different mosquitoes feed at different times.
Wear long-sleeved shirts, pants, and a hat to minimize exposed skin.
Spray or wash clothing, bedding, and bed netting (but NOT your skin) with permethrin.
Stay and sleep in screened-in or air-conditioned rooms.
```

**Fig. 7** Output of the alert

Similarly, if the alert level is moderate, then the proposed infected areas will have pin code with a difference of 3.

And if the alert level is high, the proposed infected areas will have pin code with a difference of 5 (Fig. 7).

## 4 Conclusion

The approach proposed for an alert system for diseases uses tweets posted on social media which is a rich source of information in a metropolitan city. First, the tweets were acquired from social web and remove slangs to enhance it for language processing, determining location and disease from the tweet. Then the alert level is determined based on the number of patients affected by the disease, which is normalized beforehand to account for the two outcomes of a disease—infection and death. The range of the surrounding region to be alerted is proportional to the alert level.

The proposed method in this paper can also be applied to other cyber social sources such as Facebook, and e-newspapers. Data sources on these cyber social networks needed for the proposed system are very few, and many of them not being authentic.

This method fails to consider the information which contains the description of the symptoms prevailing in the specified region rather than the name of the disease. Such information can be processed along with developing a probabilistic model, such as Bayesian network, describing probabilistic relationships between symptoms and diseases. This kind of information may help in improving the accuracy of the system.

## References

1. Anantharam, P., Barnaghi, P., Thirunarayan, K., & Sheth, A. (2015). Extracting city traffic events from social streams. *ACM Transactions on Intelligent Systems and Technology*, 6(4), 1–27.
2. Loklak—Distributed Social Media Message Search Server, Loklak.org. <http://loklok.org>.
3. Slang Dictionary—Text Slang & Internet Slang Words, Noslang.com (2017). <http://www.noslang.com/dictionary/>.

4. Word Classification Algorithms, Stack Exchange (2014). <http://cs.stackexchange.com/questions/14037/word-classification-algorithms/27857#27857>.
5. All India Pincode Directory, Open Government Data (OGD) Platform India (2017). <http://data.gov.in/catalog/all-india-pincode-directory>.
6. Unified Medical Language System (UMLS), National Library of Medicine (2017). <http://www.nlm.nih.gov/research/umls/>.
7. Calculation of infection Rates, Long-Term Care Facility Resources UDOH-EPI, Health.utah.gov (2017). [http://health.utah.gov/epi/diseases/HAI/longterm\\_resources](http://health.utah.gov/epi/diseases/HAI/longterm_resources).

# **Part III**

## **Machine Learning**

**Dr. Swati Aggarwal Section Editor**

### **Editorial**

The human ability to learn and adapt to the ever-changing environment is the most important trait that has helped them to retain their supremacy in intelligence. The increasing capacity to acquire knowledge and become better learners has always been the desire of the insatiably curious mind. According to the Wikipedia definition of learning, the ability to learn is possessed by humans, animals and some machines. Due to loads of data availability and large computation power, the avarice to learn has now shifted towards machines also. This part includes research works that propose machine learning (ML)-based solutions that deal with varied domains. They showcase various real-life problems that yearn for solutions which are extensible and malleable to the ever-changing environments.

Kumar et al. discuss a new approach to the portfolio optimization problem. The objective of the portfolio optimization problem is to search for an optimal solution for investing an amount as a stipulated value if a set of assets or securities is given. This paper proposes a new approach that uses two nascent basal parameters which are derived from the return values obtained from the basic mean–variance model, and another parameter which is conditional value at risk.

Thermonuclear fusion provides a valuable alternative source of energy. Fusion reactors use a magnetic confinement device called tokamak. Timely and accurate classification of favourable and non-favourable discharges in a tokamak is very important for plasma operation. Sharma et al. review different types of disruptions in a tokamak and their prediction techniques. They also propose a real-time disruption classifier using a convolutional neural network to differentiate between favourable and unfavourable discharges for the ADITYA tokamak.

Owing to their ability to process large volume, velocity and very high variance data, ML tools and algorithms are heavily used for intrusion detection systems, giving greater efficiency. Azeez et al. present a comparative evaluation of three well-known algorithms, namely naïve Bayes, decision tree and random forest for network intrusion detection using Weka on the KDD CUP 1999 data set from DARPA.

Recent studies reveal that there are growing controversies over the real impact and benefits of advances in artificial intelligence for today's manufacturing industries. Wogu et al. discuss an interesting aspect of the effects on human development by super-intelligent machine operations in the twenty-first-century manufacturing industries.

Ensemble learning for unsupervised outlier detection is a less explored research field. Mukhriya et al. highlight the usefulness of ensemble learning for unsupervised outlier detection by empirical means.

### **Section Reviewers:**

Aman Jain  
Amit Dua  
Apoorvi Thankur  
Charan Kumari  
Danish Contractor  
Deepak Kumar Sharma  
Deepak Sharma  
E. Poovammal  
K. K. Biswas  
Khusboo Tripathy  
Mala Saraswat  
Manju  
Nicolas Rougier  
Priti Bansal  
Rahul Duggal  
Richa Sharma  
Ruchi Mittal  
Saurabh Aggarwal  
Shalini Bhaskar  
Shikha Jain  
Suma Dawn  
Sunny Rai  
Supriya Panda  
Swati Aggarwal  
T. T. Mirnalinee  
Tapan Das  
Usha Batra  
Vidhi Khanduja  
Vikas Maheshkar  
Vikas Saxena  
Vikas Thada  
Vivek Jaglan

# A Novel Framework for Portfolio Optimization Based on Modified Simulated Annealing Algorithm Using ANN, RBFN, and ABC Algorithms



Chanchal Kumar, M. N. Doja and Mirza Allim Baig

## 1 Introduction

The portfolio optimization problem has existed in research scenario for many years, and the concern of the problem is to obtain maximum returns. Since finding an exact solution for the problem using existing algorithms may lead to difficulties in forming a framework [7], which has motivated researchers to apply SA algorithm for finding an optimal solution for a given portfolio. The application of this algorithm for optimizing portfolio can be validated with empirical results on sample data [7]. Although there is a need to find an improvement in the framework which is to be adopted for the optimal solution, the main objective of investor lies in receiving a positive expected return [7]. This has motivated to introduce two nascent basal parameters for a framework that may be utilized for a portfolio optimal solution. These parameters have quadratic equations for representing the associated costs. Thus, the optimal solution may be represented as the one that finds the overall minimum cost. A modified SA algorithm is adopted to find optimal costs. This modified version makes use of significant parameter step. The value of this parameter is obtained by another valuable parameter radius [2]. Two algorithms are employed for finding the initial values of the radius: the ABC algorithm and RBFN that is further used in this paper for the related study. Further, the value of the parameter step is found using an equation involving the value of the radius. Moreover, this value of the parameter step is modi-

---

C. Kumar (✉) · M. N. Doja

Department of Computer Engineering, Jamia Millia Islamia, New Delhi, India

e-mail: kumarchanchal943@gmail.com

M. N. Doja

e-mail: mdoja@jmi.ac.in

M. A. Baig

Department of Economics, Jamia Millia Islamia, New Delhi, India

e-mail: mabaig@jmi.ac.in

fied by another factor, which is computed from an ANN structure. The ANN structure takes the following input parameters: delta cost, temperature, and control parameter. The modified SA algorithm computes the optimal cost using this value of step. Empirical results found using ABC algorithm and RBFN algorithm are presented in the paper. As a special neural network, radial basis neural network (RBFN) is widely used with artificial fish swarm algorithm (AFSA) to forecast the stock index [5]. Artificial bee colony (ABC) is one of the worthwhile and widely used optimization techniques based on swarm intelligence which is used for multi-objective problems [4, 6]. An optimization framework has been presented in [8] which uses ANN structure combining basic mean-variance model and disparity, and it is based on Lagrangian multiplier doctrine. Yevseyeva et al. [9] developed a new procedure for the many-targeted scheme using Sharpe ratio indicator for finding a solution for portfolio selection problem. A data mining framework has been presented at [10] for the valuation of large portfolios of variable annuity contracts. A method is developed by Zhao et al. [11] for portfolio selection and loan recommendation in the peer-to-peer (P2P) lending market. Das and Banerjee [12] introduced the meta-algorithms for online portfolio selection problem. These are iterative algorithms that are combination of a pool of base optimization algorithms. Chen and Hou [13] developed a newly designed version of genetic algorithms (GAs) that provided a solution for portfolio selection problem. A portfolio optimization problem becomes quadratic optimization problem with constraints, but the traditional algorithm is not able to handle the quadratic optimization problem. This problem was overcome by multi-objective evolutionary algorithm (MOEA) that can handle various constraints of portfolio optimization problem, namely cardinality, return constraint, budget constraint, floor and ceiling constraints, and trading constraints [14]. Qin et al. [15] introduced the uncertain mean semi-deviation model for portfolio optimization problem according to the skilled assessment of forthcoming return of the assets. Combinatorial optimization is widely used for the various optimization problems. One combinatorial optimization algorithm is given by Calvet et al. [17] that has root in SimILS approach. Classical optimization methods are unable to handle the optimization problems with a large set of constraints. In these cases, simulation optimization method is a good choice to complex optimization problems [18]. A hybrid bi-objective algorithm is presented in [19] for the portfolio optimization problem. Qiu et al. [20] proposed the quantile statistics-based robust optimization method for the portfolio selection problem. A framework of remaining sections is given as: Sect. 2 gives a description of modified SA algorithm and overview of other algorithms used in the framework along with a detailed overview of ABC algorithm and RBFN as used in the framework is also provided. Lastly, in Sect. 3 we present the empirical results.

## 2 Background

### 2.1 Portfolio Optimization

An evaluation model was proposed by Markowitz to deal with issues of portfolio optimization which was basically a framework to analyze the nature of investment under uncertainty [1]. Returns obtained from the investment are modeled as stochastic variables, and the past data is used to calculate expected values. Similarly, returns obtained from the overall portfolio are analyzed and its variance is calculated to measure risk. The risk is also analyzed by making a comparative analysis of the returns obtained from individual assets. Joint return distribution is used to calculate covariance matrix. Thus, a financial portfolio aims to achieve two objectives: minimizing the variance of portfolio return and maximizing the return obtained from expected portfolio. The aim of this framework was to obtain maximum expected return with minimum value of adversity [3].

$$\text{Min} \sum_{i=1}^n \sum_{j=1}^n \sigma_{ij} x_i x_j \quad (1)$$

Subject to

$$\sum_{i=1}^n r_i x_i = r_0 \quad (2)$$

$$\sum_{i=1}^n x_i = 1 \quad (3)$$

$$x_i \geq 0, \quad i = 1, 2, \dots, n \quad (4)$$

Here,  $r_0$  represents desired return. Equation (3) describes capital budget constraint on the proportions of the assets, and Eq. (4) makes sure no short selling of assets.

### 2.2 Problem Definition

The problem formulated in the proposed framework aims to achieve objective described in the following equation [21].

$$\text{Minimize overall\_cost}(\alpha_{new}, \beta_{new}) = \sum_{i=1}^2 C_i \quad (5)$$

$$C_1 = a_1 \alpha_{new}^2 + b_1 \alpha_{new} + c_1 \quad (6)$$

$$C_2 = a_2 \beta_{new}^2 + b_2 \beta_{new} + c_2 \quad (7)$$

$C_1$  represents a cost parameter that tells about the cost associated with basal parameter  $\alpha_{new}$ , which has an indexed value correlated with maximum and minimum returns that are obtained from the basic mean-variance model. The cost parameter  $C_2$  indicates about the inclusion of uncertainty in the proposed framework, and it is correlated with basal parameter  $\beta_{new}$ , which extracts its value from conditional value at risk. Depending upon optimal values of  $\alpha_{new}$  and  $\beta_{new}$ , the objective is to minimize the overall cost. This is achieved by employing a modified dialect of simulated annealing algorithm. This algorithm uses a novel structure for finding the optimal values. This structure uses the significant parameter step and radius. Two approaches based on ANN and RBFN are presented for computing a modifying factor, which is being used for changing the value of parameter step as used in the modified SA algorithm.

1. Identify two nascent basal parameters

Parameter	Description
$\alpha_{new}$	This is derived from maximum and minimum return values obtained from mean-variance model.
$\beta_{new}$	This is derived from significant parameter conditional value at risk in the proposed framework.

The maximum and minimum return values of the mean-variance model is computed from 12-month data series in respect of randomly selected 10 assets listed on the National Stock Exchange, Mumbai, India [3].

2. Objective of the proposed framework minimizes overall cost  $(\alpha_{new}, \beta_{new}) = \sum_{i=1}^2 C_i$

$$C_1 = a_1 \alpha_{new}^2 + b_1 \alpha_{new} + c_1$$

$$C_2 = a_2 \beta_{new}^2 + b_2 \beta_{new} + c_2$$

3. Find the optimal value of overall cost using modified SA algorithm.

4. The modified SA algorithm uses the parameter given below:

radius, step

5. Radius is computed using any one of the following algorithms.

- I. ABC algorithm,
- II. RBFN.

6. The value of the parameter step is being computed using the following equation [2].

$$\text{step} = \frac{\text{radius} * (r_2 - r_3)}{\sqrt{(r_2 - r_3)^2 + (r_1 - r_3)^2 + (r_2 - r_1)^2}}$$

7. Value for parameter step is being modified in the proposed framework using the following equation:

$$\text{modified step} = \text{step} * \text{factor}_1$$

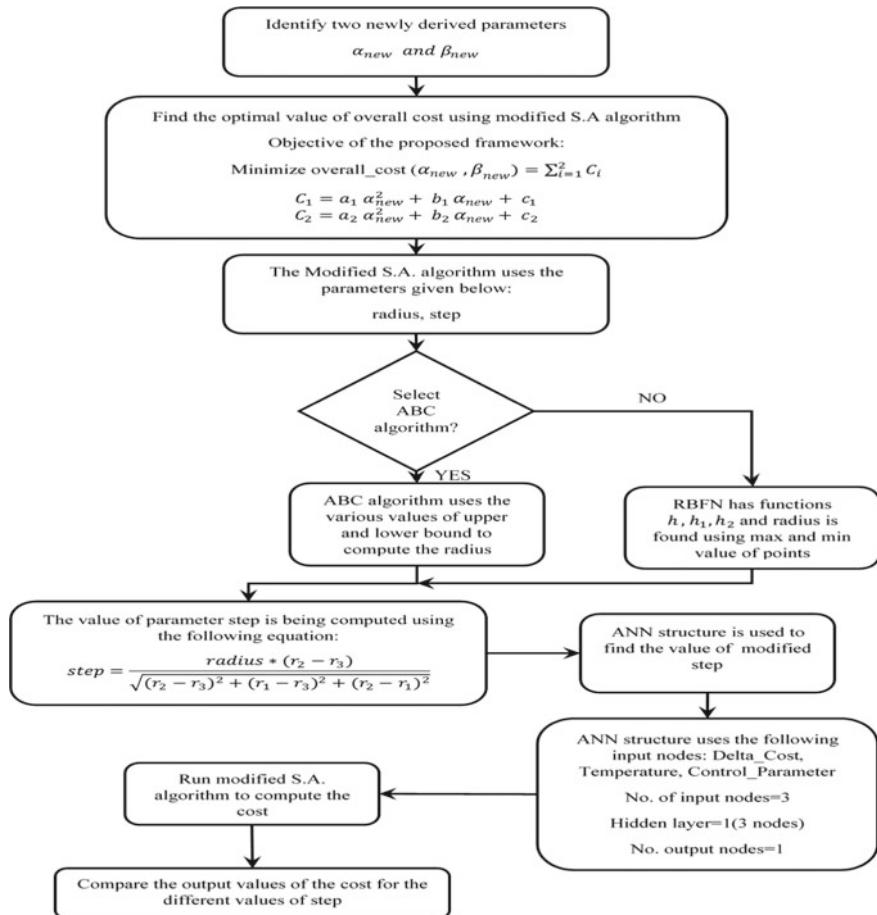
The value of  $\text{factor}_1$  is computed using ANN structure based on BPN algorithm. The ANN structure has the following input nodes:

I. Delta cost	The value of this parameter is also used in SA algorithm.
II. Temperature	The value of this parameter has significance in controlling the iterations in the modified SA algorithm.
III. Control parameters	This value may be appropriately chosen and utilized to have controlled characteristics of ANN structure.

The usage of newly constructed modified SA algorithm for finding an optimal value of the objective in the proposed framework is justified as SA algorithm is capable of generating the most suitable solution for a given optimization problem. These categories of algorithms are preferred choice for computing optimal solution in portfolio optimization (Fig. 1).

### 2.3 Module 1. (Algorithm 1) Modified Simulated Annealing Scheme

Parameter	Definitions
<i>step</i>	Value used in calculating value of P in <i>random_move()</i> function.
<i>temp</i>	Value used in basic simulated annealing for representing temperature.
<i>min_t</i>	Minimum value of temperature.
<i>del, ln_alpha</i>	Value used in calculating initial value of temperature.
<i>t<sub>0</sub></i>	Value of initial temperature.
<i>t<sub>2</sub></i>	Value of initial temperature found after calling frozen function.
<i>alpha</i>	Value used for modifying temperature in each iteration.
<i>P</i>	Value of the parameter that is to be optimized using modified simulated annealing (range = $10.0 \leq P \leq 150.0$ ).
<i>P<sub>2</sub></i>	Value of the parameter <i>P<sub>2</sub></i> is found in the neighborhood of <i>P</i> using <i>random_move()</i> function.
<i>a, b, c</i>	Coefficient used in the Eq. (8).
<i>cost, cost<sub>2</sub></i>	Cost calculated using Eq. (8) and new value of cost using modified SA.
<i>k</i>	Value used in modified SA representing Boltzmann constant normalized value used = 1.38.
<i>delta_cost</i>	$cost_2 - cost$ .
<i>counter</i>	Parameter used for representing iteration number.
<i>dt</i>	Value of the expression $\exp(-\text{delta\_cost}/(k * \text{temp}))$ .
<i>test</i>	Random number generated using <i>rand()</i> function range = 0.0–0.99
Step 1:	Initialization stage //calculate initial value of temperature



**Fig. 1** Overall structure for proposed framework

$$t_0 = (-\text{del}/\ln_{\text{alpha}})) \quad (8)$$

For the given value of P=50, calculate cost given by the following equation:

$$\text{cost} = aP^2 + bP + c \quad (9)$$

- Step 2: Call *frozen()* function and calculate the value of  $t_2$   
 $t_2 = \text{frozen}()$   
//Outer while statement  
Execute step 2.1 through 2.8 until (termination condition)  
termination condition is  $t_2 == 0$

2.1: Initialize the value of counter  
 Counter = 1  
 //Inner while statement  
 Execute step 2.2 through 2.7 until (counter  $\leq$  COUNTER\_MAX)  
 2.2: Find the value of  $P_2$  using the function random\_move

$$P_2 = \text{random\_move}(P)$$

2.3: Calculate the value of new cost represented by parameter cost<sub>2</sub>

$$\text{cost}_2 = a P_2^2 + b \cdot P_2 + c \quad (10)$$

2.4: Calculate the value of *delta\_cost*

$$\text{delta\_cost} = \text{cost}_2 - \text{cost} \quad (11)$$

2.5: Display the values of *delta\_cost* and temperature  
 Calculate the value of dt

$$dt = \exp((-delta\_cost / (k * temp))) \quad (12)$$

2.6: Found the value of random number = *test*

$$test = (\text{float})(\text{rand()} \% 100) \quad (13)$$

2.7: The range of random number (*test*) is:  $0.0 \leq test \leq 0.99$   
 Decide the value of P and cost for next iteration in inner while statement

*IF* ( $dt > test$ )      *THEN*  
 {             $P = P_2;$   
                $cost = cost_2;$   
 }  
 Increment the value of counter

2.8: counter++;  
 Calculate the value of temperature for next iteration in outer while statement

$$temp = (\alpha * temp) \quad (14)$$

Calculate the value of  $t_2$  for next iteration in outer while statement by calling *frozen()* function.

$$t_2 = \text{frozen}()$$

Step 3: Display the values of output parameters temperature, cost, and parameter P

## 2.4 Module 2. Definition of Function frozen()

Local parameters used within the function	Definitions
status	This variable is for return value given by <i>frozen()</i> . This return value has integer type.
<i>min_temp</i>	The minimum value of temperature used in the function.
<i>Ep</i>	The value used to check whether temperature value is close to <i>min_temp</i> .
<i>t<sub>2</sub></i>	The value of ep used in the function is 0.1.
Step 1:	It is subtraction of temperature and minimum temperature.
Step 1:	Calculate the value of <i>t<sub>1</sub></i> using the following equation:

$$t_1 = \text{fabs}(\text{temp} - \text{min\_temp})$$

Step 2:	Display the value of <i>t<sub>1</sub></i> Decide the value of the parameter status
	$\begin{aligned} & \text{IF } (t_1 < ep) \quad \text{THEN} \\ & \quad \{ \quad \text{status} = 1; \\ & \quad \quad \text{cost} = \text{cost}_2; \\ & \quad \} \\ & \text{ELSE} \quad \text{status} = 0; \\ & \text{Return the value of the parameter status.} \end{aligned}$

## 2.5 Module 3. Description of Steps Used for Finding the Modified Value of the Parameter Step.

Parameter	Definitions
<i>r<sub>1</sub>, r<sub>2</sub>, r<sub>3</sub></i>	These parameters represent the expected return of assets.
<i>step</i>	Value used in calculating value of P in <i>random_move()</i> function.
<i>radius</i>	We chose to construct a ball around each current solution and to restrict all neighbors to lie on the surface of this ball. The Euclidian length of each move is now simply determined by the radius of the ball [2].

- ann\_output* The output obtained from ANN algorithm.  
*Modified step* The modified value of step after multiplying with *ann\_output*.  
 Step 1: Compute the value of parameter radius using ABC algorithm.  
 Step 2: Compute the value of parameter step using the following equation [2]:

$$step = \frac{radius * (r_2 - r_3)}{\sqrt{(r_2 - r_3)^2 + (r_1 - r_3)^2 + (r_2 - r_1)^2}} \quad (15)$$

- Step 3: Compute the value of modified step using the following equation:

$$step = ann\_output * step \quad (16)$$

## 2.6 Module 4. Definition of *random\_move()* Function.

- Parameter Definitions  
*step* Value used in calculating value of P in *random\_move()* function.  
*P\_dash* The computed value of P after executing *random\_move()* function.  
 Step 1: Use the value of the step as determined in module 3.  
 value of step = modified step or value of step  
 Step 2: Compute the value of *P\_dash* using the following equation:

$$P\_dash = P - step$$

- Step 3: Output the value of *P\_dash*

## 2.7 Module 5. Computing the Radius Using Radial Basis Function Network (RBFN) with the Different Values of Lower Bound (*lb*) and Upper Bound (*ub*)

- Parameter Definitions  
*x<sub>0</sub>, sigma, ax* Parameters used in calculating the value of RBFN function.  
*h, h<sub>1</sub>, h<sub>2</sub>* Output values of RBFN function.  
*x<sub>i</sub>* Input values in RBFN function.  
*n* Represent the size of *x<sub>i</sub>*.  
*d* The values of distance parameter used in RBFN.  
*w* Parameter used for associating weight.  
*M<sub>0</sub>, M* Parameters used for calculating the value of parameter w.  
*f, f<sub>1</sub>, f<sub>2</sub>* Output values of RBFN.  
*f<sub>n</sub>* Array containing *f, f<sub>1</sub>* and *f<sub>2</sub>*.

*mx, mn*

Values of the upper bound and lower bound.

*radius*

Output parameter.

Step 1:

Initialize the values of the following parameters

Parameters:  $x_0, \sigma, ax$

Step 2:

(a) Compute the value of a parameter  $h$  using the following equation:

$$h = \exp(-0.5(ax - x_0)^2/\sigma^2) \quad (17)$$

The value of  $x_0$  used is  $x_0 = -1$ .

(b) Compute the value of parameter  $h_1$  using the following equation:

$$h_1 = \exp(-0.5(ax - x_0)^2/\sigma^2) \quad (18)$$

The value of  $x_0$  used is  $x_0 = -0.5$

(c) Compute the value of parameter  $h_2$  using the following equation:

$$h_2 = \exp(-0.5(ax - x_0)^2/\sigma^2) \quad (19)$$

The value of  $x_0$  used is  $x_0 = 1.0$

Step 3:

Initialization of the values of the following parameters; parameter is  $x_i, d$ .

Step 4:

(a) Compute the value of parameter  $M_0, M$  using the following equations:

$$M_0 = \text{abs}(x_i * \text{ones}(1, n) - \text{ones}(n, 1) * x'_i) \quad (20)$$

$$M = (1/\sqrt{2 * P_i}) * \exp(-0.5 * M_0^2) \quad (21)$$

(b) Compute the value of  $w$  using the following equation:

$$w = \text{pinv}(M) * d \quad (22)$$

(c) Compute the values of parameters  $f, f_1, f_2$  using the following equations:

$$f = w * h \quad (23)$$

$$f_1 = w * h_1 \quad (24)$$

$$f_2 = w * h_2 \quad (25)$$

(d) Compute the value of upper bound and lower bound from function ( $f_n$ ).

$$f_n = [f \ f_1 \ f_2] \quad (26)$$

- Step 5: Compute the value of upper bound and lower bound from function ( $f_n$ ).  
 Step 6: Compute the value of radius using the following equation:

$$\text{radius} = (ub - lb)/2 \quad (27)$$

## ***2.8 Module 6. Overview of Improved ABC Algorithm***

The improved ABC algorithm uses two significant parameters, upper bound and lower bound, for computing the value of radius.

- Step 1: Initialization of ABC control parameter. Colony size, the number of sources equals the half of the colony, a food source which could not be improved through ‘limits’ trials is abandoned by its employed bee, no. of epochs.
- Step 2: Problem-specific variables  
 Cost function to be optimized, the number of parameters of the problem to be optimized, lower bounds of the parameters, upper bounds of the parameters.
- Step 3: Creating the search space (food source)  
 3.1: Generation of initial population  
 Lower limit of food sources, upper limit of food sources, and generation of food sources (variable population).  
 3.2: Objective value computed for all food sources.  
 Fitness computed for all sources.  
 3.3: Rest Trail counters.  
 3.4: The best food source is memorized.  
 3.5: Employed bee phase.  
 3.6: Generated parameter fixed within limit if boundary limits are violated.  
 3.7: Evaluate new solution.  
 3.8: Greedy selection is applied between the current solution and its modified solution.  
 3.9: Generated parameter fixed within limit if boundary limits are violated.  
 3.10: Evaluate new solution.  
 3.11: Greedy selection is applied between the current solution and its modified solution.  
 3.12: The best food source is memorized.  
 3.13: Scout bee phase

## ***2.9 Module 7. Overview of Back-propagation Network***

The various parameters used in the training algorithm are as follows:

Parameters Definitions

$a$	Input layer nodes
$a$	$(a_{i1}, a_{i2}, a_{i3})$
$a_{i1}$	$\Delta_{cost}$
$a_{i2}$	$Temperature$
$a_{i3}$	$Control\_parameter$
$s$	Node of output layer
$B_k$	Desired output $u_k$
$B_j$	Error
$\alpha$	Parameter used for rate representing learning curve
$t_{0j}$	Hidden layer bias
$C_j$	Layer used for hidden nodes
$e_{0k}$	Output bias
$u_k$	Layer for output nodes

Back-propagation network (BPN) [16] algorithm is described below. Different phases are shown below:

Provide the initial values of weights used in the algorithm.

- Step 1: Select the values of the weights using random numbers.
- Step 2: Execute while statement till the condition is false; Execute step 3 to step 10.
- Step 3: Execute step 4 to step 9 for each pair of inputs  
Stage 2 representing the second phase of the algorithm: phase feed-forward.
- Step 4: Description of input unit  
Received signal (input) =  $a_i$   
Output this value to nodes in the hidden layer.
- Step 5: Find the value of the node in the hidden layer as follows:

$$C_{-inj} = t_{0j} + \sum_{i=1}^n a_i t_{ij}$$

Further, use the value of an activation function

$$C_j = f(c_{inj})$$

- Output this value to nodes in output layer.
- Step 6: Calculate the value of nodes in the output layer

$$V_{-ink} = e_{0k} + \sum_{j=1}^p c_j e_{ik}$$

Further, use the value of an activation function

$$V_k = f(v_{-ink})$$

Stage 3 representing phase 3: Errors are transmitted back.

Step 7: Calculate the value of term representing error:

$$B_{-inj} = (t_k - v_k) f(v_{-ink})$$

Step 8: Find the sum of delta inputs used in the hidden layer

$$B_{-inj} = \sum_{j=1}^n B_i e_{jk}$$

Calculate the value of term representing error.

$$B_j = B_{-inj} f(c_{-inj})$$

Find the new values of weight and updated values of biases.

Step 9: The new value of the weight is calculated as given below:

$$\Delta e_{jk} = \text{alpha } B_k c_j$$

The new value of the bias is calculated as given below:

$$\Delta e_{0k} = \text{alpha } B_k$$

$$\text{Thus, } e_{jk(new)} = e_{jk(old)} + \Delta e_{jk}$$

$$e_{0k(new)} = e_{0k(old)} + \Delta e_{0k}$$

Use the new values of weights and bias in the hidden layer.

The term used for correcting weights is given below:

$$\Delta t_{ij} = \text{alpha } B_j a_i$$

The term used for correcting bias is given below:

$$\Delta t_{0j} = \text{alpha } B_j$$

Therefore,

$$t_{ij(new)} = t_{ij(old)} + \Delta t_{ij}$$

$$t_{0j(new)} = t_{0j(old)} + \Delta t_{0j}$$

Step 10: Check the conditions for terminating the algorithm.

### 3 Empirical Results

The heuristic approach used in the framework is composed of the following improved algorithms:

- I. Modified SA,
- II. Modified ABC algorithm,
- III. Modified RBFN.

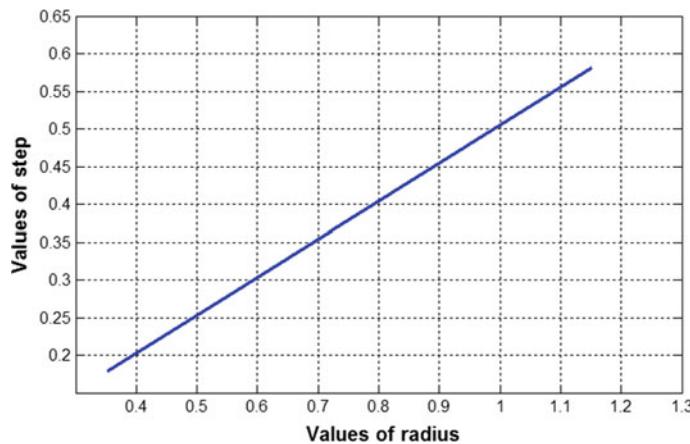
The first algorithm is implemented in ‘C’ language, and it is used to compute optimal values of the cost depending upon selected values of the parameters ‘step.’ The remaining two algorithms are coded in MATLAB and are used to find various values of the parameter ‘radius.’ Next, a brief summary of the computed results that are obtained from different modules used in the framework is listed below.

The list of values for parameter ‘radius’ which is computed from modified ABC algorithm is 0.3529, 0.5518, 0.7513, 1.1509. This list is obtained by changing the parameters upper bound and lower bound which are given as input parameters to the modified ABC algorithm. The corresponding values of the parameter ‘step’ which is computed from Eq. (15) are listed next: 0.1782, 0.2787, 0.3795, and 0.5813. The values of the parameter ‘cost’ which is computed using modified SA algorithm are given next: 809.2268, 789.3740, 766.6992, 731.0053.

The output results obtained for the values of parameter ‘radius’ and the corresponding computed values of the ‘step’ are shown in Table 1. The diagram which depicts these results is given in Fig. 2. The output results obtained from the values of parameter ‘step’ or ‘modified step’ and the corresponding computed values of the parameter ‘cost’ are shown in Table 2. The diagram which depicts these results is given in Fig. 3. The corresponding values of the parameter ‘step’ which is computed from modified RBFN are listed next: 0.6330, 0.6354, 0.6985, 2.0645. The values of the parameter ‘cost’ which is computed using modified SA algorithm are given next: 721.2436, 720.7920, 708.9675. The output results obtained for the values of parameter ‘radius’ and the corresponding computed values of the ‘step’ are shown in Table 3. The diagram which depicts these results is given in Fig. 4. The output results obtained for the values of parameter ‘step’ or ‘modified step’ and the corresponding computed values of the parameter ‘cost’ are shown in Table 4. The diagram which depicts these results is given in Fig. 5.

**Table 1** Computed values of radius and step using ABC algorithm

S. no	Radius	Step
1.	0.3529	0.1782
2.	0.5518	0.2787
3.	0.7513	0.3795
4.	1.1509	0.5813



**Fig. 2** Output of radius and step using ABC algorithm

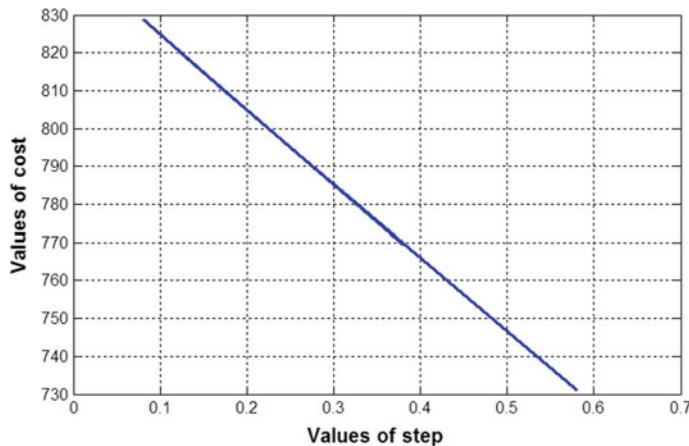
**Table 2** Output values of cost computed using modified SA algorithm based on step/modified step found using ABC

S. no	Step/modified step	P	Cost $\times 10^{-5}$
1.	0.5813	43.0244	0.007310053
2.	0.3795	45.4459	0.007696992
3.	0.2787	46.6555	0.007893740
4.	0.2614	46.8632	0.007927755
5.	0.1782	47.8616	0.008092268
6.	0.1706	47.9527	0.008107374
7.	0.1253	48.4963	0.008197697
8.	0.0802	49.0375	0.008288094

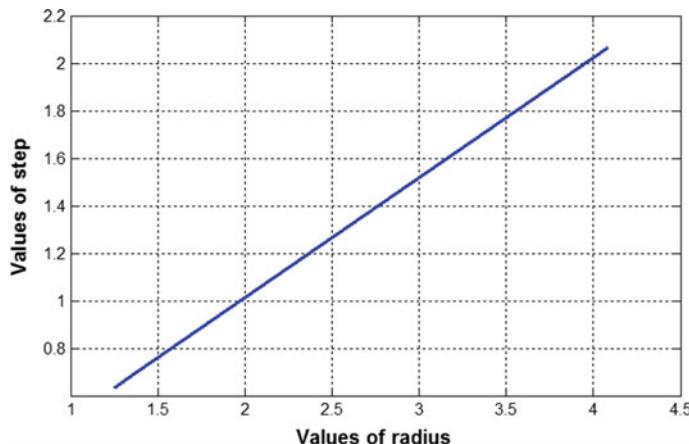
**Table 3** Computed values of radius and step using RBFN

S. no	Radius	Step
1.	1.2533	0.6330
2.	1.2580	0.6354
3.	1.3830	0.6985
4.	4.0876	2.0645

The value of step is also modified by the factor computed from ANN structure. As shown above, either modified ABC or modified RBFN may be engaged for computing the values of the parameter ‘radius’ and both the approaches generate sample cases of values of parameters, and it is found that with the decrease in value of ‘step,’ the cost calculated from modified SA algorithm is increasing.



**Fig. 3** Output of step and cost using improved ABC algorithm and modified SA algorithm



**Fig. 4** Output of radius and step using RBFN

Where PPI is representing the particular portfolio ( $i = 1-10$ ). The value (–) is representing zero, and Bi represents the name of a particular company ( $i = 1-10$ ).

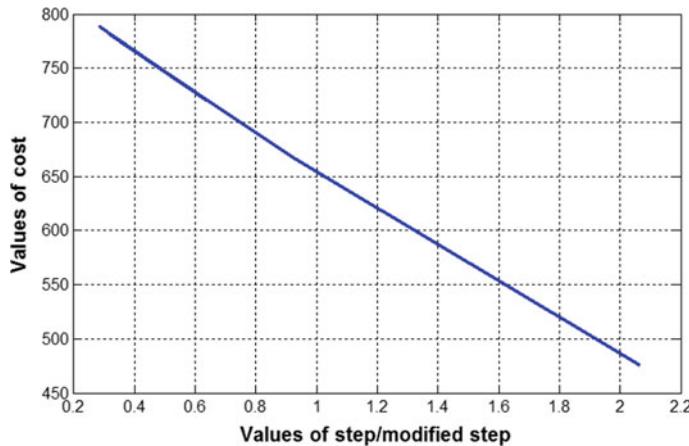
New values of expected returns are computed using the following equations:

$$\text{New Expected Return} = \text{Expected Return} + z_1 \quad (28)$$

where  $z_1$  is augmented value of expected return which represents the risk due to nascent parameters used in the proposed model.

**Table 4** Output values of cost computed using modified SA algorithm based on step/modified step found using RBFN

S. no	Step/modified step	P	Cost $\times 10^{-5}$
1.	2.0645	25.2260	0.004754195
2.	0.9284	38.8592	0.006666502
3.	0.6985	41.6179	0.007089645
4.	0.6354	42.3752	0.007207920
5.	0.6330	42.4040	0.007212436
6.	0.3141	46.1828	0.007816549
7.	0.2857	46.5715	0.007880000
8.	0.2847	46.5836	0.007881968

**Fig. 5** Output of step/modified step and cost using improved RBFN and modified SA algorithm

The expected returns are computed using output values of the parameter cost which is calculated using modified SA algorithm, and the expected returns are scaled by multiplying this cost with a scale factor of  $1 \times 10^{-5}$ . The situation considered in these computations is based upon the risk-averse tendency of the investor. Summary results for portfolio selection using the existing mean-variance model are given in Table 5, whereas the results obtained after selecting newly expected returns using the proposed model are given in Tables 6 and 7. In the proposed model, the values of the expected returns are selected by including the value of the risk in the form of augmented cost  $z_1$ . The value of this augmented cost is found using modified SA algorithm based on parameters computed using modified ABC algorithm and modified RBFN approach. A nascent model is incorporated in the proposed model which computes the value

**Table 5** Output computations for existing basic mean–variance model

	Returns (r0)	Allocation					Portfolio risk
		B1	B2	B3	B4	B5	
PP1	0.2572	—	—	—	—	0.5630	0.1622
PP2	0.2775	—	—	—	—	0.5005	0.1641
PP3	0.2979	—	—	—	—	0.4379	0.1697
PP4	0.3183	—	—	—	—	0.3754	0.1787
PP5	0.3387	—	—	—	—	0.3128	0.1905
PP6	0.3590	—	—	—	—	0.2502	0.2047
PP7	0.3794	—	—	—	—	0.1554	0.2207
PP8	0.3998	—	—	—	—	0.0485	0.2376
PP9	0.4202	—	—	—	—	0.0	0.2557
PP10	0.4405	—	—	—	—	0.0	0.2773
	Returns (r0)	Allocation					Portfolio risk
		B6	B7	B8	B9	B10	
—	—	—	—	—	—	0.4370	
—	—	—	—	—	—	0.4495	
—	—	—	—	—	—	0.5621	
—	—	—	—	—	—	0.6246	
—	—	—	—	—	—	0.6872	
—	—	—	—	—	—	0.7498	
—	—	—	0.0580	—	—	0.7866	
—	—	—	0.1376	—	—	0.8139	
—	—	—	0.1124	—	—	0.8876	
—	—	—	—	—	—	1.0	

of this augmented cost using modified SA algorithm. A risk-averse investor has an inbuilt tool to see the impact of risk inclusion in the model and select an appropriate choice from the output table for the investment (Figs. 6 and 7).

## 4 Conclusion

In this paper, we interpreted the optimization problem of portfolio rooted into two nascent basal parameters, viz.  $\alpha_{new}$ ,  $\beta_{new}$ . The establishment of a different texture of optimization problem is offered that is effectual in computing the optimal cost using quadratic equations involving these basal parameters. An elucidation of modified simulated annealing algorithm is provided, which has the capability to complete

**Table 6** Summary results of portfolio selection based on modified values of expected returns using parameters of computed using ABC (step = 0.1782, P = 47.8616, z<sub>1</sub> = 0.008092268)

	Returns (r <sub>0</sub> )	Allocation					Portfolio risk
		B1	B2	B3	B4	B5	
PP1	0.2607	0.0037	–	–	–	0.5602	0.1622
PP2	0.2816	0.0059	–	–	–	0.4398	0.1641
PP3	0.3024	0.0081	–	–	–	0.4315	0.1697
PP4	0.3233	0.0104	–	–	–	0.3672	0.1787
PP5	0.3442	0.0126	–	–	–	0.3028	0.1905
PP6	0.3651	0.0148	–	–	–	0.2385	0.2047
PP7	0.3860	0.0107	–	–	–	0.1341	0.2205
PP8	0.4069	0.0066	–	–	–	0.0293	0.2371
PP9	0.4277	0.0	–	–	–	0.0	0.2554
PP10	0.4486	0.0	–	–	–	0.0	0.2774
	Returns (r <sub>0</sub> )	Allocation					Portfolio risk
		B6	B7	B8	B9	B10	
–	–	–	–	–	–	0.4361	
–	–	–	–	–	–	0.4982	
–	–	–	–	–	–	0.5604	
–	–	–	–	–	–	0.6225	
–	–	–	–	–	–	0.6846	
–	–	–	–	–	–	0.7467	
–	–	–	0.0830	–	–	0.7722	
–	–	–	0.1668	–	–	0.7973	
–	–	–	0.1152	–	–	0.8848	
–	–	–	–	–	–	1.0	

optimal values of the cost based on a significant parameter, viz. modified step. The modified algorithm is apt for computing global solution for this optimization problem. The significant factor is based on another parameter, viz. radius. Two choices for computing the value of parameter radius are explained, which is based on ABC algorithm or by applying RBFN. ABC algorithm uses two significant parameters which are used for binding maximum and minimum values for computing the value of radius. RBFN uses three different functions, and the value of radius is computed from the maximum and minimum value of points in these functions. Ultimately, the computed value of step is modified by multiplying it by a factor computed from ANN structure. Determinately, the modified SA algorithm is applied, such that an optimal

**Table 7** Summary results of portfolio selection based on modified values of expected returns using parameters of computed using RBFN ( $\text{step} = 0.6330$ ,  $P = 42.4040$ ,  $z_1 = 0.007212436$ )

	Returns (r0)	Allocation					Portfolio risk
		B1	B2	B3	B4	B5	
PP1	0.2644	0.0034	–	–	–	0.5603	0.1623
PP2	0.2848	0.0040	–	–	–	0.4972	0.1642
PP3	0.3051	0.0047	–	–	–	0.4341	0.1699
PP4	0.3255	0.0054	–	–	–	0.3710	0.1788
PP5	0.3459	0.0060	–	–	–	0.3079	0.1906
PP6	0.3663	0.0067	–	–	–	0.2448	0.2049
PP7	0.3866	0.0034	–	–	–	0.1534	0.2208
PP8	0.4070	–	–	–	–	0.0486	0.2377
PP9	0.4274	–	–	–	–	–	0.2599
PP10	0.4477	–	–	–	–	–	0.2774
	Returns (r0)	Allocation					Portfolio risk
		B6	B7	B8	B9	B10	
	–	–	–	–	–	0.4364	
	–	–	–	–	–	0.4988	
	–	–	–	–	–	0.5612	
	–	–	–	–	–	0.6237	
	–	–	–	–	–	0.6811	
	–	–	–	–	–	0.7485	
	–	–	0.0566	–	–	0.7866	
	–	–	0.1375	–	–	0.8139	
	–	–	0.1124	–	–	0.8876	
	–	–	–	–	–	1.0	

value of the cost, as well as the optimal value of the basal parameter, may be obtained using this modified value of step. After analyzing sample cases of the output, it is discovered that the optimal cost computed from modified SA algorithm has a rising pattern when the value of the modified step is lesser. The delineation of the modified SA which is offered in the paper proves to be expedient for making an appropriate choice for the value of modified step and, in essence, selecting an optimal cost based upon basal parameters.

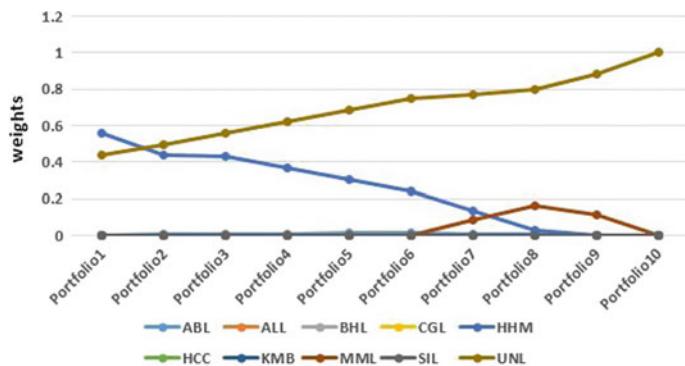


Fig. 6 Graph for output results of portfolio selection based on SA and ABC

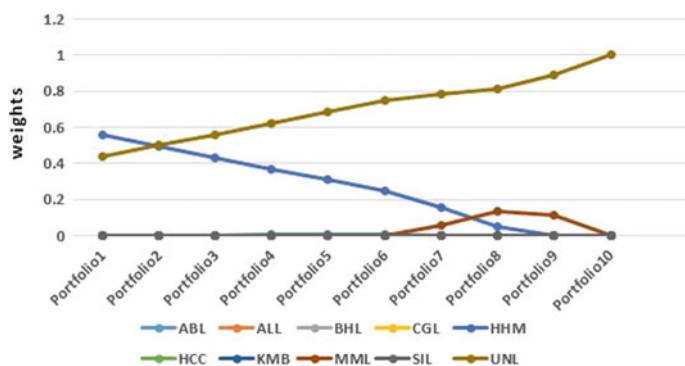


Fig. 7 Graph for output results of portfolio selection based on modified SA and RBFN

## References

- Markowitz, H. (1952). Portfolio selection. *Journal of Finance*, 7(1), 77–91.
- Crama, Y., & Schyns, M. (2003). Simulated annealing for complex portfolio selection problems. *European Journal of Operational Research*, 150(3), 546–571.
- Gupta, P., Mehlawat, M. K., Inuiguchi, M., & Chandra, S. (2014). Fuzzy portfolio optimization: Advances in hybrid multi-criteria methodologies (Vol. 316). Springer.
- Chen, W. (2015). Artificial bee colony algorithm for constrained possibilistic portfolio optimization problem. *Physica A: Statistical Mechanics and its Applications*, 429, 125–139.
- Shen, W., Guo, X., Wu, C., & Wu, D. (2011). Forecasting stock indices using radial basis function neural networks optimized by artificial fish swarm algorithm. *Knowledge-Based Systems*, 24(3), 378–385.
- Kumar, D., & Mishra, K. K. (2017). Portfolio optimization using novel co-variance guided artificial bee colony algorithm. *Swarm and Evolutionary Computation*, 33, 119–130.
- Luo, Y., Zhu, B., & Tang, Y. (2014). Simulated annealing algorithm for optimal capital growth. *Physica A: Statistical Mechanics and its Applications*, 408, 10–18.
- Yu, L., Wang, S., & Lai, K. K. (2008). Neural network-based mean–variance–skewness model for portfolio selection. *Computers & Operations Research*, 35(1), 34–46.

9. Yevseyeva, I., Guerreiro, A. P., Emmerich, M. T., & Fonseca, C. M. (2014, September). A portfolio optimization approach to selection in multiobjective evolutionary algorithms. In *International Conference on Parallel Problem Solving from Nature* (pp. 672–681). Springer, Cham.
10. Gan, G., & Huang, J. X. (2017, August). A data mining framework for valuing large portfolios of variable annuities. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 1467–1475). ACM.
11. Zhao, H., Liu, Q., Wang, G., Ge, Y., & Chen, E. (2016, August). Portfolio selections in P2P lending: A multi-objective perspective. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 2075–2084). ACM.
12. Das, P., Banerjee, A. (2011, August). Meta optimization and its application to portfolio selection. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 1163–1171). ACM.
13. Chen, J. S., & Hou, J. L. (2006). A combination genetic algorithm with applications on portfolio optimization. *Lecture Notes in Computer Science*, 4031, 197.
14. Meghwani, S. S., Thakur, M. (2017). Multi-criteria algorithms for portfolio optimization under practical constraints. *Swarm and Evolutionary Computation*.
15. Qin, Z., Kar, S., & Zheng, H. (2016). Uncertain portfolio adjusting model using semiabsolute deviation. *Soft Computing*, 20(2), 717–725.
16. Sivanandam, S. N., Deepa, S. N. (2006). Introduction to neural networks using Matlab 6.0. Tata McGraw-Hill Education.
17. Calvet, L., Kizys, R., Juan, A. A., & De Armas, J. (2016). A SimILS-based methodology for a portfolio optimization problem with stochastic returns. In *Modeling and Simulation in Engineering, Economics and Management* (pp. 3–11). Springer International Publishing.
18. Better, M., Glover, F., Kochenberger, G., & Wang, H. (2008). Simulation optimization: Applications in risk management. *International Journal of Information Technology & Decision Making*, 7(04), 571–587.
19. Qi, R., & Yen, G. G. (2017). Hybrid bi-objective portfolio optimization with pre-selection strategy. *Information Sciences*, 417, 401–419.
20. Qiu, H., Han, F., Liu, H., & Caffo, B. (2015). Robust portfolio optimization. In *Advances in Neural Information Processing Systems* (pp. 46–54).
21. Kothari, D. P., Dhillon, J. S. (2006). Power system optimization (pp. 131–244, 321–386). New Delhi: Prentice-Hall.

# A Proposed Method for Disruption Classification in Tokamak Using Convolutional Neural Network



Priyanka Sharma, Swati Jain, Vaibhav Jain, Sutapa Ranjan, R. Manchanda, Daniel Raju, J. Ghosh and R. L. Tanna

## 1 Introduction

Energy released during a nuclear fusion is one of the major alternative sources of electrical power. Fusion reactors or also known as fusion power plant or thermonuclear reactors are the place where the process of nuclear fusion takes place.

To accomplish high enough fusion reaction rates and to utilize fusion effectively as an energy source, fuel sources deuterium and tritium, which are isotopes of hydrogen, are heated at a very high temperature of more than 100 million degrees Celsius. At such extremely high temperatures, the fuel is converted to plasma. This especially hot

---

P. Sharma (✉) · S. Jain · V. Jain

Department of Computer Engineering, Institute of Technology,  
Nirma University, Ahmedabad, India  
e-mail: priyanka.sharma@nirmauni.ac.in

S. Jain

e-mail: swati.jain@nirmauni.ac.in

V. Jain

e-mail: vaibhav.jain@nirmauni.ac.in

S. Ranjan · R. Manchanda · D. Raju · J. Ghosh · R. L. Tanna

Institute of Plasma Research, Bhat, Gandhinagar, India  
e-mail: sranjan@ipr.res.in

R. Manchanda

e-mail: mranjana@ipr.res.in

D. Raju

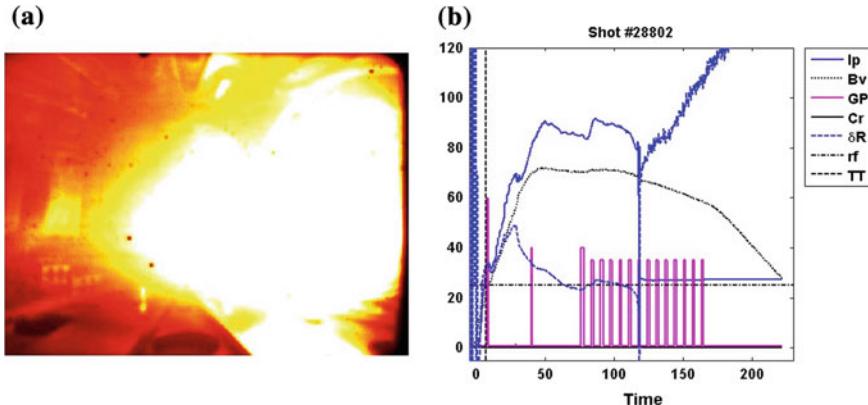
e-mail: raju@ipr.res.in

J. Ghosh

e-mail: jghosh@ipr.res.in

R. L. Tanna

e-mail: rakesh@ipr.res.in



**Fig. 1** A pictorial view of disruptions in Aditya tokamak during a shot 28802

plasma is also enormously thin and breakable, which possesses a density of million times less than air. To shield the plasma from being defective and cooled by touch with material surfaces, it is enclosed in a magnetic confinement system [10, 11].

One of the widely used magnetic confinement systems is called tokamak which stands as the fundamental concept of all fusion reactors. Tokamak is composed of magnets that yield a toroidal field and poloidal field. The toroidal field has a torus shape that encompasses the plasma, and the poloidal field moves in form of circles around the plasma. The outcome is a magnetic field that has a comparative shape to the toroidal plasma; it is attempting to restrict and encompasses it on all sides, accordingly catching it.

Tokamak's functionality is driven by certain parameters defining the plasma, such as density, beta, and plasma current. Disruption in a tokamak can be defined as a major instability that takes place abruptly in case when any of these parameter values cross the allowable limit. It includes an immediate loss of confinement which may be a reason of the sudden termination of discharges in tokamak [3].

Due to disruptions, the plasma energy gets transferred to the surrounding structures which causes to massive heat and serious damage. It affects the integrity of a tokamak. Hence, to protect tokamak functionality from disruption, the physicist must operate the tokamak with the parameters within safe limits [4]. Prediction and classification of disruption are a highly complicated job. The plasma pulse shown in Fig. 1 is a typical example of how disruptions occur in Aditya tokamak. This corresponds to a plasma current,  $Ip = 90$  KA (Fig. 1b), which disrupts at 120 ms.

The primary goal of the proposed method is to build a deep convolutional neural network (CNN)-based system which learns from existing signal patterns corresponding to both good and disruptive shots and uses that knowledge, to quickly classify shots into different categories, when the tokomak is in operation. Although, at present, the proposed work does not stand for real-time response, the system developed would be responsive enough to classify within a very reasonable amount of time, so that

the session leaders can base their action on their result. The second objective is to associate good and disruptive discharge scenarios to operational parameters prevalent at the system, so that the session leader can decide on operational parameters in advance for upcoming operations incorporating the applicable criteria that follow.

## ***1.1 Motivation***

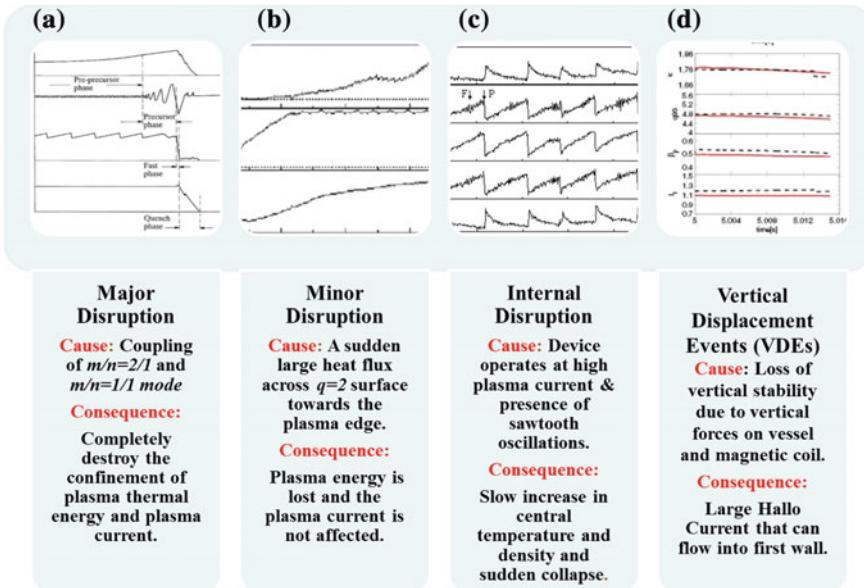
Several attempts for prediction of disruptions are made using different machine learning algorithms like SVM, ANN. One common thing in all these is that the input is the handcrafted characteristic of the diagnostics. These handcrafted characteristics can be the mean value in a given time slice, the max value, the min value, or any well-defined single value in the sliced instance [11]. These crafted features lead to information loss if they do not represent the data appropriately. The deep learning approach in contrast to this adds an additional layer which is responsible for feeding in semantically rich features to the standard ANN. This added layer can learn to extract the desired features from the diagnostics that can lead to the meaningful and accurate classification.

In the most recent decade, different machine learning methods, primarily artificial neural networks and support vector machines (SVMs), are the top trending ways to deal with disruption prediction. These computational frameworks perform ‘learning’ from the data during ‘training phase.’ When the frameworks got trained, they can be used to identify the specific behavior they were intended to distinguish. During the study of previous work, it has been also found that the acquired outcomes have not been totally acceptable [10, 11]. One of the admissible lacks observed in most of the past research work is that only specific parts of the shots have been taken into consideration. The overall growth of the discharges has not been investigated in its full length. Most of the models proposed and developed in this manner were not perfectly suitable for real-time situations where the requirement of predicting disruptions reaches out to the entire development of the pulse. They can be only able to analyze the physics of the phenomenon.

There are two basic aims behind this investigation: firstly to study various existing disruptions classification techniques and secondly to avoid the extraction of features in manual manner; the deep CNN model is implemented for automatic feature learning.

## ***1.2 Stages and Types of Disruption***

Usually, a disruption event has three stages: thermal quench, current quench, and loss of vertical position. The first stage is the result of loss of confinement due to some event related to magnetohydrodynamic (MHD) instabilities [20].



**Fig. 2** Types of disruptions, causes, and consequences [8, 9, 14, 19]

At this stage, a major amount of stored energy and some amount of plasma current is lost. Further, the rest of the thermal energy gets lost in a rapid manner on a time scale of less than one millisecond (Fig. 2a). Therefore, this stage is known as thermal quench. During thermal quench, the thermal energy gathered at plasma component may be very intense and can be the cause of melting and vaporization of component surface. The rest of the plasma is very resistive in nature, due to which the deterioration of toroidal current takes place, and thus, the current quench stage is reached.

There are several categories of disruptions which have been explained in the various available literature. Major disruptions (Fig. 2a) can be considered as the most hazardous instabilities in tokamak. They lead to complete loss of confinement of plasma and cause the termination of the discharges. Many descriptions have been introduced to explain the phenomenon of major disruptions; most of them including  $m/n=2/1$  (where  $m$  and  $n$  are the poloidal and toroidal Fourier mode numbers, respectively) amplitude increase as the main factor for major disruptions [4].

Minor disruptions (Fig. 2b) are defined by a rapid huge heat flux across the  $q=2$  surface against the plasma edge. These kinds of disruptions are called minor just because of the fact that only a portion of the plasma energy is lost without affecting the plasma current. During minor disruptions, the plasma can recover. In case, when no action is taken to control the evolution of the island at  $q=2$ , it ends the plasma in terms of a major disruption [20].

Internal disruption (sawtooth oscillations) is also disruptive instability examined in several tokamaks. This process comprises of a moderate increment in the central temperature and density followed by an immediate collapse. The existence of these kinds of event (Fig. 2c) was noticed in Aditya tokamak for the first time in December 1995, at the stage with very high plasma current. After this, the sawtooth oscillations or so-called internal disruptions have been detected in several experimental studies and under different physical conditions [4].

Another kind of disruptions which take place in tokamak like JET is called vertical displacement event (VDE). VDEs (Fig. 2d) result from a loss of vertical stability and can cause large halo currents that can flow into the first wall. In VDE, the current and thermal energy is not released until the plasma becomes limited. VDEs can cause more damage [9] but are easier to predict.

Few more types of instabilities called mode lock (ML) instabilities, H-mode, L-mode (HL) also cause the major disruptions. Radiated power disruptions and unclassified disruptions category also exists. H-mode and L-mode represent the high confinement state and low confinement state, respectively. A transition from L-mode to H-mode takes place when a threshold level of heating of plasma is crossed. Radiated power disruptions take place during thermal quench stage of density limit disruption. It is very important to classify the correct category of disruption to ensure the reliability of tokamak operation [3, 12].

In Sect. 2, the brief introduction and characteristics of Aditya tokamak have been mentioned. In Sect. 3, a brief view over the existing techniques for disruption prediction and classification has been discussed. Section 4 consists in the concept behind deep convolutional neural network. Section 5 includes the proposed method to predict the disruption using CNN model. Section 6 concludes the paper, and Sect. 6 presents the various references which have been considered during the study.

## 2 Aditya Tokamak and It's Diagnostics

Aditya is a medium-sized tokamak which exists at the Institute for Plasma Research (IPR) in India for over a decade. Some of the important aspects of Aditya include its major radius which is 0.75 m and minor radius which is 0.25 m. In extreme case, a toroidal magnetic field 1.2 T is produced by using 20 toroidal field coils distributed uniformly in the toroidal way. A transformer converter control framework is used in Aditya during operation. Pulses longer than 100 ms with 80–110 kA plasma current at toroidal field of about 0.9 T is being frequently generated for several experiments [18].

Various diagnostics used in Aditya include Mirnov oscillations, loop voltage, radiation from OI (singly charged oxygen) impurities, soft X-ray (SXR) monitor,  $H\alpha$  monitor, plasma current, radial plasma position LX, VUV radiation (spectroscopy vacuum ultraviolet). Aditya has been upgraded to include the diverters. Various works based on neural networks to predict the disruptions in Aditya tokamak have been proposed in past.

### 3 Disruption Classification and Prediction Techniques

Various phenomenal research works have been done to predict the disruptions in different reactors like ITER, JET. Some of disruptions predictors have been specifically designed for Aditya tokamak. Various ANN techniques like SVM have been introduced in disruption prediction task. This section presents a detailed overview of the previous works.

Murari et al. [15] have presented the various results obtained from disruption prediction methods developed for JET. The major aim of their research was to examine that how it is feasible to perform satisfactory prediction based on raw information prior to the occurrence of disruption. They have used supervised learning-based classification and regression trees (CART). Several unsupervised learning methods, primarily K-means and hierarchical, were further examined. Rattá et al. [12] proposed a real-time disruption prediction mechanism which was implemented for JET specifically. They have used support vector machines. The primary objective behind the investigation was to achieve a high prediction rate in a real-time simulated environment. In [3], Matteo Cacciola et al. proposed a multi-class support vector machine-based model which aimed to achieve the knowledge which led to predict the disruption. The discharges analyzed for this work have been taken from database of disruptive discharges in JET team. An automatic classification technique of disruption was proposed by Murari et al. [2]. It relies on clustering using the geodesic distance on Gaussian manifolds. The proposed work was implemented for disruption classification at JET. The proposed approach ensures that the error bars of the measurements and has achieved more effective Euclidean distance based classifier which is better than other traditional classification methods. Dormido-Canto et al. [6] described the advanced predictor of disruptions (APODIS) architecture. APODIS was implemented as a disruption predictor real-time environment operation in JET. Equated and unequaled datasets have been considered to achieve real-time predictors from scratch. The discharges have been taken in chronological pattern. In [5], P.C. de Vries et al. presented a survey conducted to identify the reasons of all 2309 disruptions of JET operations for last few decades. The main objective of this survey was to get a detailed scenario of all probable disruption reasons, to discover improved approaches, and to avoid or reduce their impact.

A white paper by William Tang [7] has presented with their idea that disruption prediction may be an innovative research aim in order to investigate that whether there is an existence of more large data dependent, supervised machine learning methodologies are available or not. Sengupta et al. [11] tried to predict the disruption boundaries for a class of disruptions called density limit disruption with the help of neural network. Various diagnostics used in Aditya tokamak have been considered at given time intervals. They have been taken as the input to the neural network, to find out, at each of these time intervals, the density boundary.

Sengupta and Ranjan [10] proposed a neural network approach to predict the disruptions during operations in Aditya tokamak. A time series prediction method

was deployed in which a series of past values of sometime dependent quantity is considered to predict its value in upcoming time (Tables 1 and 2).

In [4], the observations about major disruptions and sawtooth oscillations (internal disruptions) have been given by Asim Kumar Chattopadhyay et al. These disruptions are commonly detected in massively heated Aditya tokamak discharges. The disruptions have been studied using soft X-ray (SXR) tomography in addition of other different diagnostics. In [13], F. Salzedas has tried to solve some queries raised by various observations, like why the loss of energy confinement takes place due to only large magnetic island in periodic and asymmetric manner. Another cause for the minor disruption has been given in form of a secondary instability (SI) to the magnetic island. Zakharov et al. [20] have introduced a new form of instability mode in plasma in form of 1/1 mode. They have analyzed the VDE in JET. The solution to control the disruptions in plasma can be achieved by simplifying the plasma regime rather than sacrificing the performance of plasma.

## 4 Deep Learning Techniques and CNN

Deep learning using convolutional neural network (CNN) is a set of algorithms that aids in machine learning by extracting high-level features from multidimensional data. A convolutional neural network can be defined as a feed-forward artificial neural network. In CNN, the neurons are organized in such a manner that they respond to overlapping regions tiling the image or signal. As compared to ANN-based implementations, it gives better results, since the layers have neurons arranged in three dimensions. The technique has been found to be highly effective in image classification and speech recognition.

There are three main layers in CNN:

- Convolutional layer: This layer convolves two signals; here, it convolves the input image and the predefined kernels. The output maps obtained from convolution act as different features of the images.
- Pooling layer/subsampling layer: To make the representation more manageable and small without losing much of information is done in this layer. Mostly, down-sampling is done to pool the representations.
- Fully connected layer: This layer gets the input data which is ready for classification. This layer classifies and gives the results accordingly. Here, neurons have full connections to all the activations.

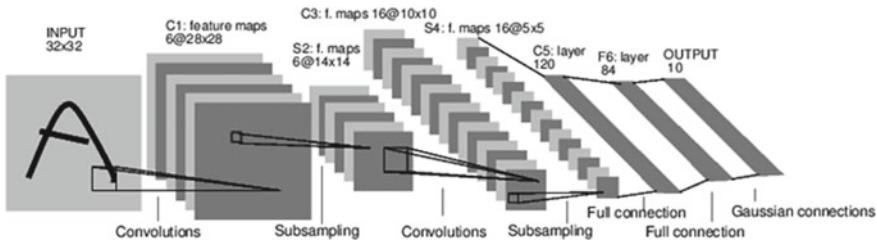
As shown in the Fig. 3, any convolution neural network is designed by layering convolution and subsampling/pooling layer. The designing parameter includes number of layers, size of kernels in convolution layers and defining the connection map to link the output generated by one layer to the next layer [16]. The computational requirement for CNN is extremely large and needs the computing power multi-core architectures of GPU.

**Table 1** A brief view on various existing approaches to classify/prediction of disruption

Source	Objective	Work done	Relevant findings
Murari et al. 2009 [15]	To perform satisfactory prediction based on raw information prior to the occurrence of disruption	<ul style="list-style-type: none"> <li>A supervised learning-based classification and regression trees (CART) implemented for disruption classification</li> <li>Several other unsupervised methods, primarily K-means and hierarchical, have also been examined</li> </ul>	<ul style="list-style-type: none"> <li>Disruption prediction can be performed in such a way that a success rate of 80% can be obtained not earlier than 180 ms before the disruption</li> <li>As a future work, various other independent methods can be used to receive the prediction at the earliest</li> </ul>
Rattá et al. 2010 [12]	To achieve a high prediction rate in a real-time simulated environment	<ul style="list-style-type: none"> <li>Support vector machine was implemented to predict the disruption in real-time environment</li> </ul>	<ul style="list-style-type: none"> <li>Testing using considerable number of pulses from the more recent campaign performed</li> <li>In the future, prediction of disruption after an alarm is being triggered</li> </ul>
Cacciola et al. 2006 [3]	To implement a multi-class SVM-based model which led to predict the disruption	<ul style="list-style-type: none"> <li>A database of disruptive discharges in JET team has been analyzed in this work as a relevant experimental example</li> </ul>	<ul style="list-style-type: none"> <li>As a future development, the authors have mentioned that broader time samples should be examined to expand the pattern length</li> </ul>
Dormido-Canto et al. 2013 [6]	To get all probable disruption reasons, to discover improved approaches	<ul style="list-style-type: none"> <li>The work attempts to provide better and effective ways to avoid or prevent JET disruptions</li> </ul>	There is a remark about the results that they have been achieved from the models which have been trained with no more than 42 disruptive discharges
Sengupta et al. 2001 [11]	To predict density limit disruption with the help of neural network	<ul style="list-style-type: none"> <li>Various diagnostics from the Aditya tokamak have been considered to find out the density boundary at specific time intervals.</li> <li>Implemented the real-time disruption alarm</li> </ul>	<ul style="list-style-type: none"> <li>The input is the handcrafted characteristic of the input diagnostics. These handcrafted characteristics can be the mean value in a given time slice, the max value, the min value, or any well-defined single value in the sliced instance</li> </ul>
Ranjan et al. 2000 [10]	Forecasting of disruptions in Aditya using neural network	<p>A time series prediction method was deployed in which a series of previously evaluated variables is taken into consideration to predict its value in upcoming time</p>	<p>They have used four Mirnov probes, one soft X-ray monitor, and one H<math>\alpha</math> monitor. All were observables with varying time instants. The disruption event has been predicted in advance</p>

**Table 2** A brief view on various existing approaches for analysis of disruptions

Source	Objective	Work done	Relevant findings
Chattopadhyay et al. 2006 [4]	To survey various types of disruptions that takes place in Aditya tokamak specifically	<ul style="list-style-type: none"> <li>Major disruptions and saw teeth oscillations are commonly detected in massively heated Aditya tokamak discharges</li> <li>Their properties have been studied using soft X-ray (SXR) tomography in addition to other different diagnostics</li> </ul>	<ul style="list-style-type: none"> <li>To analyze the phenomenon of sawtooth internal disruptions, SXR tomography is implemented using single array of detectors with an assumption of rigid rotation of the modes. Sawtooth periods have been evaluated and matched with the scaling laws and found to be in better accord</li> </ul>
Bondson 1995 [1]	To discuss major and minor disruptions in tokamaks	<ul style="list-style-type: none"> <li>Many models and numerical simulations of disruptions based on resistive MHD are reviewed</li> <li>A discussion is given of how disruptive current profiles are correlated with the experimentally known operational limits in density and current</li> </ul>	<ul style="list-style-type: none"> <li>Observations indicate that major disruptions usually occur in at least two phases, first a ‘predisruption’, or loss of confinement in the region <math>1 &lt; q &lt; 2</math>, leaving the <math>q * 1</math> region almost unaffected, followed by a final disruption of the central part, interpreted here as a toroidal <math>n = 1</math> external kink mode</li> </ul>
de Vries 2011 [5]	To understand the statistics which cause the disruptions in the operational space	<ul style="list-style-type: none"> <li>A fundamental survey performed on the precursors or triggers developed at JET protection system. Also, the strength of this method with respect to the mitigation of forces and heat loads was examined</li> </ul>	<ul style="list-style-type: none"> <li>The paper does not provide detailed explanation on the reason of disruptions or the existing physics of operational boundaries, but only gives an idea of statistics which causes the disruptions in the operational space</li> </ul>
Gerhardt et al. 2009 [7]	<ul style="list-style-type: none"> <li>To study the characteristics of the current quench stage of plasma during disruptions in NSTX</li> </ul>	<ul style="list-style-type: none"> <li>A complete observation behind the stage of current quench of the plasma during disruption has been performed for National Spherical Torus Experiment</li> </ul>	<ul style="list-style-type: none"> <li>One important observation which has been made during this study is that the plasma current before the disruption is often substantially less than the flat top value</li> </ul>
Zakharov et al. 2012 [20]	<ul style="list-style-type: none"> <li>To understand the overall progress in the field of disruption prediction.</li> </ul>	<ul style="list-style-type: none"> <li>A new form of instability mode has been introduced in the form of 1/1 mode</li> <li>VDE instabilities in the JET have been analyzed</li> </ul>	<ul style="list-style-type: none"> <li>The solution to control the disruptions in plasma can be achieved by simplifying the plasma regime rather than sacrificing the performance of plasma</li> </ul>



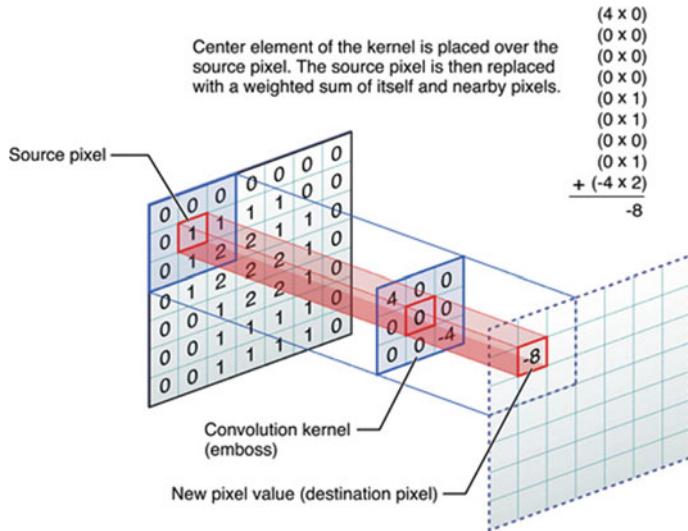
**Fig. 3** A full convolutional neural network model [16]

The convolutional layer takes an input image with dimension  $m \times m \times r$  where  $m$  is the height and width of the image, while  $r$  can be given as the number of channels; for example, in a RGB image the number of channels is three, i.e.,  $r = 3$ . The first layer of a convolutional neural network is always a convolutional layer. Suppose there is an input image with dimension  $32 \times 32 \times 3$ . The convolutional layer can be defined effectively in a manner that if we consider a  $5 \times 5$  area, which cover the top left part of the image. Now, if we imagine that  $5 \times 5$  area slides over the whole area of the input image that area is known as filter. It is also referred as a kernel and the region, which has been covered by the filter is called receptive field. The filter consists of an array of weights or parameters.

The depth of filter must be the same as the depth of the input. Therefore, the filter possesses dimensions of  $5 \times 5 \times 3$ . As the filter will convolve over the whole input image, the values in the filter will be multiplied by the original pixel values of the image. These multiplications are all summed up and thus produce a single number. It should be noted that this number just indicates that the filter is present at the top left of the image. Later, the same procedure must be followed for each of the places on the input image. In next phase, the filter gets shifted to right by one unit, then right again by 1, and so on. There is number produced at every unique location of input volume processed. A number is produced by each of the unique locations on the input volume.

Once the filter slides over the whole locations of the input image, it results in a  $28 \times 28 \times 1$  array of numbers, which is known as activation map or feature map. The logic to obtain a  $28 \times 28$  array is that there are 784 different locations where a  $5 \times 5$  filter can fit on a  $32 \times 32$  input image. These 784 numbers are mapped to a  $28 \times 28$  array. Therefore, it can be observed that the size of the filters encourages the locally connected structure which are each convolved with the image to produce  $k$  number of feature maps of size  $m - n + 1$ .

After applying a set of filters on top during the second convolutional layer operation, the resultant will be in form of activations which will represent more higher-level features. These features may include semicircles (combination of a curve and straight edge) or squares (combination of several straight edges). In a basic CNN, various other layers are dispersed in between convolutional layers. The objective of these layers is to provide nonlinearities and preservation of dimension which make the network most robust. This concept also helps in controlling the overfitting. A classic



**Fig. 4** Visualization of  $3 \times 3$  filter convolving around an input volume and producing an activation map [17]

CNN architecture is given in Fig. 4, where ReLU and pool layers have been interspersed. At the end of the network, a fully connected layer takes an input volume and outputs an N-dimensional vector where N is the number of classes that the program has to choose from.

## 5 Proposed Work

In a tokamak, both favorable and unfavorable discharges exist (good shots and minor/major disruptive shots). Detecting the onset of minor disruptions and taking proactive measures might prevent major disruptions. Due to the large number of shots and diagnostics generated during a tokamak run, manual detection of the onset of disruptions becomes a challenge. Therefore, the proposed automated system capable of predicting a disruption based on raw data, as well as diagnostic data in a reasonable amount of time, would be an advantage for experimentalists. In addition to this, knowing in advance, the possible operational scenarios that might result in both good and disruptive shots would act as a guideline for tokamak operations. A schematic diagram showing the proposed work is given in this section.

## 5.1 Database Preparation

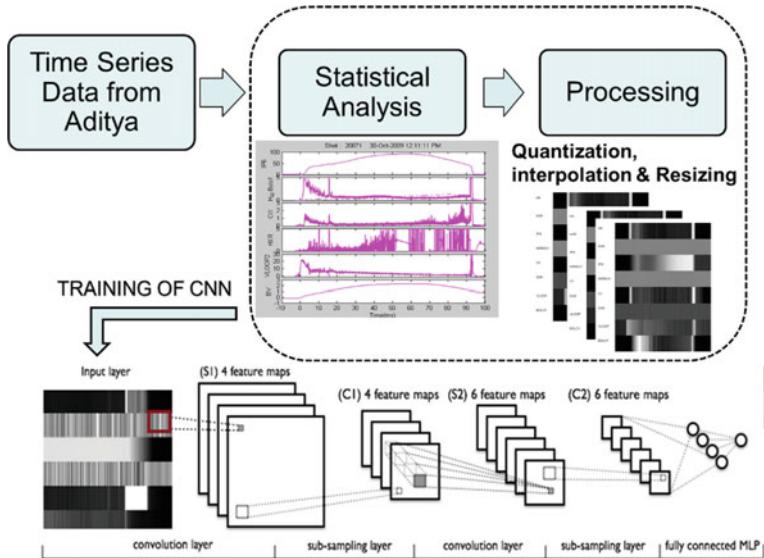
- The database for performing disruption classification has been prepared using more than 100 Aditya discharges in time series form data. These discharges have been labeled into two classes: disruptive and non-disruptive. Because dataset contains an unequal number of data per category, we have to first adjust it, so that the data in the training set and testing set is balanced. The dataset is divided into two parts: One is training, and another is validation data. Data (30%) from each set for the training data and the remaining, 70%, for the validation purpose have been considered. Randomize the split to avoid biasing the results. The training and test sets have been processed by the CNN model.
- The data collection corresponds to various input diagnostics signals like Mirnov oscillations, soft X-ray (SXR) monitor, H $\alpha$  monitor, plasma current, loop voltage, radiation from OI (singly charged oxygen) impurities, radial plasma position LX, VUV radiation (spectroscopy vacuum ultraviolet), line-averaged plasma density, toroidal magnetic field.

## 5.2 Disruption Prediction Using CNN (Proposed Method)

Figure 5 shows a schematic block diagram of the proposed workflow. At very first stage, the extraction of raw data from Aditya tokamak takes place. This time series data will be quantized and converted into images corresponding to given diagnostics.

After preprocessing images, they have been classified into two classes. One is disruptive, and another is non-disruptive. A set of images from both the class has been taken for training of the proposed convolutional neural network, and remaining images have been considered during testing phase. The convolutional neural network model has been implemented using a definite number of convolutional layer followed by ReLu and fully connected layers. After the completion of training, the new instances are used for getting either the class value of the experimental conditions for good shots.

In the results shown in Table 3, a classification of 1000 images which have been generated using the quantization of raw times series data has been performed using CNN. The images have been divided into two classes: disruptive and non-disruptive on observation basis, and an improved accuracy at every iteration for the classification has been achieved after 500 epochs.



**Fig. 5** An overview of the proposed workflow

**Table 3** Result analysis using convolutional neural network (CNN)

Epoch	Iteration	Time elapsed (seconds)	Mini-batch loss	Mini-batch accuracy (%)	Base learning rate
50	50	34.86	2.2775	85.71	0.000100
100	100	65.45	2.2134	86.34	0.000100
150	150	96.25	2.2036	86.56	0.000100
200	200	126.80	1.1824	88.08	0.000100
250	250	157.44	1.1614	89.15	0.000100
300	300	188.29	1.1614	89.15	0.000100
350	350	219.41	1.1329	89.23	0.000100
400	400	250.39	1.1162	90.03	0.000100
450	450	281.29	1.1133	90.26	0.000100
500	500	312.49	1.1026	91.22	0.000100

## 6 Conclusion

Due to disruptions, the plasma energy gets transferred to the surrounding structures which causes to massive heat and serious damage. The physical characterization of disruptions for prediction and control is an extremely complex task. Disruptions cause an extreme interrupt in tokamak operation. Therefore, an advanced predictor is needed to avoid the disruption. In this paper, a review on various kinds of disruptions and their prediction techniques is presented. Along with the review, we proposed a real-time prediction scheme for disruptions specifically for Aditya tokamak. The

proposed prediction model will be implemented using convolutional neural network. The main objective of this investigation is to design an automated system capable of predicting a disruption based on raw data, as well as diagnostic data in a reasonable amount of time which would be an advantage for experimentalists.

## References

1. A Beginner's Guide To Understanding Convolutional Neural Networks, <https://adeshpande3.github.io/>.
2. Bondeson. (1987). *Disruptions in Tokamaks*. Invited Lecture Presented at the Varenna workshop, Theory of Fusion Plasmas.
3. Cacciola, M., Greco, A., Morabito, F. C., & Versaci, M. (2006). Multi class support vector machines for disruption classification in tokamak reactors. *International Journal of Intelligent Technology*, 1(4), 274–280.
4. Chattopadhyay, A. K., Anand, A., Rao, C. V. S., Joisa, S., & Aditya team (2006). Analysis of disruptive instabilities in Aditya tokamak discharges. *Indian Journal of Pure & Applied Physics*, 44, 826–833.
5. de Vries, P. C., Johnson, M. F., Alper, B., Buratti, P., Hender, T. C., Koslowski, H. R., Riccardo, V., & JET-EFDA Contributors. (2011). Survey of disruption causes at JET. *Nuclear Fusion*, 51(5). IAEA, Vienna.
6. Dormido-Canto, S., Vega, J., Ramírez, J. M., Murari, A., Moreno, R., López, J. M., Pereira, A., & JET-EFDA Contributors. (2013). Development of an efficient real-time disruption predictor on JET & implications for ITER. *Nuclear Fusion*, 53(11). IAEA, Vienna.
7. Gerhardt, S. P., Darrow, D. S., Bell, R. E., LeBlanc, B. P., Menard, J. E., Mueller, D., Roquemore, A. L., Sabbagh, S. A., & Yuh, H. (2013). Detection of disruptions in the high- $\beta$  spherical torus NSTX. *Nuclear Fusion*, 53(6). IAEA, Vienna.
8. Institute for Plasma Research (IPR), <http://www.ipr.res.in/>.
9. Liu, Y., Guo, G. C., Ding, X. T., & Wong, K. L. (2002). Effect of Suprathermal electrons on central plasma relaxation oscillations during localized electron cyclotron heating on the HL-1 M tokamak. *Brazilian Journal of Physics*, 32(1). São Paulo.
10. Murari, Boutot, P., Vega, J., Gelfusa, M., Moreno, R., Verdoollaeghe, G., de Vries, P. C., & JET-EFDA Contributors. (2013). Clustering based on the geodesic distance on gaussian manifolds for the automatic classification of disruptions. *Nuclear Fusion*, 53(3). IAEA, Vienna.
11. Murari, Vega, J., Rattá, G. A., Vagliasi, G., Johnson, M. F., & Hong, S. H., & JET-EFDA Contributors. (2009). Unbiased and non-supervised learning methods for disruption prediction at JET. *Nuclear Fusion*, 49(5). IAEA, Vienna.
12. Performing Convolution Operations by Apple Guide, <https://developer.apple.com/library/>.
13. Qiu, Q., Xiao, B., Guo, Y., Liu, L., Xing, Z., & Humphreys, D. A. (2016). Simulation of EAST vertical displacement events by tokamak simulation code. *Nuclear Fusion*, 56(10). IAEA, Vienna.
14. Rattá, G. A., Vega, J., Murari, A., Vagliasi, G., Johnson, M. F., de Vries, P. C., & JET EFDA Contributors. (2010). An advanced disruption predictor for JET tested in a simulated real-time environment. *Nuclear Fusion*, 50(2), 1–10. IAEA, Vienna.
15. Salzedas, F., Meneses, L., deLaLuna, E., Plyusnin, V., Riccardo, V., Jaspers, R., Hender, T. C., Serra, F., & JET EFDA Contributors. (2003). Behavior of density fluctuations and electron temperature profiles in JET density limit disruptions. *EPS Conference on Controlled Fusion and Plasma Physics, EFDA-JET-CP(03)*, 1–54.
16. Sengupta, & Ranjan, P. (2000). Forecasting disruptions in the ADITYA tokamak using neural networks. *Nuclear Fusion*, 40(12), 1993–2008. IAEA, Vienna.
17. Sengupta, & Ranjan, P. (2001). Prediction of density limit disruption boundaries from diagnostic signals using neural networks. *Nuclear Fusion*, 41(5), 487–581. IAEA, Vienna.

18. Tokamak plasma instabilities, <http://slideplayer.com/slide/5273556/>.
19. Yang, Q., Yan, L., & Qian, J. (2002). Studies of mode lock instability in the HL-1 M tokamak. *Brazilian Journal of Physics*, 32(1). São Paulo.
20. Zakharov, L. E., Galkin, S. A., Gerasimov, S. N., & JET-EFDA contributors. (2012). Understanding disruptions in tokamaks. *Additional Information on Physics Plasmas*, 19(055703), 055703-1- 055703-13.

# Comparative Evaluation of Machine Learning Algorithms for Network Intrusion Detection Using Weka



Nureni Ayofe Azeez, Obinna Justin Asuzu, Sanjay Misra,  
Adewole Adewumi, Ravin Ahuja and Rytis Maskeliunas

## 1 Introduction

Even before the advent of mainstream computers, data security has been one of the topics of vital importance. Over the past few years, more computers have been plugged to the Internet, while many are being networked together, thereby increasing the risk of security threats to these systems.

A network intrusion can simply be defined as any activity considered illegal, unauthorized, unapproved, and unethical to the smooth running of computer network

---

N. A. Azeez · O. J. Asuzu

School of Computer Science and Information Systems, North West University, Potchefstroom,  
South Africa

e-mail: nazeez@unilag.edu.ng

O. J. Asuzu

e-mail: bnnsz384@gmail.com

N. A. Azeez · O. J. Asuzu

University of Lagos, Lagos, Nigeria

S. Misra (✉) · A. Adewumi

College of Engineering, Covenant University, Ota 1023, Nigeria

e-mail: sanjay.misra@covenatuniversity.edu.ng; ssopam@gmail.com

A. Adewumi

e-mail: Wole.adewumi@covenatuniversity.edu.ng

R. Ahuja

University of Delhi, New Delhi, India

e-mail: ravinahujadce@gmail.com

R. Maskeliunas

Kaunas University of Technology, Kaunas, Lithuania

e-mail: rytis.maskeliunas@ktu.lt

activities. In an effort to detect intrusion, the defender must have a very clear and precise understanding of the attack process, how it works and penetrates [1].

Usually, whenever there is a network intrusion, some of the available and valuable network resources meant for authorized user are completely absorbed and trampled upon. Therefore, the need to design and deploy an efficient network intrusion system which will prevent the intruders and hackers is essential [2].

An intrusion detection system therefore can simply be defined as an application that controls, monitors, manages, and directs a network of system from malicious and unauthorized activities that might violate various established network [1].

A network intrusion detection system is a type of intrusion detection technique that combines outputs from multiple sources and uses different approaches to detect these activities and distinguish the unwanted activity from authentic ones [3]. A lot of works have gone into developing efficient systems to detect and predict network intrusions [4].

From written literature on the development of computer framework, one noteworthy is the work of [5], who proposed a model of intrusion-detection by monitoring and statistically analyzing the auditing of system's records should in case there is abysmal and abnormal style in system usage and application. This approach is known as misuse-based detection which can detect only the known attack, but new attacks cannot be identified [6].

## 2 Literature Review

As available in many publications (both old and recent), it is very evident that many works have been carried out in the area of network intrusion [7]. In this section, therefore, efforts are made to review related works carried out by prominent researchers.

In an attempt to finding lasting solution to network intrusion [8, 9], applied a method that made use of k-means clustering to produce numerous training subsets. Neuro-fuzzy models were employed, and finally, an SVM classification model known as the radial SVM model to demonstrate the ability to attain a higher intrusion detection rate.

The result of the demonstration showed a higher performance against back-propagation and decision tree machine learning algorithms. The data used in this procedure was the KDD Cup 99 datasets.

A multiclass chi-square feature selection was proposed by [10] to reduce the number of feature to an optimal set. Also, the proposed method was chosen due to the lack of works using multiclass SVM in intrusion detection [11]. The results of the analysis show a better performance in the reduced number of false alarm when compared to other techniques.

Nidhi et al. [12] proposed the use of a four-layer framework of consecutively preprocessing, encoding, and classifying the data while integrating with a neural network to effectively detect intrusion attacks.

The experiment was conducted using KDD Cup 99 dataset. The result showed a better performance than the most existing system. In the second model, there was an increase in the accuracy of attack while significantly reducing the time.

Mahalanobis distance characteristic ranking was used by Zhao et al. [13] to choose important and vital features as well as improved search algorithm to select a variety and combination of features. The procedure adopted was very helpful to identify and decrease useless and unimportant features with the aim of improving the accuracy and precision of the results. Both the KNN and SVM algorithms were used for evaluating the techniques on the 99 datasets of KDD CUP. The experimental results showed that false rate is low, especially with the reduced feature subsets.

Bayesian network classifier was used by Fengli and Dan [14] to carry out an efficient and effective feature selection with NSL-KDD dataset. The aim of this approach was to carry out a comprehensive comparative analysis with other feature selection techniques. The results obtained show that Bayesian network classifier used low time rate for attack detection and provide an increased precision rate of detection.

Zhao et al. [13] proposed a framework to reduce redundant features, thereby reducing computational cost on the intrusion detection process. This approach uses the information gain algorithm together with chi-square for feature selection after which maximum entropy classifier was used to analyze the data. The dataset used for this procedure is the KDD CUP 99. The results show a 100% accuracy even though some features have been removed.

Fengli and Dan [14] proposed a hybrid feature selection framework based on principal component analysis (PCA) and fuzzy adaptive resonance theory (FART) for improving the detection accuracy of intrusion attacks especially the root2 local attacks. The PCA algorithm was used to reduce the dimensionality of the dataset attributes without losing vital information. The fuzzy adaptive resonance theory was used for classifying the resulting dataset after the feature selection process. The procedure was carried out on the benchmark data from KDD Cup 99 dataset.

The results of the proposed framework as compared to the result of procedures for both [15] and [16] show that the method outperforms the two in terms of detection rate and false alarm rate.

Poojitha et al. [17] proposed an approach based on multiple classifier systems. It uses a pattern recognition approach to extract suitable feature based on the characteristics that distinguish each network activity to produce three main classes. The classification problem was then subdivided into smaller classification tasks of each task related to one of the three classes. The classification result of each task was then fused into a single output based on three fusion techniques, namely voting rule, average rule, and the belief function. The procedure was carried out on UCI KDD dataset by DARPA, and the results show that the error rates of most attacks were kept very low also with low false alarm rate.

A multi-layered scheme was proposed by Adel and Mohsen [18], for intrusion detection. They adopted genetic algorithms, neuro-fuzzy networks, and fuzzy inference approach for the effective analysis of KDD Cup 99 dataset. The scheme has two main layers, with the first layer utilizing a neuro-fuzzy classifier to produce a distinct class labeled result, while the second layer utilizes a fuzzy inference module.

The result of the experimentation showed a high precision and a good performance in DOS attacks analysis [19].

### 3 Methodology

#### 3.1 Data Exploration

1999 KDD cup dataset was used for the implementation. This dataset was formed by processing portions of the 1998 DARPA IDS evaluation dataset which was also established by MIT Lincoln Lab. A close network was used to generate the artificial data. Hand-injected attacks were used to produce many different types of attack with normal activity in the background.

As the initial goal was to produce a large training set for supervised learning algorithms, there is a large proportion (80.1%) of abnormal data which is unrealistic in real world and inappropriate for unsupervised anomaly detection which aims at detecting “abnormal” data. The KDD 99 dataset generated over the span of 7 weeks is made up of half a billion records, 42 features, and 23 different types of attack. There are three main types of features in this dataset.

There are nine (9) recognized features of individual TCP connections, and content feature has thirteen (13) standard features within a connection, while there are nineteen (19) traffic features when a two-second time window is computed (Table 1).

#### 3.2 Classification Models

Basically, three different models were used to evaluate the KDD dataset and demonstrate performance. So, what are the classification models?

A classification model is a data mining operation that attempts to draw some conclusion by observing a set of tuples. Given one or more tuples, a classification model will try to predict the value of one or more outcomes.

Native Bayes, decision tree, and random forest classification models were used in this analysis.

Decision was reached on the first three algorithms because of their popularity along with observable contradictory results obtained on them from previous researches. What is more, they can provide relatively good performance on the classification task.

**Naïve Bayes.** It is considered as one of the prominent machine learning algorithms for data classification with belief of independence between a pair of features. This theorem basically tries to use a known outcome to predict a sequence of events that may have led to that outcome. It is basically used for text classification and involves the use of high-dimensional training dataset.

**Table 1** Feature description

Name	Description
<i>Basic features</i>	
Duration	Length (number of seconds) of connection
Protocol_type	Type of protocol, e.g., tcp, udp
Service	Network service on the duration, e.g., http, telnet
src_byte	Number of data byte from the source to destination
dst_byte	Number of data bytes from destination to source
Flag	Normal or error status of the connection
Land	1 if connection is from the same host/port, 0 otherwise
Wrong_fragment	Number of “wrong” fragments
Urgent	Number of urgent packets
<i>Content features</i>	
Hot	Number of “hot” indicators
num_failed_logins	Number of failed login attempts
logged_in	1 if login successful else 0
num_compromised	Number of compromised conditions
root_shell	1 if root shell is obtained else 0
su_attempted	1 if “su command” attempted else 0
num_root	Number of “root” accessed
num_file_creation	Number of file creation operation
num_shells	Number of shell prompts
num_access_files	Number of operation on access control files
num_outbound_cmds	Number of outbound commands in an ftp session
is_hot_login	1 if the login belongs to the hot else 0
is_guest_login	1 if login as guest else 0
<i>Traffic features</i>	
Count	Number of connection to the same host as the current connection in the past 2 s
serro_rate	% of connections that have “SYN” error
rerror_rate	% of connection that “REJ” error
same_srv_rate	% of connection to the same service
diff_srv_rate	% of connection to different services
srv_count	Number of connections to the same service as the current connection in the past 2 s
srv_serror_rate	% of connection that have “STN” error
srv_rerror	% of connection that have “REJ” error
srv_diff_host_rate	% of connection to different hosts

There are several applications attributed to Naïve Bayes algorithm. The prominent among them are document categorization, email spam detection, sexually explicit content detection, personal email sorting as well as language and sentiment detection.

This is regarded as a form of simple probabilistic classifiers that depends on Bayes' theorem with independence belief and assumption between the pair of features. With Naive Bayes algorithm, training of dataset can be done efficiently and reliably. With it, one can make accurate and fast predictions on the dataset [20]. It assists to compute the conditional probability distribution of each feature in a given dataset. Naïve Bayes is majorly used in some areas of application such as text retrieval, text categorization, and the problems related to judging documents. It is mathematically represented as:

$$\rho(A|B) = \frac{\rho(B|A)\rho(A)}{\rho(B)} \quad (1)$$

$\rho(A)$ : Likelihood that A and B occur independent of each other.

$\rho(B|A)$ : Likelihood that B occurs given that A also occurs.

$\rho(A|B)$ : Likelihood that A occurs given that B occurs.

**Decision Tree Model Algorithm.** Decision tree assists in building classification recursively by dividing a given dataset into subsets. It uses a tree-like graph for decision making where the possible consequences include but not limited to resource cost, chance event outcomes, and utility. It is usually used in decision analysis to identify a technique that can be best used to achieve a goal [19].

**Algorithm 1:** Algorithm for decision tree model

start

- Begin at the root
  - Carry out the test
  - Follow and trace the edge corresponding to the outcome
  - Go to 2 except leaf
  - Forecast the outcome associated with the leaf
- stop

### Decision trees used in data mining are of two main types:

- **Classification tree** which evaluates different combinations of features of an instance to determine the class to which the instance belongs.
- **Regression tree** is used when the outcome of the evaluation is a numeric value (e.g., the price of a good).

The term **classification and regression tree (CART)** was first introduced by LEO BREIMAN in 1984. This is a simple method for building both classifiers and regressors. The input space is partitioned into two dimensions.

**Decision tree learning** is a machine learning algorithm that uses the tree models to try to predict a set of outcomes base on an input. This is one of the prominent and important techniques for data classification. It has a structure of flowcharting with

each internal node connotes a status test on an attribute. Each of the branches stands for the outcome of a test, while each node embraces a class label. The root node is regarded as the topmost node. There are many types of decision tree algorithms. Among them are 4.5 (this is an extension of the basic ID3 algorithm), chi-squared automatic interaction detector (CHAID), classification and regression tree (CART) and Iterative Dichotomiser 3 (ID3), conditional inference trees, and multivariate adaptive regression splines (MARS) [19].

**Random Forests.** These are known machine learning algorithms that ensemble decision trees. They are majorly used for regression and classification. One of the main benefits of random forests is in their capability to reduce the risk of overfitting after combining many decision trees. Random forests have similar features like decision trees which include capturing feature interaction and nonlinearity. What is more? They also handle categorical features. Random forests train a set of decision trees in parallel. They do this by injecting randomness into the process of training. Combination of predictions from different trees enhances the performance on test data.

### 3.3 Data Analysis

The KDD Cup 99 consists of both training (labeled) and test (unlabeled) datasets. Because these datasets contain over a billion records, only 10% subset of each was used for this analysis. The raw data at this point is messy and cannot be used directly for this work. This data must undergo data clean up and feature engineering steps to make it suitable for analysis.

These two steps addressed the quality issue in these datasets which are as follows:

- Inconsistent values
- Duplicate records
- Missing values
- Invalid data
- Outliers
- Bias data
- Low variance data (irrelevant features).

## 4 Implementation

### 4.1 Cleaning up the Data

The training dataset was too large and time-consuming for building models, so only 10% of the data was used amounting to about 494,021 records in the dataset. Duplicate rows were removed from the dataset resulting in a total of 145,586 of the training data and 77,291 of the testing data.

The distribution of the training data shows that about 60% of the training data has been labeled as normal which makes this dataset bias. A biased data like this will have a large impact on the analysis, especially where the analysis is focused on classifying intrusion. To resolve this, the data was normalized by reducing the number of records labeled as normal.

## 4.2 Feature Engineering

This helps to reduce the data amount because there are a lot of features in this dataset. There are 41 features in total, and some of these features are either useless or irrelevant to the intrusion detection problem, so by selecting only useful features, any meaningless calculation can be avoided.

This also helps to improve the accuracy by removing misleading or unrelated features. Some feature correlated with each other can cause overfitting. Also, the variance of the values of each feature is calculated; this is done to remove features in the dataset that have the same set of values and therefore reducing the amount of unnecessary work to be done in training the dataset.

Finally, feature normalization was done on the dataset. Some features have a range of values which are very high. For example, in the dataset, the features `src_byte` and `src_dest` have their values ranging from 0 to more than 50,000, while many features have their values ranging from 0 to 1. It is wise to normalize the features so that all the features will have influence on the result.

Also, there are other methods to reduce noise in the output values like early blockage. In achieving this, we use algorithms for identifying noisy training and completely eliminate the likely and suspected noisy training. This is good as early detection is good and not expensive to implement. At the end of the preprocessing stage, there are 39 features and 145,586 records left for classification.

The experiment was set up on Intel Core i7 processors, 8 GB RAM, 1 TB HDD, Windows 10 PC, and Weka machine learning workbench was utilized for the classification task. The classification was performed based on the 22 attack categories. The testing of dataset was processed using the Naive Bayes, decision tree, and random forest classifiers.

In measuring the performance of each of the techniques used, we adopted accuracy, precision, sensitivity, and specificity rate with expressions hereunder:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (2)$$

$$\text{Sensitivity} = \frac{TP}{TP + FN} \quad (3)$$

$$\text{Specificity} = \frac{TN}{FP + TN} \quad (4)$$

$$F\text{Score} = \frac{2TP}{2TP + FP + FN} \quad (5)$$

**Table 2** Confusion matrix

		Predicted classes		
		a	b	c
Actual classes	a	TP		
	b		TP	
	c			TP

**Table 3** Accuracy for training dataset for each algorithm

Accuracy (%)				
	First run	Second run	Third run	Average
Naïve Bayes	77.04	76.77	75.41	76.41
Decision tree	99.86	99.85	99.83	99.85
Random forest	99.93	99.91	99.92	99.92

$$Precision = \frac{TP}{TP + FP} \quad (6)$$

where FN is false negative, TN is true negative, TP is true positive, and FP is false positive. The accuracy is determined by finding the probability of a correct classification which is calculated by dividing the total number of attacks detected by the total number of attacks in the dataset.

The sensitivity is the ability of the system to detect an anomaly, and specificity is the ability of the system to correctly rule out an attack in a normal connection.

A “confusion matrix” in most cases can be used to signify the result, as shown in Tables 2. This table correlates all the actual classes in the row against the predicted classes in the columns. Each class is represented by a short character. For example, the class “back” is represented by the character “a”. In confusion matrix, a cell which has the same class for both the row index and column index is the true positive, while other cells are either false positive or false negative.

The total accuracy of the algorithm was calculated from the confusion matrix. The accuracy is the ratio of a number of correctly classified instances or record to the total number of instances or record set.

From Tables 1, 2, 3 and 4, it can be deduced that the diagonal cell shows the numbers of correctly classified records (instances) which are known as the true positives, while the rest of the cell holds are miss-classification count for the corresponding class. The miss-classified instances can be referred to as either false negative or false positive depending on context. The total accuracy is the ratio of sum of TP divided by the total number of records

$$Total\ Accuracy = \frac{\sum TP}{Total} = \frac{TP + TN}{Total} \quad (7)$$

**Table 4** Performance evaluation of each model for all classes

	Precision						Sensitivity						Specificity						Accuracy					
	NB	DT	RF	NB	DT	RF	NB	DT	RF	NB	DT	RF	NB	DT	RF	NB	DT	RF	NB	DT	RF			
Back	97.32	99.73	100.00	97.40	99.14	99.90	99.98	100.00	100.00	99.96	99.99	99.99	99.99	99.99	99.99	99.99	99.99	99.99	100.00	100.00	100.00			
Teardrop	98.54	100.00	100.00	99.94	100.00	100.00	99.99	100.00	100.00	99.99	100.00	100.00	99.99	100.00	100.00	99.99	100.00	100.00	100.00	100.00	100.00			
Loadmodule	2.17	22.22	88.89	83.33	19.44	63.89	99.83	100.00	100.00	99.83	100.00	100.00	99.83	100.00	100.00	99.99	100.00	100.00	99.99	100.00	100.00			
Neptune	99.99	99.98	99.98	99.50	99.98	100.00	99.99	99.99	99.99	99.99	99.99	99.99	99.99	99.99	99.99	99.99	99.99	99.99	99.99	99.99	99.99	99.99		
Rootkit	0.34	0.00	—	37.50	0.00	0.00	98.06	100.00	100.00	98.05	100.00	100.00	98.05	100.00	100.00	99.99	100.00	100.00	99.99	100.00	100.00			
phf	2.16	83.33	100.00	100.00	100.00	100.00	99.91	100.00	100.00	99.91	100.00	100.00	99.91	100.00	100.00	99.91	100.00	100.00	99.99	100.00	100.00			
Satan	46.80	98.18	99.92	93.23	97.16	97.23	99.16	99.99	100.00	99.11	99.99	99.99	99.99	99.99	99.99	99.99	99.99	99.99	99.99	99.99	99.99			
Buffer_overflow	11.54	77.41	86.10	25.00	81.67	97.78	99.92	100.00	100.00	99.90	99.90	99.90	99.90	99.90	99.90	99.90	99.90	99.90	99.99	99.99	100.00			
ftp_write	0.18	33.33	100.00	50.00	16.67	50.00	99.00	100.00	100.00	98.99	100.00	100.00	98.99	100.00	100.00	98.99	100.00	100.00	99.99	100.00	100.00			
Land	48.23	88.80	89.93	96.97	83.11	93.64	99.98	100.00	100.00	99.98	100.00	100.00	99.98	100.00	100.00	99.98	100.00	100.00	99.99	100.00	100.00			
Spy	—	0.00	—	—	—	—	—	—	—	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00			
ipsweep	8.94	99.40	99.82	83.94	98.47	97.11	95.15	100.00	100.00	95.09	100.00	100.00	95.09	100.00	100.00	99.99	100.00	100.00	99.99	100.00	100.00			
Multihop	4.60	0.00	100.00	55.56	0.00	77.78	99.95	100.00	100.00	99.95	100.00	100.00	99.95	100.00	100.00	99.95	100.00	100.00	99.99	100.00	100.00			
Smurf	6.47	100.00	100.00	99.42	98.79	99.69	92.65	100.00	100.00	92.69	100.00	100.00	92.69	100.00	100.00	99.99	100.00	100.00	99.99	100.00	100.00			
pod	1.86	100.00	100.00	98.82	100.00	99.70	92.36	100.00	100.00	92.37	100.00	100.00	92.37	100.00	100.00	92.37	100.00	100.00	99.99	100.00	100.00			
Perl	0.00	100.00	100.00	0.00	66.67	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00			
Wareclient	13.75	97.44	99.25	49.14	97.35	98.15	97.47	99.99	100.00	97.09	99.97	99.97	99.97	99.97	99.97	99.97	99.97	99.97	99.99	99.99	99.99			
nmap	6.21	92.32	99.00	21.81	92.83	97.02	99.52	99.99	100.00	99.41	99.99	99.99	99.99	99.99	99.99	99.99	99.99	99.99	99.99	99.99	99.99			
imap	23.98	83.34	100.00	81.11	28.89	67.78	99.98	100.00	100.00	99.97	100.00	100.00	99.97	100.00	100.00	99.97	100.00	100.00	99.99	100.00	100.00			
Warezmaster	3.71	68.89	95.83	82.87	76.85	82.87	99.67	100.00	100.00	99.67	100.00	100.00	99.67	100.00	100.00	99.99	100.00	100.00	99.99	100.00	100.00			
Portsweep	17.12	97.00	98.27	91.44	96.71	98.06	98.20	99.99	100.00	98.18	99.98	99.98	99.98	99.98	99.98	99.98	99.98	99.98	99.99	99.99	99.99			
Normal	99.90	99.88	99.91	62.18	99.93	99.99	99.91	99.82	99.86	76.96	99.88	99.88	99.88	99.88	99.88	99.88	99.88	99.88	99.94	99.94	99.94			
Guess_passwd	31.16	97.70	100.00	90.49	90.49	92.25	99.91	100.00	100.00	99.91	100.00	100.00	99.91	100.00	100.00	99.91	100.00	100.00	99.99	100.00	100.00			

## 5 Results and Discussion

Table 3 shows the accuracy of each classifier for each run and a computed average. Naïve Bayes classifier performed the worst in detecting most of the attacks with an average accuracy of 76.41%, while random forest algorithm is the best with an average accuracy of 99.92 followed by decision tree with 99.85% accuracy in average.

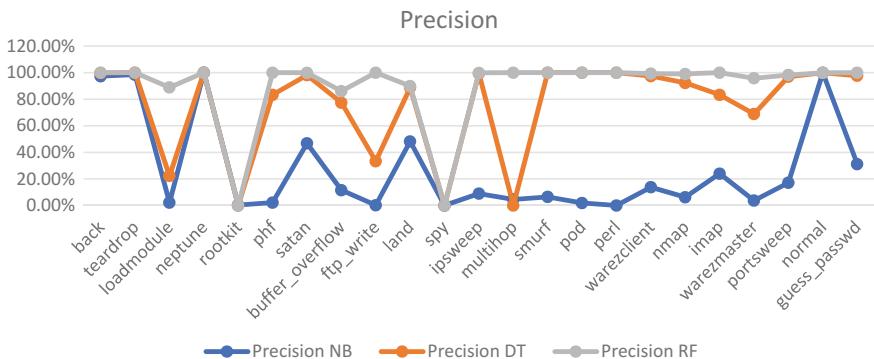
The test result for each classifier is summarized in Table 4. The table represents the measure in terms of average precision, sensitivity, specificity, and accuracy for the three classifiers. The averages are computed from the three different test runs labeled as NB for Naïve Bayes, DT for decision tree, and RF for random forest.

Figure 1 shows the average precision of each algorithm against all the attack types. Naïve Bayes scores the least in this evaluation. Also, Fig. 1 shows a very low precision loadmodule, rootkit, spy meaning the algorithm falsely flag them as attacks especially Naïve Bayes.

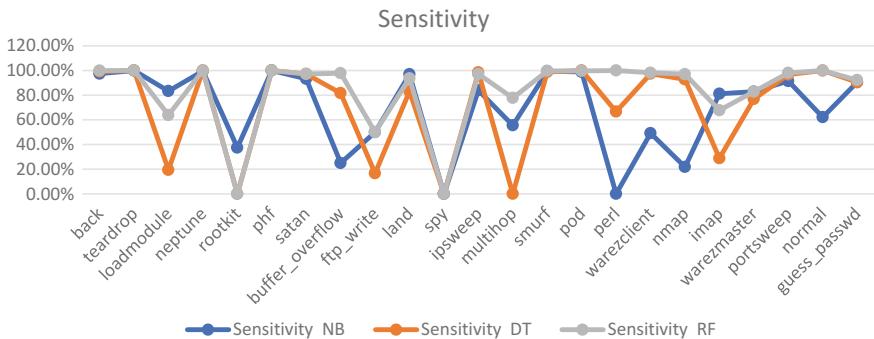
Figure 2 also shows the average sensitivity of each algorithm. Again, there is very low sensitivity loadmodule, rootkit, spy, meaning the algorithm could not correctly flag them as attacks.

Figure 3 shows the average specificity of each algorithm. This figure shows that all three algorithms could to some degree correctly specify which attack it was, and Naïve Bayes still scores the least in this evaluation.

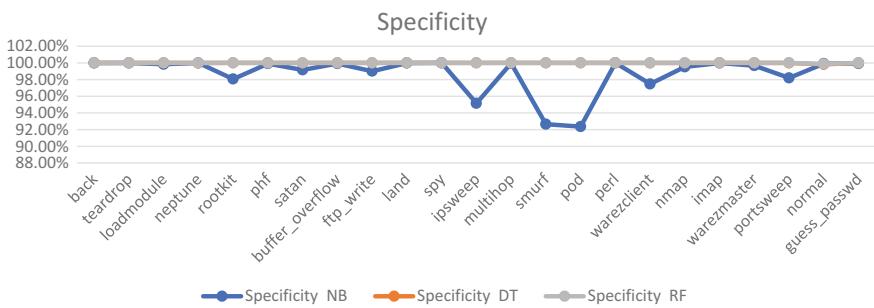
Figure 4 shows a good performance on the accuracy of the algorithm, especially the random forest which had the best accuracy across all attacks, while Naïve Bayes still performed the least.



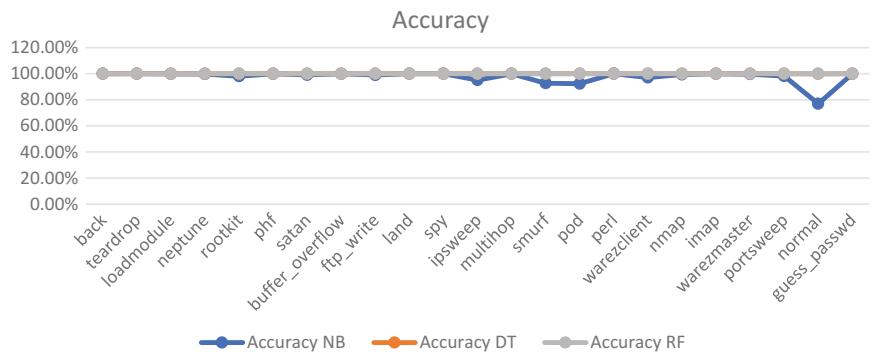
**Fig. 1** Precision evaluation of each model for all classes



**Fig. 2** Sensitivity evaluation of each model for all classes



**Fig. 3** Specificity evaluation of each model for all classes



**Fig. 4** Accuracy evaluation of each model for all classes

## 6 Conclusion

This work reviewed and evaluated the performance of three of the commonly used machine learning algorithms for intrusion detection. These algorithms were evaluated using a big data and machine learning data processing tool developed by University of Waikato, New Zealand, called Weka. The authors used a real-time artificial dataset generated by MIT Lincoln Lab by simulating a closed network and hand-injected attacks. There are three runs in the procedure of which the dataset was split into different ratios (60 : 40, 50 : 50, 40 : 60) for both training and test data for each run. To improve the performance of the result, a lot of preprocessing was carried out on the data to remove correlated and useless features, overfitted, duplicate, biased and noisy data. This procedure also improved the time taken to classify the attacks. The accuracy of the all the algorithms was improved as the ratio of training data to test data was increased. This procedure saw a very good performance across the three runs for decision tree and random forest while experiencing a poor performance from Naïve Bayes algorithm.

## References

1. Azeez, N. A., & Ademolu, O. (2016). CyberProtector: Identifying compromised URLs in electronic mails with Bayesian classification. In *International Conference Computational Science and Computational Intelligence*, pp. 959–965.
2. Azeez, N. A., Okunoye, O. B., Oladeji, F. A., & Edafeadjeke, E. O. (2015). Towards an adaptive and scalable access control model for a cloud-based environment. In *Nigerian Computer Society (NCS) 12th Annual Conference International Conference on Information Technology for Inclusive Development* (Vol. 26, pp. 214–223).
3. Singh, R., & Singh, D. (2014). A review of network intrusion detection system. *International Journal of Engineering and Technoscience*, 5(1), 10–15.
4. Giorgio, G., & Fabio, R. (2003). Intrusion detection in computer networks by multiple classifier systems. *Pattern Recognition Letters*, 1795–1803.
5. Denning, E. D. (1987). An Intrusion-detection model. *IEEE Transaction on Software Engineering*, 222–232.
6. Azeez, N. A., & Babatope, A. B. AANtID: An alternative approach to network intrusion detection. *Journal of Computer Science and Its Application*, 23(1) (2016).
7. Azeez, N. A., & Venter, I. M. (2013). Towards ensuring scalability, interoperability and efficient access control in a multi-domain grid-based environment. *SAIEE Africa Research*, 104(2), 54–68.
8. Azeez, N. A., & Irwin, B. (2010). Cyber security: Challenges and the way forward. *GES: Computer Science and Telecommunications*, 1512–1232.
9. Chandrasekhar, A. M., & Raghuveer, K. (2013). Intrusion detection technique by using k-means, fuzzy neural network and SVM classifiers. In *2013 International Conference Computer Communication and Informatics (ICCCI)*. Coimbatore, India.
10. Sumaiya, T. I., & Aswani, K. C. (2016). Intrusion detection model using fusion of chi-square and multi class SVM. *Journal of King Saud University*.
11. Smaha, R. E., & Haystack. (1988). An intrusion detection system. In *Proceedings of the IEEE Fourth Aerospace*. Orlando, FL.

12. Nidhi, S., Krishna, R., Rama, K. C. (2013). Novel intrusion detection system integrating layered framework with neural network. In *IEEE 3rd International Advance Computing Conference (IACC)*. Ghaziabad.
13. Zhao, Y., Zhang, Y., Tong, W., & Chen, H. (2013). An improved feature selection algorithm based on MAHALANOBIS distance for network intrusion detection. In *Sensor Network Security Technology and Privacy Communication System (SNS & PCS), 2013 International Conference*. Nangang, China.
14. Fengli, Z., & Dan, W. (2013). An effective feature selection approach for network intrusion detection. In *Networking, Architecture and Storage (NAS), 2013 IEEE Eighth International Conference*. Xi'an, China.
15. Yang, L., Bin-xing, F., You, C., & Li, G. A (2006). lightweight intrusion detection model based on feature selection and maximum entropy model. In *Communication Technology, 2006. ICCT '06. International Conference*. Guilin, China.
16. Preecha, S., & Woraphon, L. (2015). Anomaly traffic detection based on PCA. *The International Arab Journal of Information Technology*, 253–260.
17. Poojitha, G., Naveen, K. K., & Jayarami, P. R. (2010). Intrusion detection using artificial neural network. In *2010 International Conference on Computing Communication and Networking Technologies (ICCCNT)*. Karur, India.
18. Adel, N. T., & Mohsen, K. (2007). A new approach to intrusion detection based on an evolutionary. *Computer Communications*, 2201–2212.
19. Hui, L., & Jinhua, X. (2009). Three-level hybrid intrusion detection system. In *International Conference on Information Engineering and Computer Science, 2009. ICIECS 2009*. Wuhan, China.
20. Zhang, H. (2004). *The optimality of Naive Bayes*. New Brunswick, Canada: University of New Brunswick.
21. Azeez, N. A., Ademola, P. A., Ademola, O. A., & Kehinde, K. A. (2011). Ancae: A novel clustering algorithm for energy efficiency in wireless sensor networks. *Wireless Sensor Network*, 307–312.
22. Balogun, A. O., & Jimoh, R. G. (2015). Anomaly intrusion detection using an hybrid of decision tree. *A Multidisciplinary Journal Publication of the Faculty of Science, Adeleke University, Ede, Nigeria*, 67–73.
23. Heady, R., Luger, G., Maccabe, A., & Servilla, M. The architecture of a network. Technical Report, Department of Computer Science, University (1990).
24. Govind, P. G., & Manish, K. (2016). A framework for fast and efficient cyber security network. In *6th International Conference on Advances in Computing & Communications, ICACC 2016*. Cochin, India.
25. Manning, C. D., Raghavan, P., & Schutze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.

# Super-Intelligent Machine Operations in Twenty-First-Century Manufacturing Industries: A Boost or Doom to Political and Human Development?



I. A. P. Wogu, S. Misra, P. A. Assibong, S. O. Ogiri,  
R. Damasevicius and R. Maskeliunas

## 1 Introduction

Recent studies reveal that achievements in the field of artificial intelligence (AI) indicates that today's machines have gone beyond just being robots with human-like characteristics for solving simple human problems like calculating and processing complex data as seen in Google search algorithms [1] which now can outplay grand masters in the sophisticated games of chess [2–4], AlphaGo [5] and in other complex games such as the poker game [6]. Today, these super-intelligent machines, studies reveal [6–10], have acquired some degree of consciousness and intentionality, a feature in machines which largely allows them to combine new algorithms and deep machine learning experiences for solving complex human-related problems, a

---

I. A. P. Wogu · P. A. Assibong

Department of Political Science & International Relations, College of Leadership Development Studies, Covenant University, Ota, Ogun State, Nigeria  
e-mail: ike.wogu@covenantuniversity.edu.ng

P. A. Assibong

e-mail: patrick.assibong@covenantuniversity.edu.ng

S. Misra (✉)

Department of Electrical Information Engineering, College of Engineering, Covenant University, Ota 1023, Nigeria  
e-mail: sanjay.misra@covenantuniversity.edu.ng; ssopam@gmail.com

S. O. Ogiri

Department of Psychology, Covenant University, Ota, Ogun State, Nigeria  
e-mail: sophie.ogiri@stu.cu.edu.ng

R. Damasevicius · R. Maskeliunas

Department of Software Engineering, Kaunas University of Technology, Kaunas, Lithuania  
e-mail: robertas.damasevicius@ktu.lt

R. Maskeliunas

e-mail: rytis.maskeliunas@ktu.lt

feature never before was believed could be possible to machines. This AI feature in the field of computer science, in many ways, mimics the human brain, a feature which allows machines to teach themselves and acquire new knowledge which their inventors may not have initially intended or proposed for them to acquire. This perhaps would explain why Max Tegmark, the President of the Future of Life Institute (FLI), expressed his fears about the possible consequences of these advancements in [9].

As ridiculous as this idea may sound, the thought that machines have now developed and acquired minds of their own which enables them to do and archive nearly impossible feats [10], which is a bigger issue arising from research in machine intelligence [11, 12]. The advances in AI technology and the innovations and simulations of human-like characteristics into the designing of machines and devises in the twenty-first-century, now more than before, are discovered to pose serious existential risks to humanity in the labour industry. Marquart [13] corroborate this fear when they observed that ‘machines are coming to the threshold of possessing super-intelligent abilities in no distant time’, a trait which is believed, ‘will give machines the dominant advantage over their human counterparts no sooner than is expected’ [14]. The manufacturing industry seems to be the worst hit by this threat.

## 1.1 *The Problem*

The rise of super-intelligent machines and technology stare’s up recollections of the movie recreation by Mary Shelley’s *Frankenstein*, which in the eyes of many, provoked both the feeling of horror and awe, to a majority of researchers and thinkers, these advances in AI innovations sparked off fascinating and exciting possibilities that can only be limited by the imaginations of individuals. These inconsistencies notwithstanding, renewed studies on advances in AI research and technology [7, 9, 13, 15, 16, 17, 18], indicate that there are rising concerns about the place of man in the presence of these innovations in AI technology and its direct implications and consequences on the existential state of persons who work in manufacturing industries (MIs) for their sustenance and livelihood. With the reality of these AI innovations now existing in smart phones, automated self-driving cars and in most heavy duty machines in MIs, one big question that looms man in the face is the question of ‘whether the benefits of AI outweighs the existential risks it poses to workers in the labour force?’. Another group of studies conducted by FLI [1, 7, 8, 16, 17], for instance, all focused on raising serious awareness about what Moshe Vardi in [14, 18] described as the *scary extinction risks*.

## 1.2 *Objectives of the Study*

In the light of some of the research questions raised in the above page, this study, among other things, focused on doing the following:

1. Evaluating critically, the existential risks claim and the perceived benefits believed to be associated with rising innovations and automations of human jobs by super-intelligent machines and technology.
2. Attempt a valid inference of the future of mankind in the light of rising innovations and automations in the manufacturing industry.
3. Identify and recommend pathways that should facilitate aligning the goals of super-intelligent technology with those of mankind in the twenty-first century.

## 1.3 *Methodology and Theoretical Foundations for the Study*

Karl Marx's *alienation theory* which essentially highlights the various kinds of estrangements that individuals experience from aspects of their essence is adopted for the study [19–21]. The theory was chosen because it offers basic foundations and justifications for conducting investigations in the subject areas of the paper. This paper argues that the situation arising from this context creates derogatory relations which makes man feel very helpless in the face of the forces of technology he created himself [14, 21, 22]. The ex post facto research method [22, 23] was considered an appropriate methodology for the study, since it largely relies on the analysis of data from other studies, debates and arguments conducted on the advantages and disadvantages of super-intelligent machines, now fully operational in today's manufacturing industries. Deriders' deconstructive and critical reconstructive analytic method of enquiry in philosophy [24–26] was also adopted for the study because it essentially interrogates the meaning of concepts, arguments and issues in the ongoing debates on the pros and cons of innovations in AI technology and the existential threats of super-intelligent machines to mankind in the twenty-first century.

## 2 The Advent of Artificial Intelligence

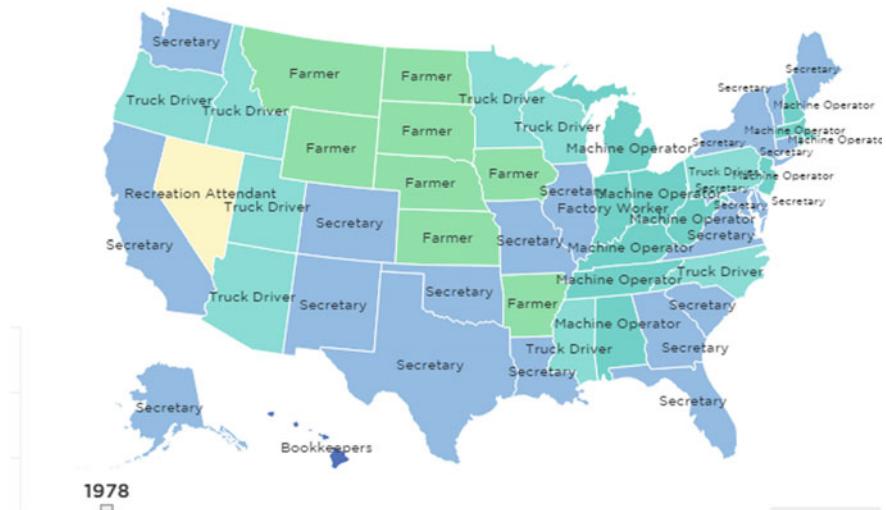
AI as a concept is often referred to as 'the general mental capacity to process data intellectually and abstractly, which results in the knowledge of new ways for addressing issues for the benefit of mankind' [2]. Most researchers ascribe the feature of intelligence to man alone; hence, those in this category tend to reject the idea that such feature of intelligence is replicable in artefacts such as machines. The belief that intelligence is exclusive to man alone is premised on the notion that the feature of intelligence gives man the edge and dominant advantage of reason and resourceful retrospection, a feature that makes it possible for him to thrive well and surpass all his

contemporaries in all things [27]. Any other things or artefact which seems to possess this similar feature of intelligence can only be regarded as *artificial intelligence* (AI) and not intelligence as it was.

## 2.1 Artificial Intelligence (AI)

The diverse notions of intelligence notwithstanding, the advent of AI technology and super-intelligent machines (SIMs) since the turn of the century have sparked off several renewed debates focused on affirming or negating the pros and cons perceived to be associated with the knowledge that machines can now simulate and in most cases outdo man in nearly all the features known to be typical of man and humans alone, since the last 20 years. Responding to the debate about whether machines could really possess the feature of intelligence or not, Skinner in [28] observed that: ‘the real problem in the study of human behaviour and intelligence is not whether machines think, but whether human beings really engage in that act of reasoning’.

**The Most Common\* Job In Each State 1978-2014**



**Fig. 1** Common job distribution in the USA. Source Wade, L (2015). Sociological images <https://thesocietypages.org/socimages/2015/03/05/the-most-common-job-in-every-state-1978-2014/>



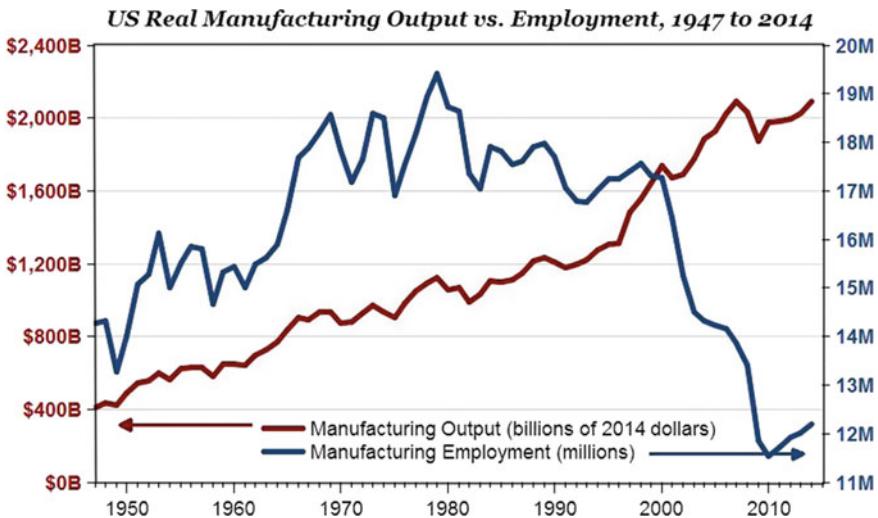
**Fig. 2** Indications of existential and ontological implications of innovations and automations in the manufacturing industry. *Source* Ferrari's Car Factory. <https://www.youtube.com/watch?v=EkSPMxyjcgg>

### 3 AI Technology and Manufacturing Industries in the Twenty-First Century

See Figs. 1 and 2

#### 3.1 AI Machines in Today's Manufacturing Industries

In view of some studies conducted between 2010, 2014 and 2015 and considered for this paper—as indicated in (Figs. 3, 4 and 5)—the authors of this article, alongside other researchers [18], found viable grounds for inferring that the advent of AI into the labour force and the MIs is the major propellant of the Fourth Industrial Revolution [29] currently taking place. Consequently, some scholars like [1, 7, 10, 14, 30, 31] were of the opinion that this revolution will herald the wiping out of nearly half of human jobs in the next two decades (see Figs. 4 and 5). Their opinion is corroborated by another study which reveals that virtually all owners of the means of production are now opting to replace the jobs of humans with super-intelligent machines that can do human jobs better at little or no cost implication to the owner. Their resolve in this direction is strengthened by the desire to boost the industry's turn-around productivity and also to increase its efficiency. By so doing, the MIs



**Fig. 3** US real manufacturing output versus employment (1947–2014) in the light of rising automation and innovations in AI technology. *Source* Daniel Miessler. (2014). US manufacturing is as strong as ever <https://danielmiessler.com/blog/u-s-manufacturing-is-as-strong-as-ever-we-just-need-way-fewer-people-to-do-it/>

maximize profits for their shareholders at the expense of humans and the labour force. Advanced MP technology (AMPT), an organization dedicated to providing technological service, corroborates this view thus:

Replacing humans with robots in manufacturing industries is a trend that we can't stop or avoid. As technology advances, the low cost, high-accuracy and efficiency of robot are going to benefit the human society as a whole on a broader level. At Advanced MP Technology, we always stay tuned with the most recent trend in electronic industry to stay ahead of the supply chain and provide the best service to our customers [31].

The opinion in the quotation is re-enforced by the fact that every activity which was typical to humans in time past is now possible to machines. So why pay more to humans if you can get the best and more at lesser cost from machines? [31].

The diagram in Fig. 2 is a good example of the direct consequence and effect of what some twenty-first-century MIs look like, as a result of the massive automation of human jobs going on in these MIs. Please note that there are virtually no human staff in the Ferrari assembling plant where production is going on. Only one or two persons are required to perhaps turn the machines on or off, especially when the need arises to either service or maintain the machines. Some 20 years ago, the place would have been swarming with individuals doing the jobs that machines are now doing very well. The diagram thus confirms the beginning of the existential risks which Davey in [18] and Hawking et al. [1] feared would soon become the fate of mankind if no new counter measures are put in place.

### 3.2 Existential Threats Issues in Today's Manufacturing Industries (MI)

An analysis of some studies conducted in the USA, as indicated in Figs. 3, 4 and 5, which were used for attaining the objectives of this study, reveals that delibera-

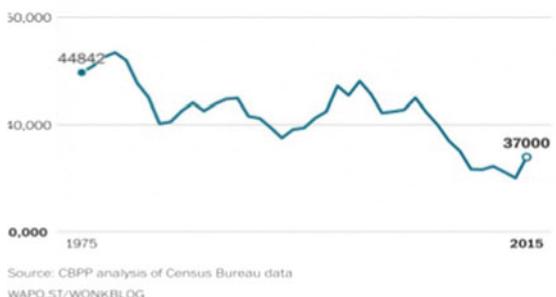


**Fig. 4** Productivity in GDP in the light of rising automations and innovations in super-intelligent/AI technologies for the labour force. *Source* Census Bureau, Bureau of Labour Statistics 2012

#### Real Median Income

##### Real median yearly incomes, white men with no college degree

Non-Hispanic white men saw their best income growth in more than a decade in 2015  
- but they're still way down from 1975 levels, after adjusting for inflation



**Fig. 5** Consequence on income earning for the ordinary white male in the USA. (Class I, II, III, IV Alienation). *Source* CBPP analysis of Censes Bureau WAPO.ST/WONKBLOG



tions about the threats posed by super-intelligent machines in American and in other industrialized countries/zones have gathered more trust and widespread recognition than the regular and commonly discussed issues like that of global warming and climate change, which before now were issues that were on the front burner [32, 33].

Most profound of all other existential threats to mankind is the one occasioned by the rising need for owners of the means of production to automate jobs with human-like features in virtually every human profession—for the purpose of boosting productivity and efficiency in today's MIs. This threat to existentialists is known as the threat to the *beingness* of humans in the world of work. The rising number of workers who lose their jobs on a daily basis is known to suffer from this kind of threat since their pain is exhibited in either of the four classes of the Marxian alienation [34]. This study is quick to observe that the kind of alienation experienced in this case is the kind occasioned by the efforts of certain individuals and owners of the means of production whose goals, among other things, are to maximize profits at all cost. (See Figs. 4 and 5). Kasanoff [35] discusses the behaviour and craze exhibited by certain individuals regarding the way they maximize profit in business at the expense of his fellow man. His report discussed in summary, the nature of man regarding profit maximization: 'Here's a quick summary of what's happening: humans can make money replacing humans with computers... but the number of humans who profit from this is a tiny fraction of the number of humans who lose their jobs because of this' [35]. The reality of this class of alienation, as indicated in (Figs. 3, 4 and 5), explains why there is a rise in the number of factory workers whose jobs in the current dispensation have become irrelevant and obsolete.

Against the claims that these innovations in technology are creating new jobs for the workforce which invariably replaces old jobs, there are those who believe that these claims cannot be sustained in the light of present circumstances. This is because, while a few individuals are able to find new jobs in this new dispensation to remain in the work force, the majority who lose their old jobs are rendered jobless since they are not able to find new jobs in the wake of massive automation of routine and non-routine jobs. This existential reality is foretold as one situation which mankind may not be able to phantom or comprehend, seeing that never before has mankind competed with its own creation in the world of work [19–21]. Erik Brynjolfsson and Andrew McAfee corroborate this view when they argued that:

There's never been a better time to be a worker with special skills or the right education, because these people can use technology to create and capture value. However, there's never been a worse time to be a worker with only 'ordinary' skills and abilities to offer, because computers, robots and other digital technologies are acquiring these skills and abilities at an extraordinary pace [36].

The scenario explained in the quotation above perhaps further explains why in Fig. 3, the study recorded a sharp decline in the number of people employed during the period under review. Interestingly, however, this decline in the number of people employed does not in any way affect the productivity of the MIs. On the contrary, these MIs continue to record outstanding increase in outputs in billions of dollars since year 2000.

In Fig. 4, the series of studies conducted for this research reveals that while productivity in rising GDP is sustained for jobs with no automation, private employees are also able to sustain production. But median jobs like the manufacturing jobs who get their jobs automated every now and then tend to have a drop in the productivity scale. This scenario that Karl Marx argues establishes the presence of alienation in all its various stages. The diagram in Fig. 5 shows how in the past 50 years in the USA, there has been a steady decline in the number of persons who are able to remain in employment, in view of rising automations and innovations in AI technologies. Consequently, you find these classes of unemployed person—usually between the ages of (25–44)—vent their anger in some of the recent cases of riots recorded in America's recent history. A more recent case was recorded with a sect known as the Klu Klux Klan (KKK), believed to be strong supporters of the Trump administration. Analysts observed that the real reason for their anger is the feeling of alienation which they now suffer from as a result of rising loss of job for the age bracket identified above.

### ***3.3 AI, Karl Marx Alienation Theory and Human Development***

Some socio-scientific researchers basically associate existential and ontological threat problems arising from studies on AI technology with Karl Marx theory of alienation [19, 21, 34]. This Marxian theory basically seeks to understand how the minds and bodies of individuals are affected by the type of work they regularly participate in for the purpose of earning a means of sustenance for themselves and their families. Karl Marx in this alienation theory largely sought to address the fundamental question: ‘How do the ways in which people earn their living affect their bodies, minds and daily lives?’ [20]. The four classes of alienation arising from the study Karl Marx conducted include: (1) alienation of the worker from his work and its product, (2) alienation of the worker from working and production, (3) alienation of the worker from what Karl Marx called their *Gattungswesen* (species-essence) and (4) alienation from human nature [19, 34].

The appropriate response to this question was delivered in one of Karl Marx’s alienation theory documented in the *Economic and Philosophic Manuscripts of 1844* by Karl Marx also referred to as The Paris Manuscripts. Marx in the document observed that ‘because workers in the capitalist economy do not own the means of production (machines) nor do they own the materials (factories) which are necessary for production, they are left with no choice than to sell their ‘labour power’, which is their ability to do work, without which they cannot earn any wage’. Where these factors of production are further disrupted or taken away, the worker is seen to experience either of the four forms/classes of alienation [34] described by Karl Marx.

In view of this, most capitalist economies now strive to cut down the cost of production via the transformation and replacement of the production processes with tools forged from the crucible of AI technology. A tool identified to enhance and

drastically boost production at little or no cost to the owners of the means of production. In the words of Judy Cox: ‘Never before have we felt so helpless in the face of the forces we ourselves have created’ [21]. Consequently, while a few prefer to identify these technological advancements in human nature as points and basis for making a case and affirming the existence of development in a polity, there are a host of scholars who believe that until development positively transforms and influences both the psyche and the environment of individuals, to the point where he/she is totally free from all forms of Alienation, development cannot be said to have been established.

### ***3.4 Analysis of Selected Studies on the Impact of AI in MI***

Apart from the studies whose findings and analysis are indicated in the charts and figures in (Figs. 1, 2, 3, 4 and 5) above, other studies considered for this paper focused on identifying the effects of the rising adoption of super-intelligent machines and AI technology for manufacturing industries in the USA. This section discusses some of the salient findings recorded from the study and the corresponding type of Marxian alienation believed to be inherent in the findings made. For lack of space, we shall only discuss two here:

1. Studies reveal that ‘truck driving jobs’ (Fig. 1), one of the most lucrative jobs in the USA, may soon experience class III and IV Alienation, since super-intelligent machines and current AI technology have now successfully perfected the automation of self-driving cars and trucks favoured for its ability to have the capacity to reduce road hazards to the barest minimum, compared to the road hazards recorded by human drivers in the past. Hazards caused by either fatigue from long hours of driving or a syndrome known as ‘texting while driving’ are examples of some of the major causes of these road hazards caused by human drivers. Where jobs are lost as the consequences of super-intelligent machine operations, class 1, 2 and 4 Marxian alienation theory can be inferred to be at play.
2. In another study in 1990, it was observed in the state of Detroit that three of its top companies with a combined staff strength of 1.2 million work force were valued to worth \$65 billion dollars in stocks, shares, bonds, etc. But in 2016, it was observed that three top companies in the Silicon Valley, with only about the staff strength of 190,000 work force, were valued at \$1.5 trillion. The massive drop in the size of the workforce in the period under review creates Class III and IV Alienation. This ultimately affects human development.

The findings made from the instances discussed above are indicative of how innovations in AI technology and the automation of more human jobs by super-intelligent machines continue to heighten the existential threat inferred to exist by most researchers of AI and super-intelligent machines [1, 8, 37, 38]. This perhaps explains why Stuart Armstrong opined that contemporary researchers must begin to invest into finding how humans will begin to retain their existential relevance in

the face of rising AI technology research. Where this is ignored, super-intelligent machines may begin to make the decision of downsizing or clearly removing humans from the scene.

## 4 Super-Intelligent Machines and Human Development in the Twenty-First Century: Boost or Doom?

### 4.1 A Critical Review of Some Studies Considered for This Research

This section of the paper critically examines whether the advent of super-intelligent machines and technology really spells doom to mankind or, on the contrary, if it spells development to mankind on every side. In this regard, a critical review of all the studies and arguments presented for this study via the data presented in (Figs. 2, 3, 4 and 5) is further analysed in this section.

As it stands, there are many (scientists and social scientist) who believe that as far as it concerns AI technology, one can only be limited by ones' imaginations, especially in the presence of inventions like smart phones, self-driving cars, chess and poker playing robots and computers, all innovations and technology that have, in so many ways, lightened the burden of man and improved his condition of living here on earth. Other specific areas where AI technology has been very beneficial to man are in the management and processing of big and complex data. Google's DeepMind, for instance, has successfully managed large sums of data in the healthcare sector which now facilitates access to improved, timely and qualitative healthcare service delivery. According to Sapin [39], 'AI will also help better secure the Internet of Things (IoT) world by anticipating and fighting intruders more quickly than human beings can'.

These benefits and advantages notwithstanding advanced research into *ethical*, *existential* and *ontological* perspectives of IA innovations [1, 5, 7, 8, 16] reveals that there are renewed degrees of ontological and existential risk fears, emerging as a result of the improved ability of computers and super-intelligent machines to become aware of their environment and the need to protect themselves from their creators (humans). The fear arising from this reality has been described as *the scary extinction risks*. A fear already affirmed by Moshe Vardi in a study he carried out for (FLI) in [18].

The diagram and charts presented in (Figs. 2, 3, 4 and 5) add credence to the ontological fears, which have become a reality, in view of the millions of individuals who now have lost their jobs to robots and super-intelligent machines. A scenario that have also unfavorably affected the economic state of man to the point of exposing him to all the four classes of Karl Marxist alienation theory. Consequently, the authors of this paper observed that while the operations of AI could be likened to '... a double-edged sword', research in AI continues to advance, thus making it seem as though

the advantages inherent in its strides are worth taking the risks for, until perhaps when proven otherwise [39].

## 5 Conclusion

### 5.1 *Summary of Findings*

The arguments presented for the extinction risk claim in this study by Hawking et al. and Davey [1, 18], and several other studies conducted by FLI, MIRI, etc., (Figs. 2 and 3) largely go to affirm the existence of the rising ‘extinction risks fears’ already inferred to exist in degrees that are inimical to the psyche of all individuals directly or indirectly affected by the operations of super-intelligent machines. Studies here indicate that those already affected by this extinction risks—as indicated in (Figs. 4 and 5)—are forced into variant degrees and classes of Alienation. The direct consequence of this extinction risks fears was also identified to have both economic and political development implications, since those directly or indirectly affected by the massive loss of jobs from today’s manufacturing industries often resort to acts of violence and demonstrations as some of the avenue for venting the anger arising from joblessness. These classes of unemployed people increase the difficulty government has in governing states where the inhabitants are predominated by jobless and unemployed youths of working age (25–44). See Figs. 3, 4 and 5.

Results obtained from analysing the first and second objectives of this paper clearly highlight the urgent need for researchers to take fervent steps towards addressing the various issues arising from the rapid adoption of super-intelligent machines and technology for today’s manufacturing industries. The studies analysed for the objectives of this paper clearly revealed that there seem to be no clear-cut and generally accepted ethical codes and principles governing the practice and use of advanced AI technology in the world today. Hence, this study applauds the efforts made by the Future of Life Institute (FLI) who in their last conference called out for the international endorsement of 23 AI principles today known as (the 23 Asimolar principles). These 23 principles are the first attempt by researchers and scientist to introduce some degree of sanity into the field of research and use of AI and super-intelligent technology in the world generally. This paper therefore lends its voice to the call on researchers and scientists to extend research in the areas of understanding the complexities associated with the creations and use of AI in today’s manufacturing industries.

## 5.2 Conclusion

This paper largely investigated the consequences of the operations of super-intelligent machines in today's manufacturing industries, with the view to ascertaining whether the claims to the huge economic and development benefits (advantages) ascribed to mankind and the polity via innovations in AI truly outweigh the hazards and risks which other scholars largely allude to. Thus, the findings of this paper, as represented in (Figs. 2, 3, 4 and 5), gave the authors justified and rational grounds for making the following deductions:

While the benefits of the use of super-intelligent machines today result in enhanced outputs, increased efficiency in productivity for the owners of the means of production, etc., the hazards associated with the massive adoption of these super-intelligent machines (SIMs) and technologies have adverse iminical implications on the jobs of individuals who must lose their place in MIs so that the supposed increased productivity and enhanced output would take place. The dehumanizing pain arising from the loss of man's place in these industries is what Karl Marx described as the derogatory feeling of Alienation.

The paper discovered that millions of individuals directly or indirectly affected by the emergence of super-intelligent technologies suffer from one of the four classes of alienation prescribed by Karl Marx. The authors are therefore quick to observe that the kind of Alienations experienced by those who lose their jobs to innovations and automations in today's manufacturing industries has a kind of adverse crippling effect on the political economy and development of host nations. The diagrams in (Figs. 1, 2, 3, 4 and 5) are vivid examples which portray how a nation like America is presently wallowing under the consequences of the massive adoption of super-intelligent machines and technology by her manufacturing industries.

In closing, while this study acknowledges the fair resolution by Sapin, [39] who views the relationship of AI in MIs and on mankind as that of 'a double-edged sword', the opinion of the authors of this paper differs slightly from this position. Without prejudice to the views expressed earlier by other researchers on the subject of this paper, the authors here believe that the hazards associated with the operations of SIM for twenty-first-century MIs far outweigh the advantages so far discussed in the studies reviewed for the paper.

## 5.3 Recommendation

In the light of the above findings and deductions made in this study, the paper found grounds and reasons for making the under listed recommendations:

- Researchers necessarily need to advance further research towards identifying ways of aligning the goals of super-intelligent machines with the interests of man. This would largely reduce the rising extinction risk fear among humans.

- Individuals must begin to make special efforts to relearn new skills, apart from the one they already have, as this move will increase their chances of remaining relevant with fresh AI innovations expected to come into play in the future.
- There is an urgent need to fast track the quick endorsement and implementation of the 23 Asimolar principles proposed by researchers from FLI in 2017. These authors believe this action would facilitate the entrenchment of desired sanity in the field of AI research and implementation.

## References

1. Hawking, S., Tegmark, T., Russell, S., & Wilczek, F. (2014). Transcending complacency on super-intelligent machines. Hoffpost. Online publication. [http://www.huffingtonpost.com/stephen-hawking/artificial-intelligence\\_b\\_5174265.html](http://www.huffingtonpost.com/stephen-hawking/artificial-intelligence_b_5174265.html).
2. Wogu, I. A. P. (2011). *Problems in mind: A new approach to age long problems and questions in philosophy and the cognitive science of human development* (pp. 495). Pumack Nigeria Limited Education Publishers. ISBN 978-978-50060-7-0.
3. Kasparove, G. (1996). The day I sensed a new kind of intelligence. *Time Magazine*.
4. Poe, E. A. (1977). Maeizel's chess-playing machines. *Southern Literary Messenger*. (April 1836) Reprinted in the portable Poe, eds. Philip Van Doren Stern (pp. 511–512). New York: Penguin Books.
5. Bryant, M. (2014). Artificial intelligence could kill us all. Meet the man who takes that risk seriously. The Next Web (TNW). Online publication. [https://thenextweb.com/insider/2014/03/08/ai-could-kill-all-meet-mantakes-risk-seriously/#.tnw\\_hVchaHqU](https://thenextweb.com/insider/2014/03/08/ai-could-kill-all-meet-mantakes-risk-seriously/#.tnw_hVchaHqU).
6. RileyMar, T. (2017). Artificial intelligence goes deep to beat humans at poker. Online publication. <http://www.sciencemag.org/news/2017/03/artificial-intelligence-goes-deep-beat-humans-poker>.
7. Griffin, A. (2015). AI could wipe out humanity when it gets too clever. Independent News. Online publication. <http://www.independent.co.uk/life-style/gadgets-and-tech/news/stephen-hawking-artificial-intelligence-could-wipe-out-humanity-when-it-gets-too-clever-as-humans-a6686496.html>.
8. Russell, S. (2015). This artificial intelligence pioneer has a few concerns. *Quanta Magazine*. Online publication. <https://www.wired.com/2015/05/artificial-intelligence-pioneer-concerns/>.
9. Tegmark, M. (2016). Benefits and risks of artificial intelligence. *Future of Life*. Online publication. <https://futureoflife.org/background/benefits-risks-of-artificial-intelligence/>.
10. Wallace, H. (2017). *The Future of Artificial Intelligence in the Poker World*. CASINO GAMS PRO. Online publication. <http://www.casinogamespro.com/2017/10/16/the-future-of-artificial-intelligence-in-the-poker-world>.
11. ALPAC. (1966). Languages and machines: Computers in translation and linguistics. A report by the Automatic Language Processing Advisory Committee, Division of Behavioral Sciences, National Academy of Sciences, National Research Council. Washington, D.C.
12. Light-hill's Report (1973). This report was commissioned by the Science Research Council (SRC) to give an unbiased view of the state of AI research primarily in the UK in 1973. The two main research groups were at Sussex and Edinburgh.
13. Marquart, S. (2017). Aligning super intelligence with human interests. *Future of Life*. Online publication. <https://futureoflife.org/2017/07/18/aligning-superintelligence-with-human-interests/>.
14. Wogu, I. A. P., Olu-Owolabi, F. E., Assibong, P. A., Apeh, H. A., Agoha, B. C., & Sholarin, M., et al. (2017). Artificial intelligence, alienation and ontological problems of other minds: A critical investigation into the future of man and machines. In *Conference paper presented at the 2017 ICCNI/IEEE International Conference on Computing, Networking and Informatics*.

- proceedings of the IEEE International Conference on Computing, Networking and Informatics (ICCNI 2017)*, pp. 29–31 October, 2017, Covenant University, Ota.
15. Azuela, J. H., Gerhard, R., Jean, S., & Cortés, U. (2005). *Advances in artificial intelligence theory*. Research on computer science, published by the Center for Computing Research of IPN, Vol. 16.
  16. Huffington Post UK (2014) Artificial intelligence poses ‘extinction risk’ to humanity’, says Oxford University’s Stuart Armstrong. An online publication of Huffington Post UK. [http://www.huffingtonpost.co.uk/2014/03/12/extinction-artificial-intelligence-oxford-stuart-armstrong\\_n\\_4947082.html](http://www.huffingtonpost.co.uk/2014/03/12/extinction-artificial-intelligence-oxford-stuart-armstrong_n_4947082.html).
  17. Andrews, S. A. (2014). Alexander Peysakhovich’s theory on artificial intelligence. *Pacific Standard Magazine*. Online publication (2017). <https://psmag.com/magazine/alexander-peysakhovich-30-under-30>.
  18. Davey, T. (2017). Artificial intelligence and the future of work: An interview with Moshe Vardi. *Future of Life*. Online publication. <https://futureoflife.org/2017/06/14/artificial->.
  19. Mészáros, I. (1970). *Marx’s Theory of Alienation*. Online publication. <http://www.marxist.org/archive/meszaros/works/alien/>.
  20. Ollman, B. (1976). Alienation: Marx’s conception of man in capitalist society. Online publication. <http://www.alienationtheory.com/>.
  21. Cox, J. (1998). An introduction to Marx’s theory of alienation. (79) 5 *International Socialism*: Quarterly Journal of the Socialist Workers Party (Britain) Published July 1998 Copyright © International Socialism. (1998).
  22. Cohen, L., Manion, L., & Morison, K. (2000). *Research methods in education*. London: Routledge Falmer.
  23. Marilyn, K. (2013). Ex-post facto research: Dissertation and scholarly research, Recipes for success. Dissertation Success LLC, Seattle, WA. <http://www.dissertationrecipes.com/wp-content/uploads/2011/04/Ex-Post-Facto-research.pdf>.
  24. Derrida, J. (1976). Of Grammatology. Baltimore: Johns Hopkins University Press.
  25. Balkin, J. M. (1987). *Deconstructive practice and legal theory*, 96 Yale L.J.
  26. Derrida, J. (1992). Force of law: Deconstruction. Translated by Mary Quaintance, eds.
  27. McCorduck, P. (2004). Machines who think. 25th Anniversary Edition; RAQ Online. [http://www.pamelamc.com/html/machines\\_who\\_think.html](http://www.pamelamc.com/html/machines_who_think.html).
  28. Lawhead, F.W. (2003). A case for artificial intelligence, In F. W. Lawhead (Ed), *Philosophical journey: An interactive approach* (2nd edn.). McGraw Hill.
  29. Vashisht, M. (2017). How is artificial intelligence changing the manufacturing industry in *Artificial Intelligence*. Online Publication. <http://www.ishir.com/blog/4654/artificial-intelligence-in-manufacturing-industry.htm>.
  30. Conn, A. (2017). Can we properly prepare for the risks of super-intelligent AI? Future of Life Institute (FLI). Online publication. <https://futureoflife.org/2017/03/23/ai-risks-principle/>.
  31. AMPT. (2014). Artificial intelligence applications in manufacturing. Advance MP Technology. <http://www.advancedmp.com/artificial-intelligence/>.
  32. Dvorsky, G. (2010). Can we build an artificial superhuman intelligence that won’t kill us? *FitBeatz*. Online publication. <http://www.fitbeatz.com/can-we-build-an-artificial-superintelligence-that-wont-kill-us/>.
  33. Muehlhauser, L., Koch, C., & Russell, S. (2014). On machine super-intelligence. Machine Intelligence Research Institute (MIRI). Online publication of MIRI. <https://intelligence.org/2014/05/13/christof-koch-stuart-russell-machine-superintelligence/>.
  34. Gouldner, A. W. (1984). *The two Marxisms* (pp. 177–198). New York: Oxford University Press.
  35. Kasanoff, B. (2014). If ‘humans need not apply,’ Will all our jobs disappear? In the *Little Black Book of Billionaire Secrets*. Online publication. <https://www.forbes.com/sites/brucekasanoff/2014/08/18/if-humans-need-not-apply-will-all-our-jobs-disappear/#4611bb3e47ba>.
  36. Brynjolfsson and McAfee (2014). *The second machine age: Work, progress, and prosperity in a time of brilliant technologies*. New York, London: W.W. Norton and Company.

37. Armstrong, S. (2014). Artificial intelligence poses ‘extinction risk’ to humanity says Oxford University’s Stuart Armstrong, in *Huffington Post UK*, An online publication of Huffington Post UK. [http://www.huffingtonpost.co.uk/2014/03/12/extinction-artificial-intelligence-oxford-stuart-armstrong\\_n\\_4947082.html](http://www.huffingtonpost.co.uk/2014/03/12/extinction-artificial-intelligence-oxford-stuart-armstrong_n_4947082.html).
38. Marcus, G. (2013). Why we should think about the threat of Artificial Intelligence. The New Yorker. <http://www.newyorker.com/tech/elements/why-we-should-think-about-the-threat-of-artificial-intelligence>.
39. Sapin, E. (2017). How AI and IoT must work together. Venture Beat. FacebookTwitterLinkedInGoogle + EmailPrint.

# Exploring Ensembles for Unsupervised Outlier Detection: An Empirical Analysis



Akanksha Mukhriya and Rajeev Kumar

## 1 Introduction

Outlier detection is the process of identifying those rare observations which deviate significantly from the rest of data. Outliers are not only the resultant of some measurement or other types of errors, but are those interesting patterns also which provide important insights about the data. In fact, in real-world scenario, outliers are mainly known as source of meaningful information about the unusual behavior in the data, e.g., network anomaly detection, fraud detection.

Variety of methods have been proposed in the literature for outlier detection, e.g., k-NN distance-based [3], density-based LOF [4], those specifically designed for high-dimensional data [5–8]. All these methods employ so disparate approaches that, practically, there is no particular winner approach for outlier detection for all or majority of datasets. For example, on many high-dimensional datasets, even the trivial distance-based average k-NN method outperforms those giving specific treatment to high-dimensional data, e.g., SOD [8], ABOD [7]. Similarly, SOD [8] sometimes outperforms average k-NN and LOF [4] methods on low-dimensional datasets as well. Certainly, it depends and varies according to dataset. Nonetheless, selection of a suitable outlier detection method for a given dataset is enormously challenging. Hence, for a given dataset, be it any, it is good to consider outlier results of different detection methods for further analysis, which leads to ensemble learning for outlier detection.

Ensemble learning has been widely used for various knowledge discovery problems such as classification and clustering. On the contrary, outlier detection

---

A. Mukhriya (✉) · R. Kumar (✉)

School of Computer and Systems Sciences, Jawaharlal Nehru University,

New Delhi 110067, India

e-mail: akankshamukhriya@gmail.com

R. Kumar

e-mail: rajeevkumar.cse@gmail.com

ensembles have caught attention in recent years only [1, 2, 9, 10]. Moreover, the goal of classification ensembles is to get more robust results than the constituents, whereas clustering ensembles are used to get multiple insights into data using multiple sets of clustering results.

On the other hand, due to highly imbalanced-data, the task of outlier detection is very sensitive to the prediction accuracy of even one outlier point, and in fact, detecting only one new outlier is also worthwhile. This makes ensemble learning for outlier detection quite dissimilar and more fruitful than the traditional ensembles. In practice, outlier detection ensembles are mainly intended to get a number of new true outliers from each component detector in the ensemble. However, robustness of result is also enhanced in outlier ensembles by increasing the confidence level of outliers resulted by most detectors.

Aggarwal [1] has discussed the fit or unfit of ensemble learning methods for classification and clustering to outlier detection. Zimek et al. [2] have presented the possible challenges and open questions for outlier ensembles. Complementary to their points, in this paper, we analyze and discuss various aspects of unsupervised outlier detection ensembles, e.g., member selection, score unification, result combination. We perform empirical tests to understand and analyze the associated issues and challenges and to explore meaningful insights as well. In addition, we illustrate the importance of outlier ensembles by means of more empirical analysis.

This paper is organized as follows. In the next section, we discuss various aspects of member selection and some models with related issues. Section 3 focuses on methods for combination of outlier results and associated issues. In Sect. 4, we focus on emphasizing the importance of unsupervised outlier detection ensembles. Section 5 concludes the paper.

## 2 Member Selection

### 2.1 Accessing Accuracy and Diversity of Detectors

For an ensemble, an outlier detector should be accurate as well as diverse enough from the other detectors in the ensemble. It is suggested to maintain a trade-off between accuracy and diversity of members [2, 11]. Diverse but inaccurate or less accurate detectors will mislead the detection results, and accurate but non-diverse detectors will result in a similar set of outliers, which in turn leads to an unfruitful ensemble.

Note that, in unsupervised scenario, due to the absence of ground truth, accuracy and diversity of a detector cannot be accessed directly. Yet, a limited number of attempts has been made to address such challenging task in the literature. For accuracy evaluation, correlation with union of top-k results of all detectors [12], or with average score vector [13], and element-based majority voting approach [13] is used. For diversity evaluation also, the correlation between different score vectors [12] is used

in the literature. Although all these measures are not promising enough, for instance, union of top-k outliers for accuracy evaluation is highly sensitive to inaccurate or bad detectors at input. However, besides the above, no other accessing measure is available for this purpose.

## 2.2 Selection Models

Using accessing measures discussed in the previous subsection, a few selection models have been proposed in recent past, e.g., greedy selection [12], vertical selection, horizontal selection [13], etc. Greedy selection [12] uses union of top-k outliers of all input detectors as a target vector for accuracy evaluation. Detector with the highest correlation to target is added first to ensemble. Then, detectors are picked one by one based on their lowest correlation to current ensemble to maximize diversity. If the detector in question enhances the correlation of the ensemble to target, it gets selected, rejected otherwise. Vertical selection [13] focuses on accuracy and uses average of all normalized score vectors as target. Detectors which increase correlation of ensemble with average score vector are selected.

Horizontal selection [13], unlike the previous two, is an element-based approach. It first uses majority voting to get a set of target or outliers. Then, order statistics is used to compute the probability that for an outlier point, a given ordering of ranks across detectors is generated by a null model. Detectors giving more than expected scores to most of these anomalies are rejected; rest get selected.

Greedy selection [12] focuses more on diversity, in target as well (due to union of top-k), and while picking detectors as well. Vertical selection focuses entirely on accuracy. Similarly, horizontal selection also focuses more on accuracy, while the chances of getting diversity are limited to majority voting only. However, as suggested in the literature also [2, 11], both should be considered into account in a balanced way. None of these methods focuses on both of these aspects well together, though it is equally challenging as well to have a proper balance of bias–variance trade-off.

To clearly understand the selection quality, we evaluate these models for several ensembles on four benchmark datasets in Fig. 1, whose details are given in Table 1. For ann-thyroid and spam-base datasets, we use not-normalized and without duplicates variants in version 4 given by Campos et al.<sup>1</sup> We use the Satimage dataset from UCI repository,<sup>2</sup> where from class 4, we have picked 250 out of 626 observations as outliers randomly. For optical digits dataset, we have picked 571 points from class 1 as normal and 100 points from class 0 as outliers. For selection, we pick one from each family, i.e., greedy selection [12] as diversity oriented and vertical selection [13] as accuracy oriented. Figure 1 shows all the input detectors for selection with their correlation plot. For score unification, z-score normalization is used. From Fig. 1,

---

<sup>1</sup><http://www.dbs.ifi.lmu.de/research/outlier-evaluation/DAMI/>.

<sup>2</sup>[https://archive.ics.uci.edu/ml/datasets/Statlog+\(Landsat+Satellite\)](https://archive.ics.uci.edu/ml/datasets/Statlog+(Landsat+Satellite)).

we can analyze that how well the accuracy or diversity measure of the two selection models is performing. Some of the key observations are given as follows:

- Greedy selection performs poorly when diverse detectors are less accurate or inaccurate. For example, in Fig. 1d, clearly three highly diverse detectors are: LOF[k = 1](0.4795), LOF[k = 5](0.4455), and SOD[k, l = 10](0.5657). Since the first two are quite inaccurate, the greedy selector picks the SOD detector. On the contrary, the vertical selector picks one of the accurate detector from the candidates, i.e., Average k-NN [k = 10] due to its higher correlation to average score vector.
- Vertical selection results in more accurate ensembles in some cases (Figs. 1a, d), but due to no diversity concern, it is unable to pick many accurate but diverse detectors (Fig. 1e), hence, the required accuracy diversity trade-off is not maintained.
- For the majority of the cases, LOF and SOD detectors show significant diversity. For example, ensembles in Figs. 1b, c, d, e more or less in all cases, the lower correlations between LOF and SOD detectors is clearly shown by comparatively darker cells. Although, there can be exceptions, and it may vary from dataset to dataset. However, it is good to be considered to induce diversity between the candidate detectors at input.

Overall, greedy selection is sensitive to less accurate or inaccurate detectors at the selection input. In such cases, the target vector, which is union of top-k outliers of all, comprises of many false positives. This in turn deteriorates selection quality. On the other side, vertical selection targets on accuracy only, but less diverse detectors make the ensemble unfruitful.

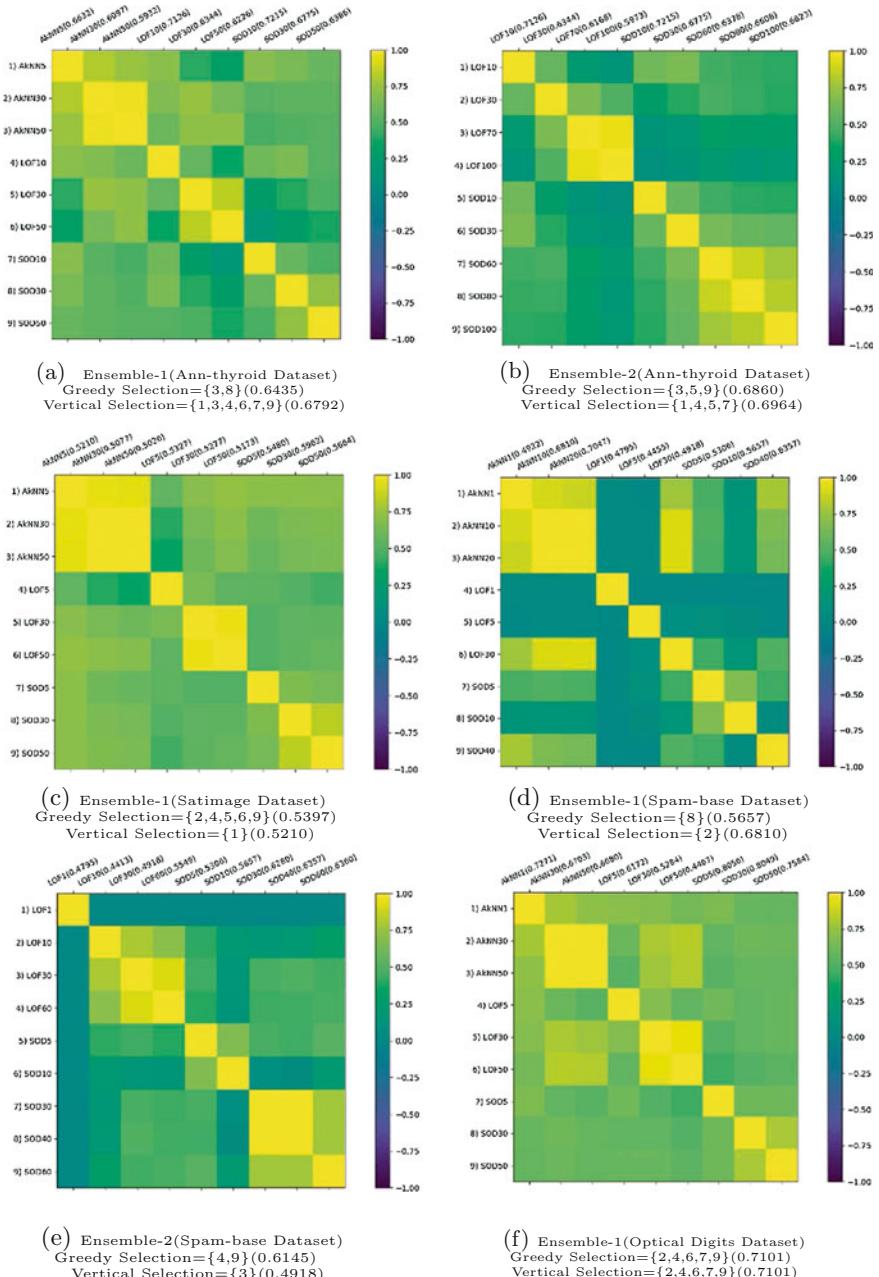
### **2.3 Issues and Challenges of Member Selection**

In the context of member selection for outlier detection ensembles, a number of issues and research questions have already been discussed in [1, 2]. In addition, some other issues we observed, and challenges for future research we see, are given as follows:

- A comprehensive analysis on diversity between different detectors based on their characteristics would probably lead to a better understanding on the issues of diversity.  
To develop better diversity measures for outlier scores, as suggested in [2].
- To develop selection methods which target both accuracy as well as the diversity of detectors would lead to better selection quality. Though, the challenge is to maintain an accuracy-diversity trade-off in the absence of ground truth.

## **3 Combination**

Despite a good set of (i.e., diverse but accurate) selected detectors for an ensemble, it is equally challenging to combine them effectively. Here, by effective combination,



◀Fig. 1 Correlation plots between several input detectors for ensemble member selection. Lighter cells show higher correlation between detectors, and darker cells show lower correlation and higher diversity. Accuracy information (ROC AUC) of detectors is given in brackets at horizontal axis. The selected detectors (numbered at vertical axis) along with combined ensemble result (ROC AUC) using average function are given below in each plot

**Table 1** Summary of datasets used in Fig. 1

Dataset	Instances	Attributes	Outliers (%)	Outlier versus normal class
Ann-thyroid	6942	21	347 (5%)	Non-inlier versus inlier images
Satimage	6059	36	250 (4.1%)	Damp gray soil versus others
Spam base	4207	57	1679 (39.9%)	Spam versus non-spam emails
Optical digits	671	64	100 (14.9%)	0 versus 1

we mean that how well the quality outliers by each member brought together in a single ensemble result.

### 3.1 Combination of Outlier Ranks

Rank-based combinations use the order of outlier scores to facilitate their comparability and combination, e.g., Kemeny–Young voting method [14], robust rank aggregation (RRA) [15]. However, ranking equally treats all similar rankers of different detectors, neglecting their possible outlying probabilities by individual detectors. This is not desirable, since, for a given  $k$ , not all top- $k$  results of each detector are probable outliers. Although, on the contrary, the significance of relative differences of outlier scores is also debatable.

### 3.2 Combination of Outlier Scores

Ranks are known for their crude behavior as they do not provide any information about outlying probability of a point or between subsequent points. This is why scores have been used mostly for combination of various outlier results. Moreover, with scores, there are precombination issues as well, which we discuss in the following subsections.

### 3.2.1 Score Unification

Outlier scores by heterogeneous measures result in huge variations in their ranges and scales hence are not directly comparable. If these raw scores are combined, then detectors with higher range scores would dominate the ensemble. In fact, same detectors with different parameter values have this issue as well. Therefore, all these score vectors should be unified to lead a meaningful score combination. Normalization is conventionally used for this purpose to transform all of them in a common range [0, 1], e.g., linear scaling and z-score normalization.

Besides, few methods specific to outlier score unification have been proposed in the literature which are specifically designed for outlier score unification. The first such approach calculates posterior probabilities of observations to be outlier [16] and assumes that both inliers and outliers follow the same score distribution. The other approach [16] used a mixture model for likelihood probabilities of each observation to both normal and outlier classes. Nguyen et al. [17] used scaling by absolute deviation from mean for each score. Kriegel et al. [18] discussed statistical inference for different outlier detection methods and thus proposed specific scaling methods to them, e.g., Gaussian scaling, gamma scaling [18]. One of their objectives is to provide significant contrast between normalized scores of outliers and inliers in data.

However, with any of the above scaling methods, the primary goal is to avoid domination of detectors with higher range scores during combination. Additionally, the purpose of using scores is that, for a given detector, their relative score differences convey the relative outlying probability of points. This can be considered as an objective during outlier score normalization as well, as done in [16] and [18].

### 3.2.2 Score Combination Methods

Two combination functions are commonly used for score combination in outlier ensembles: average and maximum. The average is known as a low-risk and low-reward function, whereas maximum is known as a high-risk and high-reward function [1]. Other than average and maximum, a few of their variants are also proposed by Aggarwal and Sathe [11], i.e., average of maximum, maximum of average. In average of maximum, maximum is first applied on different groups or bins of members and then finally combined using average. Similarly, in maximum of average, average is first used in each bin, and then, final combination is done using maximum. Both of these methods are a bit better, if ensembles are big enough

### **3.3 Issues and Challenges of Combination**

In the context of combining results of several outlier detectors, a number of issues and research questions have already been discussed in [1, 2]. In addition, some other issues we observed, and challenges for future research, are given as follows:

- Zimek et al. [2 Sect. 5.4] have discussed the issue of calibration of scaling methods, i.e., “can scores actually be converted into outlier probabilities”? Adding to that, we pose a question that if normalized scores are not the outlying probabilities, then is it fair to consider detectors and their outliers with lower normalized scores inferior to those having higher normalized score values?
- Improved scaling methods which will provide some contrast between outlier and normal scores, but with good score comparability across the ensemble, within the  $[0, 1]$  range, can contribute to a more effective ensemble combination.
- Should choice of a combination function also consider base detectors of that ensemble? For example, we observe empirically that despite averaging in average k-NN detectors, they usually dominate the others even after normalization, particularly with higher k-values. Therefore, using maximum for combination in such cases will lead to enhance the domination more and will result in unfair or ineffective combination.
- Since, average of maximum and maximum of average [1] are suitable for larger ensembles to have bins, so can a hybrid approach be designed for combination, which can integrate the strengths of both of these well, and is suitable for smaller ensembles too?

## **4 Significance of Outlier Detection Ensembles**

In this paper, one of our objectives is to highlight the usefulness of ensembles for unsupervised outlier detection. Moreover, we believe that outlier ensembles are highly useful, as finding a very small number of outliers, even one only, may lead to high impact actions in real-world applications like fraud detection, anomaly detection. To understand, we illustrate commonness and diversity between members of various ensembles using top-k results of each for different datasets. Our prime motive by such illustration is to signify that how much more powerful it is when the constituents come together in ensembles than individually. Note that, by powerful we do not mean the overall ROC AUC improvement, but finding more confident and new outliers, particularly in top-k and  $m * \text{top-}k$  ( $m$ : number of ensemble members) of an ensemble. The details are given below.

**Table 2** Summary of datasets used in Fig. 2

Dataset	Instances	Attributes	Outliers (%)	Outlier versus normal class
Credit card fraud dataset	19492	30	492 (2.5%)	Fraudulent versus normal transactions
Credit card fraud dataset	19492	29	492 (2.5%)	Fraudulent versus normal transactions
Ann-thyroid	6942	21	347 (5%)	Non-inlier versus inlier images
Spam base	4207	57	1679 (39.9%)	Spam versus non-spam emails
German credit risk dataset	1000	24	300 (30%)	Bad or good credit risks

#### 4.1 Dataset and Ensemble Description

We perform an empirical analysis on four benchmark datasets for different outlier ensembles. Their brief is given in Table 2. The credit card fraud detection dataset from Kaggle,<sup>3</sup> we use a smaller subset of data with 15,000 normal and 492 fraudulent transactions, having 30 attributes as Version1. For version 2, we have removed time attribute from version 1 dataset.

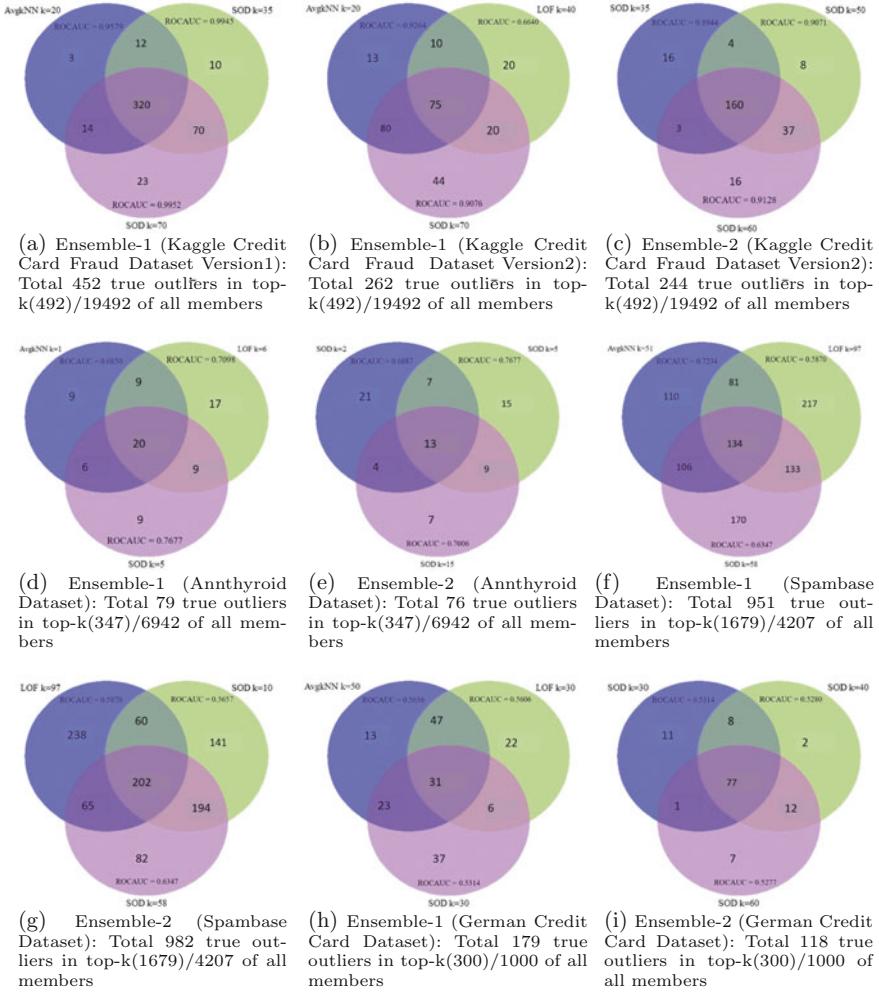
For Statlog's German credit risk dataset,<sup>4</sup> we skip the use of cost matrix for simplicity. For ann-thyroid and spam-base datasets, we use not-normalized and without duplicates variants in version 4 given by Campos et al.<sup>5</sup> Then, for outlier detection, we use three prominent methods from different families of detectors in the literature, namely average k-NN, local outlier factor (LOF) [4], and subspace outlier degree (SOD) [8].

In unsupervised scenario, the common practice is to pick top-k outlier points by any detection method, and we do the same here. To analyze top-k results, we use Venn diagram to represent the number of common as well as diverse outliers by each ensemble member. To avoid representation complexity of Venn diagrams, we limit ensemble sizes in Fig. 2 to three detectors, which represent the base detectors of various ensembles on datasets given in Table 2.

<sup>3</sup><https://www.kaggle.com/dalpozz/creditcardfraud>.

<sup>4</sup>[https://archive.ics.uci.edu/ml/datasets/statlog+\(german+credit+data\)](https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data)).

<sup>5</sup><http://www.dbs.ifi.lmu.de/research/outlier-evaluation/DAMI/>.



**Fig. 2** Number of diverse as well as common true outliers in top-k between all members of the ensemble. These Venn diagrams represent that how diverse are the ensemble base detectors from each other, whereas the number of true outliers indicates that how accurate are the top-ks of these members

## 4.2 Observations

We summarize key observations from ensembles in Fig. 2 as follows:

- For all ensembles in Fig. 2, the top-k results of members collectively consist of more number of true outlier points than any of the components. For example, ensemble in Fig. 2b contains a total of 262 true outlier or fraud transactions in top-492 (total number of fraud transactions in the dataset) points from all members,

whereas the members individually contain 178 (average k-NN), 125 (LOF), and 219 (SOD) true fraud transactions in their top-429(k). Certainly, it implies that ensemble members are resulting in good diversity. For above example also, out of these 262 outliers, there are 13, 20, and 44 diverse outliers resulted by the three base detectors, respectively.

- Heterogeneous detectors in the ensemble usually result in more number of new or diverse outliers. For instance, ensemble in Fig. 2b consists of top-k of three different detection methods, namely average k-NN, LOF, and SOD consists of 77 ( $= 13 + 20 + 44$ ) diverse outliers and 262 in total. Besides, ensembles in Fig. 2h, d, f are also with heterogeneous detectors, resulting in a good number of diverse outliers.

Note that, despite less or moderate accuracy of some ensemble members, heterogeneity brings in diversity. For instance, in Fig. 2b, the LOF detector is the least accurate one, whereas the other two members are comparatively much more accurate. Yet, the LOF detector introduces 20 outliers new to the other two in its top-k. Similarly, the least accurate LOF detector (0.5870) in Fig. 2f introduces maximum number of diverse outliers (217) in top-k.

- Homogeneous detectors with distant k-NN values result in better ensemble diversity. For example, for ensemble of all SOD detectors in Fig. 2c, SOD detectors with  $k, l = 50$  and  $k, l = 60$  have 37 true outliers in common, whereas SOD detector with  $k, l = 35$  has only 4 and 3 outliers in common with SOD[ $k, l = 50$ ] and SOD[ $k, l = 60$ ] detectors respectively. Clearly, more common directly imply less diverse results, and  $k, l = 35$  is at better gap with 50 and 60. Besides, ensembles in Fig. 2i, e implying similar inference. Though, there can be exemptions to this inference.
- Not only diverse but even common outliers get benefit of the ensemble. The reason is when individual detectors come together in an ensemble, the common true positives or outliers, either between all of them, or between few or majority of them, would be labeled as more confident or probable outliers.

Overall, the empirical tests and observations imply that outlier detection ensembles usually strengthen the results of its constituents, both in terms of robustness by enhancing the confidence level of common or majority outliers and in terms of diversity by garnering new outliers. This is true despite the presence of a few less or moderately accurate detectors. However, in some cases, the overall ensemble performance (say ROC AUC) may be less than some of its constituents. Yet, in those cases too, the  $m * \text{top-}k$  outliers ( $m$ : number of ensemble members) would probably consist of more number of new as well as confident outliers than its individual members. This, in true sense, is the power of an outlier detection ensemble.

## 5 Conclusion

Ensemble learning for outlier detection gathers the goodness of characteristics of various detection methods in order to provide a handful of new outliers. However, the power of an outlier detection ensemble depends on its constituents. As much diverse but accurate the constituents, that much effective is the ensemble.

Complementary to the details on unsupervised outlier ensembles, presented by Aggarwal [1] and Zimek et al. [2], in this paper, we empirically analyze various aspects of member selection, unification, and combination for unsupervised outlier detection ensembles. We discuss several new issues and research questions related to each of these aspects. We then accentuate comprehension of the potential of an unsupervised outlier ensemble through empirical analysis, which is driven by the fact that finding even one new outlier is worthy due to their rare occurrences. All of the observations and analysis imply that ensembles for outlier analysis seem to be promising to reinforce the quality of outlier detection.

**Acknowledgements** The authors acknowledge financial support from University Grants Commission under the scheme University with Potential for Excellence during the course of this work. One of the authors also acknowledges CSIR for the fellowship assistance.

## References

1. Aggarwal, C. C. (2013). Outlier ensembles: Position paper. *ACM SIGKDD Explorations Newsletter*, 14(2), 49–58.
2. Aggarwal, C. C., & Sathe, S. (2015). Theoretical foundations and algorithms for outlier ensembles. *ACM SIGKDD Explorations Newsletter*, 17(1), 24–47.
3. Aggarwal C. C., & Yu, P. S. (2001). Outlier detection for high dimensional data. *ACM Sigmod Record*, 30, 37–46. ACM.
4. Breunig, M. M., Kriegel, H. P., Ng, R. T., & Sander, J. (2000). Lof: Identifying density-based local outliers. *ACM Sigmod Record*, 29, 93–104. ACM.
5. Gao, J., & Tan, P. N. (2006). Converting output scores from outlier detection algorithms into probability estimates. In *6th IEEE International Conference on Data Mining (ICDM)* (pp. 212–221). IEEE.
6. He, Z., Deng, S., & Xu, X. (2005). A unified subspace outlier ensemble framework for outlier detection. In *Advances in Web-Age Information Management* (pp. 632–637).
7. Keller, F., Muller, E., & Bohm, K. (2012). Hics: High contrast subspaces for density-based outlier ranking. In *28th IEEE International Conference on Data Engineering (ICDE)* (pp. 1037–1048). IEEE.
8. Kemeny, J. G. (1959). Mathematics without numbers. *Daedalus*, 88(4), 577–591.
9. Kriegel, H. P., & Zimek, A. et al. (2008) Angle-based outlier detection in high-dimensional data. In *14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 444–452). ACM.
10. Kriegel, H. P., Kröger, P., Schubert, E., & Zimek, A. (2008). Outlier detection in axis-parallel subspaces of high dimensional data. In *Advances in Knowledge Discovery and Data Mining* (pp. 831–838).
11. Kolde, R., Laur, S., Adler, P., & Vilo, J. (2012). Robust rank aggregation for gene list integration and meta-analysis. *Bioinformatics*, 28(4), 573–580.

12. Kriegel, H. P., Kroger, P., Schubert, E., & Zimek, A. (2011). Interpreting and unifying outlier scores. In: *SIAM International Conference on Data Mining* (pp. 13–24). SIAM.
13. Lazarevic, A., & Kumar, V. (2005). Feature bagging for outlier detection. In *11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining* (pp. 157–166). ACM.
14. Nguyen, H., Ang, H., & Gopalkrishnan, V. (2010). Mining outliers with ensemble of heterogeneous detectors on random subspaces. In *Database Systems for Advanced Applications* (pp. 368–383). Springer.
15. Ramaswamy, S., Rastogi, R., & Shim, K. (2000). “Efficient algorithms for mining outliers from large data sets. *ACM Sigmod Record*, 29, 427–438. ACM.
16. Rayana, S., & Akoglu, L. (2015). Less is more: Building selective anomaly ensembles with application to event detection in temporal graphs. In *SIAM International Conference on Data Mining* (pp. 622–630). SIAM.
17. Schubert, E., Wojdanowski, R., Zimek, A., & Kriegel, H. P. (2012). On evaluation of outlier rankings and outlier scores. In *SIAM International Conference on Data Mining* (pp. 1047–1058). SIAM.
18. Zimek, A., Campello, R. J., & Sander, J. (2014). Ensembles for unsupervised outlier detection: challenges and research questions a position paper. *ACM SIGKDD Explorations Newsletter*, 15(1), 11–22.

# **Part IV**

# **Web Intelligence**

**Dr. Sushma Nagpal Section Editor**

## **Editorial**

The Web has become a ubiquitous tool to perform numerous activities such as sharing information, filtering information, conducting businesses, enabling education, executing e-commerce. Web intelligence entails a confluence of research areas which aim at developing a knowledge-intensive Web offering intelligent services to perform all such activities efficiently, adaptively, and in a scalable manner. They include knowledge management, ontologies, information retrieval, language technology and human-computer interaction. The research works in this part encompass these areas and strive to distil more power from the Web.

Rai et al. perform a quantitative analysis of the performance of different machine learning algorithms for detecting type III metaphors. Their experiments reveal that, given a large data set, conventional approaches such as support vector machines and logistic regression perform at par with alternative techniques such as neural networks, in terms of both accuracy and precision-recall balance. Garg and Lobiyal propose a multi-class classification of sentiments in Hindi sentences by tapping both language-dependent and language-independent features with fuzzy neural networks. The authors present a new term weighting scheme that combines term frequencies with their Hindi SentiWordNet scores which give promising results when tested on Hindi movie and tourism corpora created by IIT Bombay.

Sharma et al. present a semi-supervised learning model using generative adversarial networks to identify the language of the transliterated text in the Roman script which conveys meaning in Hindi. The authors demonstrate reliable results of their semi-supervised approach when compared with the state-of-the-art Google language detector.

Recommender systems have stepped in to resolve the problem of information overload that makes it difficult to find the right information from the morass of Web data. Jain and Dixit propose a slew of similarity metrics to alleviate the data sparsity problems in context-aware recommender systems. These metrics adapt to local and global ratings, common rating proportions, user rating preferences and contextual similarity. The authors illustrate their suitability for highly sparse data with due

consideration for contextual information, but also caution against dependence on data set for Eskin measure.

Kumar and Gupta present an empirical study and analysis of supervised learning techniques on a benchmark credit card transaction data set, for fraud detection in online transactions. They evaluate the performance of logistic regression, nearest neighbours, linear and RBF SVMs, decision trees, random forest and naïve Bayes and find the best performance coming from logistic regression. Mittal and Mishra propose a trust and reputation-based model to tackle denial-of-service attacks in mobile agent systems. The authors show that their model is effective in the protection of agents to the platform as well as the platform to agent types of denial-of-service attacks.

### **Section Reviewers:**

Akshi KumarAmita Jain

Anil Goel

Anita Singrova

Bhawna Gupta

Deepak Kumar Sharma

Divya Choudhary

Gagandeep Kaur

Geeta Rani

Kavita Pandey

Mala Saraswat

Mukta Goyal

Neetu Sardana

Payal Khurana

P. Raghu Vamsi

Poonam Bansal

Pradeep Atrey

Priti Bansal

Rama Krishna Challa

Rahul Katarya

Rohit Beniwal

S. K. Dhurandher

Sunny Rai

Sushama Nagpal

Srishti

Tapan Kumar Das

Upasana Pandey

Vikas Maheshkar

# Effect of Classifiers on Type-III Metaphor Detection



Sunny Rai, Shampa Chakraverty and Ayush Garg

## 1 Introduction

Metaphors are a fascinating component of human language which encapsulates a bundle of information. A metaphor is generated when a comparison is made between two disparate domains with no apparent resemblance between them. Traditionally, a metaphor has been viewed as an poetic device employed for vividness and distinction in literary text. Lakoff and Johnson contradicted this view and stated that it is not only a literary phenomenon but a property of thought, that is, a cognitive phenomenon [1]. The term *conceptual metaphor* was coined to describe it.

This view postulates that metaphor is not limited to just similarity-based meaning extensions of individual words, but rather involves reconceptualization of a whole area of experience in terms of another. Thus, a metaphor always involves two concepts or conceptual domains: the *target* (also called *topic* in linguistics literature) and the *source* (also known as *vehicle*). For example, “AFFECTION IS WARMTH” is one such conceptual mapping. Its one of the many manifestations is “He is a cold person.” Here, we are conveying the *withdrawn and introvert nature* of the subject with the metaphor *cold*. The utility of a metaphor to achieve a optimal transfer of information in human language is indisputable. Recently, we observed a notable interest in metaphor processing [2–7] from researchers of natural language processing and cognitive domain. One of many applications of metaphor processing is machine translation.

---

S. Rai (✉) · S. Chakraverty  
Division of Computer Engineering, Netaji Subhas Institute of Technology,  
New Delhi, Delhi, India  
e-mail: post2srai@gmail.com

S. Chakraverty  
e-mail: apmabs.nsit@gmail.com

A. Garg  
Amazon Development Centre India Pvt. Ltd., Hyderabad, India  
e-mail: ayushgarg@hotmail.co.uk

A system capable of recognizing and interpreting metaphorical language in open text is an indispensable component of real-world NLP applications such as information retrieval (IR) [8, 9], machine translation (MT) [10, 11], and opinion mining [12, 13]. Due to the absence of metaphor processing component, these applications often fail to interpret metaphorical component in text correctly. For example, the Hindi translation for the sentence “My lawyer is a shark” is “mera vakeel shaark hai.” Similarly, for the phrase “dark thoughts,” Google Translate<sup>1</sup> generates the interpretation “andhere vichaar.” For another phrase, “My car drinks gasoline,” we obtain syntactically incorrect translation “meree kaar petrol gaisoleen.” In all of these cases, Google Translate tries to provide literal interpretations of the given phrases instead of their metaphorical senses. This leads to incorrect translations in the target language, that is, Hindi from the source language, English.

In the late 1980s, Nunberg categorized metaphors into two categories, namely *dead metaphors* and *novel metaphors* [14]. As the terminology implies, dead metaphors are overly exploited mappings in daily parlance and thus adopt the metaphorical meaning as an extended literal interpretation. One such example is the usage of the word *sweet* in the phrase “sweet child.” The word, *sweet*<sup>2</sup> initially meant ‘*having or denoting the characteristic taste of sugar*’ or ‘*having a high residual sugar content*’ but at present, its meaning extends to ‘*having a sweet nature befitting an angel or cherub*’ which is a metaphorical interpretation. Dead metaphors are considered equivalent to literal text as one does not need to perform any mental mapping to interpret it. In contrast, novel metaphors are newly generated mappings and require common-sensical knowledge of concepts involved in the mapping to extract contextually coherent metaphorical interpretation.

Later, Krishnakumaran and Zhu classify the metaphors into three categories on the basis of their pattern of occurrence [6]. The three categories are *Type-I*, that is, Subject-Object (SO) metaphors, *Type-II* which are also known as Subject-Verb-Object (SVO) metaphors and *Type-III*, that is, Adjective-Noun (AN) metaphors. In this paper, we analyze the third category, i.e., AN metaphors where an adjective is used metaphorically to highlight certain attributes of the noun it modifies. Let us take a pair of sentences to illustrate the concept of AN metaphors.

The reason behind deaths is *burning charcoals* in the closed spaces. (a)

She has a *burning desire* to practice law. (b)

In sentence (a), the adjective *burning* is used in literal sense to indicate the combustion of charcoals. However, the same adjective is used metaphorically to convey the urgent and passionate desire to pursue law by the subject in sentence (b). The adjective *burning* helps in visualizing the urgent state of her desire and thus, it can be said that “PASSION IS FIRE.”

Prior works employ the notion of relative abstractness [15, 16] and word embeddings [17, 18] to detect Type-III metaphors in text. For example, in the phrase *burning desire*, the metaphor *burning* is a relatively imageable and concrete concept than

---

<sup>1</sup>Google Translate: <https://translate.google.co.in/>

<sup>2</sup>WordNet search(*sweet*): <https://goo.gl/8cxvns>

*desire*. In addition, techniques such as random forest [17] and neural network [18] have been used for Type-III metaphors. However, as the No Free Lunch Theorem suggests, there is no universally best learning algorithm. Therefore, in this paper, we analyze the efficacy of different algorithms over a common dataset to understand their relevance for detecting Type-III metaphors.

The rest of the paper is organized as follows. We provide a brief overview of prior works on Type-III metaphor detection in Sect. 2. We perform the critical analysis of existing features sets and different techniques in Sect. 3. We conclude in Sect. 4.

## 2 Related Work

In this section, we discuss prior works pertinent to Type-III metaphor detection. Krishnakumaran and Zhu utilize hyponymy relations from WordNet to identify metaphorical adjectives [6]. They use Web IT corpus to create a set of literal instances through the concept of conditional probability which serves as the base to match common hyponymy relation. Lakoff and Johnson postulate that a metaphor is created when an abstract concept is mapped to a concrete concept to facilitate the process of communication [1]. Turney et al. in [15] and Neuman et al. in [16] use the same notion of relative abstractness between the target word and its context of usage in their work. They contend that by analyzing the abstractness, it is plausible to segregate metaphorical usages from the literal usages.

Tsvetkov et al. [17] use a combination of features such as supersense, abstractness and word embeddings to identify metaphorical instances [17]. They train their model on a dataset of 1768 <adjective, noun> pairs and test on a test dataset of 200 instances. Shlomo and Last use a set of different features such as *abstractness*, *cosine similarity* and *corpus statistics* by extracting <adjective, noun> phrases from Reuters corpus [19]. Gutiérrez et al. [20] train a distributional model using positive pointwise mutual information on a corpus of 4.58 billion tokens. For each adjective in the dataset, they separate its literal instances from the metaphorical ones to model the literal and metaphorical sense of a candidate adjective. They test their approach on a test dataset of 8592 AN phrases. Recently, Bizzoni et al. use a single layer neural model trained on pretrained vector embeddings, namely *word2vec* [21], GloVe [22], and Levy-Goldberg embeddings [23] to gauge the metaphoricity of a given AN phrase by providing a probability measure between 0 and 1 [18]. They train their model over the dataset introduced in [20].

We also observe that word embeddings such as *word2vec* have been successfully utilized to predict abstractness. The authors in [20] also outline the observation that their model basically learns concrete-abstract polarity while segregating metaphorical phrases from literal ones. Furthermore, *word2vec* embeddings are a reliable indicator to gauge the relatedness between two concepts due to its inherent philosophy of distribution hypothesis [4, 24]. Therefore, we use only pretrained *word2vec* embeddings as feature set while training different classifiers for identifying adjective metaphors.

**Table 1** Datasets for type-III metaphors

Dataset	#Adjectives	Metaphor	Literal	Total
Tsvetkov et al. [17]	405	884 + 100	884 + 100	1768 + 200
Gutiérrez et al. [20]	23	4601	3991	8592
Dataset3	409	5485 + 100	4875 + 100	10360 + 200

### 3 Effect of Classifiers on Type-III Metaphor Detection

For our study, we select five different techniques namely linear support vector machine (LSVM) [25], radial support vector machine (RSVM), logistic regression (LR), random forest (RF), and single-layer neural network (NN).

#### 3.1 Datasets

To the best of our knowledge, there are two publicly available datasets for Type-III metaphors provided by Tsvetkov et al. [17] and Gutiérrez et al. [20]. The details about these datasets are given in Table 1. The corpus was designed such that every adjective in the dataset can be easily classified as metaphor or not metaphor by a human without requiring a larger context.

From Table 1, we observe that Tsvetkov et al. dataset has a total 1768 training instances with 200 instances for testing. It comprises 405 different adjectives. However, Gutiérrez et al. [20] dataset has only 23 different adjectives but 8592 samples in the dataset. Thus, it provides a reasonable number of cases to model each <adjective, noun> pair. We use both of the datasets to conduct our analysis. We further combine both datasets to create a third dataset, say *Dataset3* and test the efficacy of different models on it.

#### 3.2 Comparison

We use Python v2.7 and Scikit Library [26] to implement the classifiers for detection of Type-III metaphors. We use the default values of tuning parameters in case of LSVM, RSVM, LR, and RF. We use the first architecture of neural network proposed by Bizzoni et al. in [18]. We use TensorFlow [27] and Keras [28] for implementing the neural network. The feature set comprises of concatenated pretrained *word2vec* 300-dimensional embeddings for the parsed (adjective, noun) pairs.

We performed tenfold cross-validation on each dataset. In ten-fold cross-validation, we partition the dataset into ten parts. Out of ten parts, nine parts are used for training the model and testing is performed on the remaining fold. We evaluated the

performance of the trained classifiers using the metrics *accuracy*, *precision*, *recall*, and *F-score* which are used widely in the literature on metaphor detection. Let  $N_o$  be the total number of instances in the test dataset,  $TP$  be the true positive cases,  $TN$  be the true negative cases,  $FP$  be the false positive cases, and  $FN$  be the false negative cases. Then, *Accuracy* is defined as  $\frac{TP+TN}{N_o}$ . *Precision* is defined as  $\frac{TP}{TP+FP}$ , whereas *Recall* is defined as  $\frac{TP}{TP+FN}$ . *F-score* is defined as  $\frac{2 * precision * recall}{precision + recall}$ . We compute precision, recall, and f-score separately for metaphor and literal classes.

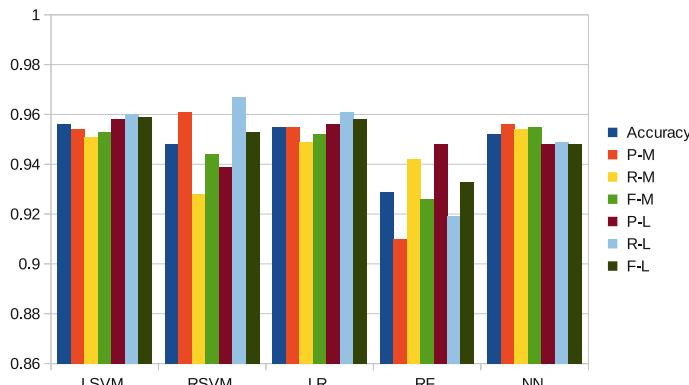
In Table 2, we provide results obtained on Tsvetkov et al. dataset. We observe that LR provides the best accuracy of 81.4% and RSVM provides the best F-score of 81.5% for metaphorical class. The RF model performs the worst with an accuracy of 76.4%. Overall, models namely LSVM, RSVM, LR, and NN, show comparable performance with respect to accuracy. We illustrate the performance of models using a bar plot in Fig. 1. We observe that LSVM, LR, and NN models provide a consistent and comparable performance with regard to every metric, whereas RSVM and RF models have significant variations in the values obtained for different metrics.

In Table 3, we provide results obtained on Guterrez et al. Dataset. We observe a significant improvement, that is, up to 14% in accuracy of models in comparison with models trained on Tsvetkov et al. [17] dataset. This may be attributed to the increased sample size and thus sufficient instances to attune the model parameters effectively. The LSVM model provides the best accuracy of 95.6% followed by LR model. The

**Table 2** Results on Tsvetkov et al. [17] dataset

Technique	A	P-M	R-M	F-M	P-L	R-L	F-L
LSVM	0.805	0.814	0.796	0.80	0.797	0.814	0.805
RSVM	0.806	0.784	<b>0.852</b>	<b>0.815</b>	<b>0.836</b>	0.761	0.7950
LR	<b>0.814</b>	<b>0.823</b>	0.805	0.813	0.806	<b>0.824</b>	<b>0.814</b>
RF	0.764	0.741	0.821	0.778	0.794	0.705	0.746
NN	0.807	0.799	0.814	0.806	0.816	0.799	0.807

Legend A Accuracy; P Precision; R Recall; F-F Measure; M Metaphor; and L Literal

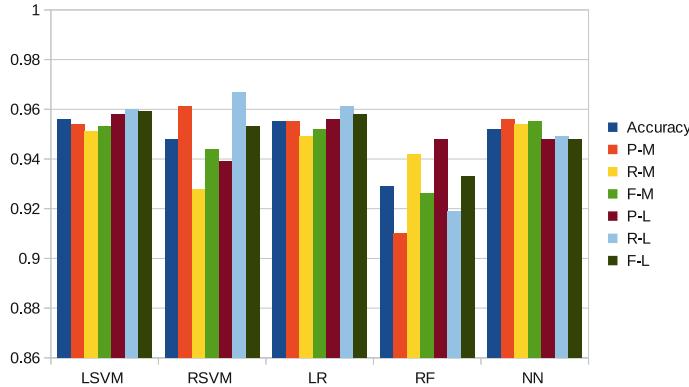


**Fig. 1** Performance on Tsvetkov et al. [17] dataset

**Table 3** Results on Gutiérrez et al. [20]

Technique	A	P-M	R-M	F-M	P-L	R-L	F-L
LSVM	<b>0.956</b>	0.954	0.951	0.953	<b>0.958</b>	0.960	<b>0.959</b>
RSVM	0.948	<b>0.961</b>	0.928	0.944	0.939	<b>0.967</b>	0.953
LR	0.955	0.955	0.949	0.952	0.956	0.961	0.958
RF	0.929	0.91	0.942	0.926	0.948	0.919	0.933
NN	0.952	0.956	<b>0.954</b>	<b>0.955</b>	0.948	0.949	0.948

Legend A Accuracy; P Precision; R Recall; F-F Measure; M Metaphor; and L Literal

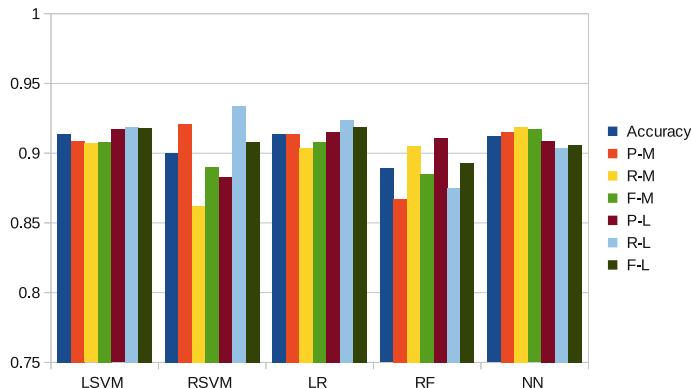
**Fig. 2** Performance on Gutiérrez et al. [20] dataset**Table 4** Results on Dataset3

Technique	A	P-M	R-M	F-M	P-L	R-L	F-L
LSVM	<b>0.914</b>	0.909	0.907	0.908	<b>0.917</b>	0.919	0.918
RSVM	0.9	<b>0.921</b>	0.862	0.890	0.883	<b>0.934</b>	0.908
LR	<b>0.914</b>	0.914	0.904	0.908	0.915	0.924	<b>0.919</b>
RF	0.889	0.867	0.905	0.885	0.911	0.875	0.893
NN	0.912	0.915	<b>0.919</b>	<b>0.917</b>	0.909	0.904	0.906

Legend A Accuracy; P Precision; R Recall; F-F Measure; M Metaphor; and L Literal

NN model performed the best if we consider the F-score metric for metaphor class. The RF model again showed the worst performance with an accuracy of 92.9%. We plot the performance of different models using a bar graph in Fig. 2. From Fig. 2, we observe a similar pattern in the performance of metrics as we observed in the case of models trained on Tsvetkov et al. [17] also illustrated in Fig. 1.

In Table 4, we provide results obtained on the combined dataset, i.e., *Dataset3* which has 10,360 training samples and comprises of 409 different adjectives. We observe a dip of around 4% in the accuracy of every model in comparison with models trained with Gutiérrez et al. [20] dataset. This may be attributed to lack of sufficient instances in the training dataset from Tsvetkov et al. [17] dataset to model the behavior of every adjective. However, we again observe that the performance



**Fig. 3** Performance on Dataset3

of metrics is similar to the models trained on Gutiérrez et al. [20] dataset. We illustrate the performance metrics in Fig. 3. Further, we observe that the traditional machine learning techniques such as LSVM and LR perform better than NN and RF with respect to many parameters. The LSVM and LR models provided the best accuracy of 91.4% which is 4.2% lower than the best accuracy obtained by LSVM model trained on Gutiérrez et al. [20] dataset. The RF model performed the worst performance with an accuracy of 88.9%.

## 4 Conclusion

In this paper, we performed a comparison among different machine learning techniques on two publicly released datasets for Type-III metaphor detection. Our study illustrates that the results obtained using traditional machine learning techniques such as support vector machine and logistic regression are at par with techniques such as neural network. We note that techniques, namely linear SVM, logistic regression, and neural network provide an optimal balance between recall and precision in comparison with techniques such as radial SVM and random forest. We also observed that a large dataset for training improves the performance of an algorithm irrespective of its type.

## References

1. Lakoff, G., & Johnson, M. (2008). *Metaphors we live by*. Chicago: University of Chicago press.
2. Rai, S., Chakraverty, S., & Tayal, D. K. (2016). Supervised metaphor detection using conditional random fields. In *Proceedings of the Fourth Workshop on Metaphor in NLP*, (pp. 18–27). USA: Association of Computational Linguistics.

3. Rai, S., Chakraverty, S., & Tayal, D. K. (2017). Identifying metaphors using fuzzy conceptual features. In *Proceedings of the International Conference on Information, Communication and Computing Technology*. Berlin: Springer.
4. Rai, S., & Chakraverty, S. (2017). Metaphor detection using fuzzy rough sets. In *International Joint Conference on Rough Sets*, (pp. 271–279). Berlin: Springer.
5. Rai, S., Chakraverty, S., Tayal, D. K., & Kukreti, Y. (2017). Soft metaphor detection using fuzzy c-means. In *International Conference on Mining Intelligence and Knowledge Exploration*, (pp. 402–411). Berlin: Springer.
6. Krishnakumaran, S., & Zhu, X. (2007). Hunting elusive metaphors using lexical resources. In *Proceedings of the Workshop on Computational Approaches to Figurative Language*, (pp. 13–20), 26 April 2007. New York: Association for Computational Linguistics.
7. Rai, S., Chakraverty, S., Tayal, D. K. & Kukreti, Y. (2018). A study on impact of context on metaphor detection. *The Computer Journal*.
8. Croft, W. B., Metzler, D., & Strohman, T. (2010). *Search engines: Information retrieval in practice* (Vol. 283). USA: Addison-Wesley Reading.
9. Manning, C. D., Raghavan, P., & Schütze, H. et al. (2008). *Introduction to information retrieval* (Vol. 1). Cambridge: Cambridge university press Cambridge.
10. Bahdanau, D., Cho, K., & Bengio, Y., (2014). *Neural machine translation by jointly learning to align and translate*. [arXiv:1409.0473](https://arxiv.org/abs/1409.0473).
11. Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K. et al. (2016). *Google's neural machine translation system: Bridging the gap between human and machine translation*. [arXiv:1609.08144](https://arxiv.org/abs/1609.08144).
12. Wang, S., Chen, Z., & Liu, Bing. (2016). Mining aspect-specific opinion using a holistic lifelong topic model. In *Proceedings of the 25th international conference on world wide web*, (pp. 167–176). International World Wide Web Conferences Steering Committee.
13. Poria, S., Cambria, E., & Gelbukh, A. (2016). Aspect extraction for opinion mining with a deep convolutional neural network. *Knowledge-Based Systems*, 108, 42–49.
14. Nunberg, G. (1987). Poetic and prosaic metaphors. In *Proceedings of the 1987 Workshop on Theoretical Issues in Natural Language Processing*, (pp. 198–201). Association for Computational Linguistics.
15. Turney, P. D., Neuman, Y., Assaf, D., & Cohen Y. (2011). Literal and metaphorical sense identification through concrete and abstract context. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, (pp. 27–29). UK: John McIntyre Conference Centre.
16. Neuman, Y., Assaf, D., Cohen, Y., Last, M., Argamon, & S., Howard, N., et al. (2013). Metaphor identification in large texts corpora. *PloS One*, 8(4), e62343.
17. Tsvetkov, Y., Boytsov, L., Gershman, A., Nyberg, E., & Dyer, C. (2014). Metaphor detection with cross-lingual model transfer. In *52nd Annual Meeting of the Association for Computational Linguistics*, (pp. 248–258), June 22–27, 2014. USA: Association for Computational Linguistics.
18. Bizzoni, Y., Chatzikyriakidis, S., & Ghanimifard, M. (2017). Deep learning: Detecting metaphoricity in adjective-noun pairs. In *Proceedings of the Workshop on Stylistic Variation*, (pp. 43–52).
19. Shlomo, Y. B., & Last, M. (2015). Mil: Automatic metaphor identification by statistical learning. In *DMNLP'15 Proceedings of the 2nd International Conference on Interactions between Data Mining and Natural Language Processing* (Vol. 1410, pp. 19–29). Germany: Aachen.
20. Gutiérrez, E. D., Shutova, E., Marghetis, T., & Bergen, B. (2016). Literal and metaphorical senses in compositional distributional semantic models. *ACL, 1*.
21. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed Representations of Words and Phrases and their Compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, & K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems 26*, (pp. 3111–3119). USA: Lake Tahoe, Nevada.
22. Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, (pp. 1532–1543).

23. Levy, O., & Goldberg, Y. (2014). Neural word embedding as implicit matrix factorization. In *Advances in neural information processing systems*, (pp. 2177–2185).
24. Firth, J. R. (1957). *The Technique of Semantics. Papers in Linguistic Theory 1934–1951.* London: Oxford University Press (Reprinted, first published 1957).
25. Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3), 273–297.
26. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12, 2825–2830.
27. Abadi, Martín, Barham, Paul, Chen, Jianmin, Chen, Zhifeng, Davis, Andy, Dean, Jeffrey, et al. (2016). TensorFlow: A system for large-scale machine learning. *OSDI*, 16, 265–283.
28. Chollet, F. (2013). Keras: Deep Learning library for python. Convnets, recurrent neural networks, and more. Runs on Theano and TensorFlow. *GitHub repository*.

# Multi-class Classification of Sentiments in Hindi Sentences Based on Intensities



Kanika Garg and D. K. Lobiyal

## 1 Introduction

Sentiment analysis is sub-field of Natural Language Processing and Information Retrieval. Sentiment information is crucial for NLP industry and thus is a non-trivial task to perform. Nowadays, sentiment analysis finds a great variety of applications. This field has provided marketing and research groups a boost. This has showed its usage in advertisements, politics, education and so on.

Earlier, the sentiments have been classified as binary, i.e. either positive or negative. But nowadays, growth in the e-commerce field has demanded sentiments to be more flexible. Online reviews make a strong impact on the services and products offered by various companies. Many reviews make it difficult to choose between either positive or negative. Therefore, other classes have been introduced to further expand the horizons of the classification of sentiments based on intensities. As in Fig. 1 below, the user has both good and bad experiences regarding the product. He liked the sound quality but still he is unhappy with the higher volumes as well as the product he got. Therefore, this sentence cannot be only positive or only negative but it has varied intensity for positivity as well as negativity.

The various classes based on intensities for 5-class classification are weakly positive (wp), highly positive (hp), neutral (nu), weakly negative (wn) and highly negative (hn). Similarly, for 7-class classification, the various classes are weakly positive (wp), moderately positive (mp), highly positive (hp), neutral (nu), weakly negative (wn), moderately negative (mn) and highly negative (hn).

---

K. Garg · D. K. Lobiyal

School of Computer and Systems Sciences, JNU, New Delhi, Delhi, India  
e-mail: kanika97\_scs@jnu.ac.in

D. K. Lobiyal  
e-mail: lobiyal@gmail.com

Top critical review  
[See all 533 critical reviews >](#)

15 people found this helpful

Overall its a great product to have !

By Jeet on 27 May 2016

The sticker of product on box reads Sony MDR-ZX110A but the 'Operating Instructions' manual inside the box reads product name as Sony MDR-ZX110. which makes me believe that I didn't get the product I ordered. But still I loved this one because it's a genuine sony product and it looks pretty awesome. the sound quality is better than my expectation in comparison to other headphones of same price range though at high volumes sound is audible to others too. After 4-5 hours of continuous use it may cause slight pain in your ears. But the seller needs to clarify about the version of the product.

**Fig. 1** Snapshot of an Amazon review

Prevailing techniques in opinion mining can be divided into two groups: classified feature-based methods and sentiment knowledge-based methods. As Hindi is a resource scarce language, it is essential to unfold a language independent learning algorithm to model the features which expect less linguistic intuition and feature selection process.

Intensity-based classification in more than three classes has not been done for Hindi sentences before. Therefore, to the best of our knowledge, this work is the first to take up the classification of Hindi sentences into 5-classes or 7-classes. The fuzzy approach is aptly able to handle multi-class classification as a review can partly be in positive class or negative class. Therefore, fuzzy sets are necessary to classify the sentences into multiple classes.

The paper is organised as follows: Sect. 2 includes the related work done in sentiment classification. Section 3 includes various feature metrics used in this work. The proposed method is included in Sect. 4. Section 5 includes the experimental results and discussions. Conclusion and future work is included in Sect. 6.

## 2 Related Work

Over a decade, there is a continuous research going on in the area of sentiment classification for Indian Languages. Hindi is the most commonly spoken language in India. Therefore, there is an inclination towards Hindi language for researchers as the web contents are also increasing in Hindi. Web 2.0 has given Hindi a boost, and thus, more people are able to use the internet in Hindi language. The researches that have been done in the recent past in the area of Hindi language processing are discussed below.

Joshi [1] had developed annotated corpora of Hindi movie reviews. Then they classified various Hindi sentences using on machine learning, machine translation and Hindi-SentiWordNet. They also developed sentiment-labelled corpus and lexical resource, i.e. Hindi-SentiWordNet for Hindi movie review domain [2]. They

performed sentence-level sentiment classification using fuzzy theory with maximum membership principle. Balamurali [3] has done a document-level classification using semantic space. It is a supervised sentiment classifier developed based on WordNet senses. Bakliwal [4] created a subjective lexicon containing adverbs and adjectives with their polarity scores using Hindi WordNet and annotated corpora of product reviews. They used a graph-based method to add synonym and antonym relations to the initial seed list. With this enhanced lexicon and product reviews corpora, they have been able to achieve approximately 79% accuracy.

Bakliwal et al. [4] used a graph-based method to build a subjective lexicon using WordNet. In this work, a small seed list of words having sentimental importance has been prepared. For each word, their synonyms, antonyms and their sentiment scores have been determined using WordNet. Antonyms and synonyms are added in the graph with relevant nodes using edges. This system has achieved 74% accuracy on reviews data set. Jha et al. [5] created an opinion mining system and named as Hindi Opinion Mining System (HOMS). In this system, they had used supervised machine learning method Naïve Bayes and part-of-speech tagging for unsupervised opinion mining. They have achieved an accuracy of 87%. Ramarkhiyani et al. [6] performed the comparative analysis of word2vec and JoBimText. JoBimText present words and phrases with vectors of very high dimensions based on their co-occurrences, whereas the word2vec applies neural distributed approach which uses neighbouring words for context.

Pang and Lee [7] introduced rating inference problem. In their work, sentiments are categorised into multi-point scale using meta-algorithms. In 2015, Liu et al. [8, 9] performed multi-class multi-label sentiment classification. In this paper, total eleven states of art classification methods had been compared and evaluated using eight evaluation metrics. This has been performed on two data sets taken from micro-blogs. Cui [10] integrated distributed semantic features of fixed-sized word-sequence and POS sequence together. Since this feature set is independent of language, this method can be applied to any language. In this work, best performance has been achieved using bag-of-n-Gram with  $n = 3$ . Gaspar [11] presented a qualitative assessment of sentiments in reviews and reactions obtained from social media. This work emphasised on human thinking and psychology. Tripathy [12] carried out sentiment classification using n-gram machine learning techniques like maximum entropy, Naïve Bayes, support vector machines and stochastic gradient descent. In the paper, they concluded that for higher values of  $n$  in n-gram approach, performance of the classifier decreases. Machine learning methods provided the best performance with unigrams and bigrams but with trigrams, four-grams and five-grams, the accuracy degraded. Li et al. [13] in their paper introduced a novice approach towards multi-label sentiment classification. They introduced multi-label maximum entropy model to classify various emotions in a short text. Their results are comparable with the results of major baseline methods. Nadali et al. [14] proposed fuzzy semantics to classify product reviews into varied strength of sentiments. They introduced fuzzy sets for eight different data sets of English language. Similar work is proposed by [15] but for Hindi Language. K. Garg and D. K. Lobiyal proposed fuzzy logic for sentiment classifica-

tion of Hindi sentences. They worked on movie review domain and achieved good classification accuracy.

### 3 Feature Engineering

#### 3.1 Weighting Schemes

##### 1. Tf-idf

Term frequency inverse document frequency [16] is used to give importance to a particular term in a document. It is basically the number of times the particular term occurs in a document. This number is then normalised so that bias for the longer documents can be prevented. It is important because in a longer document, a term may occur many times and it is quite possible that the term is not important in opinion mining.

Let  $C$  be a set of classes  $C = \{c_{WP}, c_{MP}, c_{HP}, c_N, c_{HN}, c_{MN}, c_{WN}\}$  and a set of documents  $D = \{d_1, d_2, \dots, d_N\}$ . Every document is given a class label; a new document is classified in one of these classes based upon the information fetched from the labelled documents.

$$Tf - idf_{ij} = tf_{ij} \times idf_i \quad (1)$$

where  $tf_{ij}$  is the number of times the term  $t_i$ , occurred in the document  $d_j$  and  $idf_i$  is the measures of the importance of the term in the document. The tf-idf are normalised as:

$$|tf - idf_{ij}| = \frac{|tf - idf_{ij}|}{max_{ij} |tf - idf_{ij}|} \quad (2)$$

This tends to filter out the commonly occurring terms in a document.

##### 2. Tf-icf

This term frequency inverse class frequency [17] is used to give importance to a particular term in a given class. It is the number of times term occurs in that particular class. icf is calculated as:

$$icf = \log \frac{|C|}{|\{c : t_i \in c\}|} \quad (3)$$

where  $|C|$  is the total no. of classes and  $|\{c : t_i \in c\}|$  is the no. of classes in which the term exists. Then,

$$tf - icf_{ij} = tf_{ij} \times icf_j \quad (4)$$

and its normalised form is:

$$\left| tf - icf_{ij} \right| = \frac{tf - icf_{ij}}{\max_{ij}} \quad (5)$$

$$tf - icf_{ij}$$

This tends to filter out the commonly occurring terms in a class.

### 3. Mutual Information

Mutual information [18] is the measure of mutual dependence between two variables. It is the expected value of pointwise mutual information over all possible outcomes. Given the term  $t_i$  and document  $D_j$ , the mutual information is defined as:

$$MI(t_i, D_j) = \log \frac{P(t_i, D_j)}{P(t_i) \times P(D_j)} \quad (6)$$

where

$$P(t_i, D_j) = \frac{n_c(t_i)}{|D|} \quad (7)$$

and

$$P(t_i) = \frac{\sum_k n_k(t_i)}{|D|} \quad (8)$$

or this can be written as:

$$MI(t_i, d_j) = \log \frac{n_c(t_i) \times |D|}{\left\{ \sum_k n_k(t_i) \right\} \times n_c} \quad (9)$$

$n_c(t_i)$  = number of c class documents that contain the term  $t_i$

$D_j$  = set of documents in a certain class

$|D|$  = number of documents in training corpus. Then the weight is calculated as follows:

$$TW_{MI}(t_i) = \max\{MI(t_i, D_1), MI(t_i, D_2) \dots MI(t_i, D_k)\} \quad (10)$$

where k is the number of classes.

### 4. $X^2$ (chi-square)

CHI statistics [19] measures the independence between random variables. This can be implemented on term  $t_i$  and set of document  $D_j$  of a particular class  $c_k$ . This measure can be derived using a contingency table (Table 1).

**Table 1** Contingency table for chi-square

Term $t_i$	Exists in $c_k$	Doesn't exists in $c_k$
$T_i$ present	$N_{11}$	$N_{12}$
$T_i$ not present	$N_{21}$	$N_{22}$

So,

$$\chi^2(t_i, D_j) = \frac{|D| \times \{N_{11} \times N_{22} - N_{12} \times N_{21}\}}{(N_{11} + N_{12}) \times (N_{21} + N_{22}) \times (N_{11} + N_{21}) \times (N_{12} + N_{22})} \quad (11)$$

Then the term weighting based on Eq. (11) is given by

$$TW_{CHI}(t_i) = \max\{CHI(t_i, D_1), CHI(t_i, D_2), \dots, CHI(t_i, D_k)\} \quad (12)$$

where k is the number of classes.

### 3.2 Proposed Term Weighting Scheme

In this work, a movie corpus has been taken from IIT-B. In this corpus, distinguished sets are present for negative reviews and positive reviews. Therefore, in the present work all the words present in the negative data set have been labelled as negative and terms occurring in positive data set are taken as positive.

To accomplish the task of sentiment analysis, two methods of term weighting are proposed in this work which is as follows:

1. First method is based on term frequency. Based upon their frequencies of the words, they have been given a score as:

$$Score_w = \begin{cases} \frac{\text{frequency in negative set}}{\text{total occurrences}}, & \text{frequency in negative set} \geq \text{frequency in positive set} \\ \frac{\text{frequency in positive set}}{\text{total occurrences}}, & \text{frequency in negative set} < \text{frequency in positive set} \end{cases} \quad (13)$$

In Eq. 13, preference is given to the negativity in this work because negative reviews are more important in comparison to positive reviews. Negative reviews are the basis for any new developments into the products and services. They are the key threats in the competitive market.

2. Another scheme that has been used is the product of the  $Score_w$  calculated in Eq. 13 and the score of a word in Hindi-SentiWordNet. In this scheme, POS tagging has been done first, then only verbs, nouns, adverbs and adjectives are retained. Then each word is searched in HSWN. If any word is missing in the

H SWN, it is translated to English using Google Translator and then the score is fetched from SentiWordNet. The entire approach is stated below:

- I. All the reviews are converted into sentences.
- II. Then for each sentence, POS tagging is done.
- III. Words for the tags nouns, verbs, adverbs and adjectives are retained and rest of all words are discarded.
- IV. Each word is searched in H SWN
  - a. If the word is present in H SWN then its score is obtained.
  - b. If the word is not present in H SWN, it is converted to English using Google Translator and then score from SentiWordNet is obtained.
- V. For each word, senti-weight is calculated as below:

$$senti - weight_w(n) = \frac{frequency\ in\ negative\ set}{total\ occurrences} \times negative\ score\ of\ word\ in\ SentiWordNet \quad (14)$$

$$senti - weight_w(p) = \frac{frequency\ in\ positive\ set}{total\ occurrences} \times positive\ score\ of\ word\ in\ SentiWordNet \quad (15)$$

Then,

$$senti - weight_w = max(senti - weight_w(n), senti - weight_w(p)) \quad (16)$$

### 3.3 Features Used

#### 1. N-grams

N-grams are one of the most popular features used for sentiment analysis. In this method,  $n$  terms are taken together which helps in classifying documents into various classes. When only one term is taken at a time, it is known as unigrams. Unigrams are used to estimate likelihood of a word to occur in a document based on its frequency of occurrence. Two terms taken together are called as bigrams in which likelihood of a word occurring in the context of other word is taken and then trigrams in which context of previous two words are taken into consideration. Bigrams and trigrams are able to capture contextual information.

#### 2. POS tags

Part-of-speech tagging is done to exclude irrelevant part of the sentences. A sentence formation includes several POS but for our work, only nouns, adjectives, adverbs and verbs are of importance. Here, POS tagger used is created by IIT-B to tag Hindi sentences.

## 4 Method

Movie reviews [1] and Tourism corpora [20] have been taken from IIT-B. Movie Reviews and tourism reviews are differentiated already in positive and negative reviews. Totally 125 positive reviews and 125 negative reviews are present in movie domain data set, and 100 positive and 100 negative reviews are present in tourist reviews. These reviews are then transformed into sentences and are given to five annotators to provide them with the score on scale of  $[-2, 2]$  for five class and on scale of  $[-3, 3]$  for seven class. Finally the tag is taken for which maximum annotators have voted. In case scores of all the annotators are different, then average score is calculated for each sentence and accordingly the label is provided to the sentences. For example, sample sentence (See Appendix) for five-class classification is shown in Table 2.

In the above sentence, maximum of the annotators voted for weakly negative so this sentence has been labelled as weakly negative. In case of tie, one with higher score is considered.

Sample sentence (See Appendix) for 7-class shown in Table 3.

Then features have been fetched using above discussed methods and weights have been assigned. All the scores have been calculated on unigrams and bigrams. We have used supervised machine learning methods for sentiment analysis task.

In this work, fuzzy neural network has been introduced as shown in Fig. 2 below. This is a hybrid system that mimics human thinking and learns in uncertain and imprecise environment. Neural networks are able to recognise patterns in the sentence and are able to adapt to diversifying environments. Fuzzy if-then rules lack the adaptability to deal with changing environment [21, 22]. Therefore, neural network incorporates learning concepts in fuzzy inference system. Also, both exhibit fault tolerance. Deletion of any neuron in a NN or a rule in FIS does not necessarily destroy the system. Instead, the system continues performing because of its parallel and redundant architecture, although performance quality gradually deteriorates.

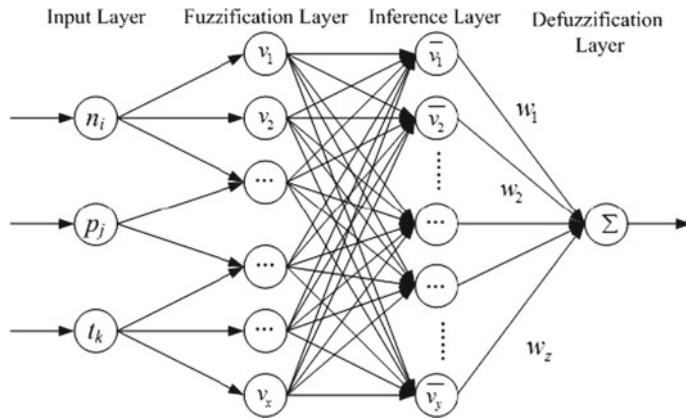
In fuzzy, it is important to correctly choose the membership functions. In this work, we have chosen triangular and semi-trapezoidal membership functions. Semi-trapezoidal functions are used for the corner members like weak negative and weak

**Table 2** Labels provided by 5 annotators for 5-class

Sentence	Annotator1	Annotator2	Annotator3	Annotator4	Annotator5
शायद दर्शकों को उनसे उतना लगाव नहीं रहा	-1	-2	-1	-2	-1

**Table 3** Labels provided by 5 annotators for 7-class

Sentence	Annotator1	Annotator2	Annotator3	Annotator4	Annotator5
फिल्म के किरदार थोड़े हुए नहीं लगते	1	1	2	1	1
फिल्म देखा विवेश में जारदार कमाई कर रही है	3	2	3	2	3
कला की ऐसी दुर्दशा में पहले नहीं देखी थी	-3	-2	-2	-3	-3



**Fig. 2** Structure of fuzzy neural network ([https://www.researchgate.net/figure/261494763\\_fig2\\_Fig-2-The-structure-of-fuzzy-neural-network](https://www.researchgate.net/figure/261494763_fig2_Fig-2-The-structure-of-fuzzy-neural-network))

**Table 4** Sample fuzzy rules

Rule1	If sentiment 1 is strong positive or sentiment 2 is strong positive then sentiment is strong positive
Rule 2	If sentiment 1 is strong negative or sentiment 2 is strong negative then sentiment is strong negative
Rule 3	If sentiment 1 is weak negative and sentiment 2 is weak negative then sentiment is weak negative

positive. The natural language is ambiguous and imprecise; therefore, fuzzy logic can be applied to natural languages. Some sample fuzzy rules are shown in Table 4.

## 4.1 Experimental Set-up

### 4.1.1 Data set

Reviews from movie domain and tourism are taken. Movie domain has 484 labelled sentences including positive and negative reviews. Tourism domain has 100 positive and 100 negative reviews. In this work, tenfold cross-validation is employed where data set is randomly divided into tenfold. Ninefold is used for training purpose and onefold is used for testing. Detailed description of the data set is given in Tables 5 and 6.

**Table 5** Movie reviews

Movie review domain	Reviews	Sentences
Positive	127	258
Negative	125	224

**Table 6** Tourist reviews

Tourist review domain	Reviews	Sentences
Positive	100	256
Negative	100	249

#### 4.1.2 Features

Each sentence is POS tagged, and the stop words are removed. Then unigrams and bigrams are fetched and saved into the unigram and bigrams table with their frequencies. Words occurring less than two times are ignored. Then feature engineering has been done on these sentences by applying all the feature set on them and stored in separate matrices.

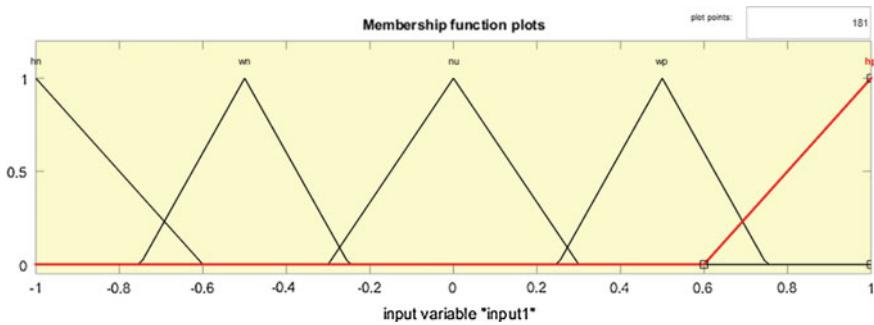
#### 4.1.3 Method

Then these matrices are fuzzified using fuzzy membership function. In our experiment, sugeno fuzzy method is used to fuzzify the features. Then these fuzzified values are fed to the neural networks for training and learning. FNN method is implemented using MATLAB. These feature sets have also been tested on other machine learning methods such as Naïve Bayes, SVM and MaxEnt.

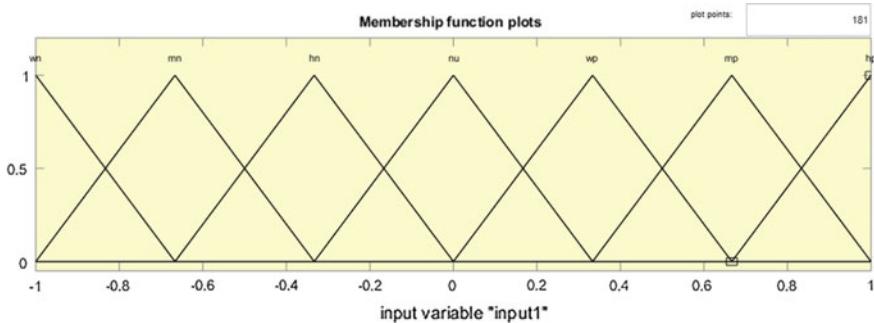
#### 4.1.4 Evaluation Metric

Since our main focus is on the achieving higher accuracy for classifying sentences into different classes, classification accuracy is used as evaluation metric and it is given by:

$$\text{Accuracy} = \frac{\#true\ positives + \#true\ negatives}{\#total\ instances} \quad (17)$$



**Fig. 3** Membership function for 5-class



**Fig. 4** Membership function for 7-class

## 5 Results and Discussions

In this work, experiments are conducted on two domains: movie review domain [1] and tourism domain [20]. These are labelled reviews, and therefore, supervised machine learning methods are used in this work. All sentences have been POS tagged<sup>1</sup> and stop words are removed in the pre-processing step. Various features have been extracted using above discussed methods. Hybrid FNN method has been introduced in this work for multi-class classification. All the feature values are fuzzified and then fed to the neural nets. For fuzzification process, Sugeno method is used. Triangular membership functions are taken for 5-class and 7-class classifications in Figs. 3 and 4.

The results so obtained are compared with the results of existing baselines. Classification accuracy is taken for evaluation. In this work, we have worked on proposed weighting scheme and various other schemes such as tf-idf, tf-ifc, mutual information and chi-square. Various results have been shown in the tables for 5-class classification (Tables 7, 8, 9, 10, 11, 12, 13 and 14).

It is clear from the above given tables for 5-class and 7-class classifications; the proposed hybrid method outperformed the other classification baselines. The

<sup>1</sup><http://www.cfilt.iitb.ac.in/Tools.html>.

**Table 7** Results with FNN method for 5-class classification

Term weighting schemes	Tf-idf (%)	Tf-icf (%)	Chi-square (%)	MI (%)	Score (%)	Senti-weight (%)
Movie reviews	82.7	84.1	82.9	80.7	82.6	89.4
Tourist reviews	79.1	80.8	83.3	81.4	84.2	87.7

**Table 8** Results with Naïve Bayes method for 5-class classification

Term weighting schemes	Tf-idf (%)	Tf-icf (%)	Chi-square (%)	MI (%)	Score (%)	Senti-weight (%)
Movie reviews	86.1	84.2	87.8	85.8	87.6	88.1
Tourist reviews	82.9	81.1	80.4	84.9	85.0	86.4

**Table 9** Results with SVM method for 5-class classification

Term weighting schemes	Tf-idf (%)	Tf-icf (%)	Chi-square (%)	MI (%)	Score (%)	Senti-weight (%)
Movie reviews	85.7	83.8	85.4	87.6	84.8	88.7
Tourist reviews	83.7	80.2	82.2	84.7	84.1	86.8

**Table 10** Results with MaxEnt method for 5-class classification

Term weighting schemes	Tf-idf (%)	Tf-icf (%)	Chi-square (%)	MI (%)	Score (%)	Senti-weight (%)
Movie reviews	86.3	84.8	85.9	86.7	85.1	89.0
Tourist reviews	83.2	81.7	83.3	85.6	85.2	87.4

**Table 11** Results with FNN method for 7-class classification

Term weighting schemes	Tf-idf (%)	Tf-icf (%)	Chi-square (%)	MI (%)	Score (%)	Senti-weight (%)
Movie reviews	80.7	86.1	84.9	78.7	84.6	89.8
Tourist reviews	82.1	83.8	81.3	84.4	86.2	88.7

**Table 12** Results with Naïve Bayes method for 7-class classification

Term weighting schemes	Tf-idf (%)	Tf-icf (%)	Chi-square (%)	MI (%)	Score (%)	Senti-weight (%)
Movie reviews	88.1	80.2	83.8	88.8	85.6	89.1
Tourist reviews	86.9	85.1	81.4	84.9	83.0	87.4

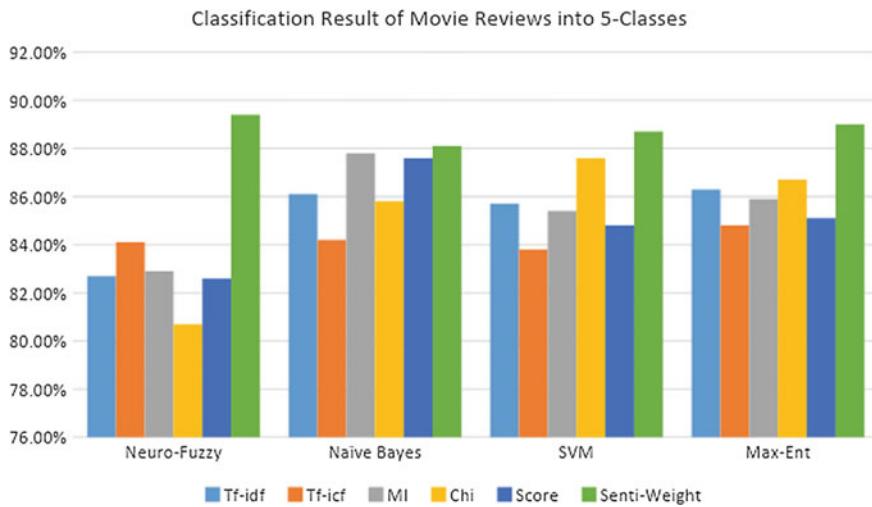
**Table 13** Results with SVM method for 7-class classification

Term weighting schemes	Tf-idf (%)	Tf-icf (%)	Chi-square (%)	MI (%)	Score (%)	Senti-weight (%)
Movie reviews	87.7	81.8	88.4	85.6	86.8	88.7
Tourist reviews	84.7	82.2	83.2	85.7	81.1	86.8

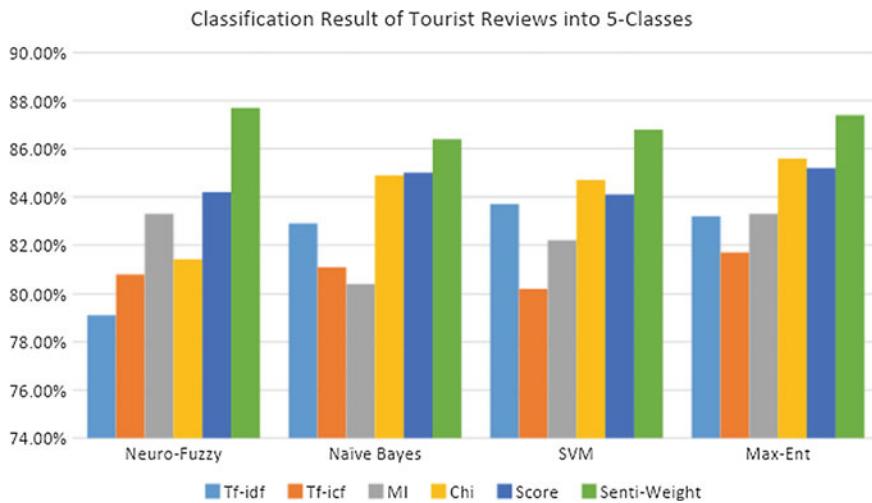
**Table 14** Results with MaxEnt method for 7-class classification

Term weighting schemes	Tf-idf (%)	Tf-icf (%)	Chi-square (%)	MI (%)	Score (%)	Senti-weight (%)
Movie reviews	81.3	88.8	83.9	84.7	87.1	89.0
Tourist reviews	82.2	84.7	86.3	80.6	81.2	87.4

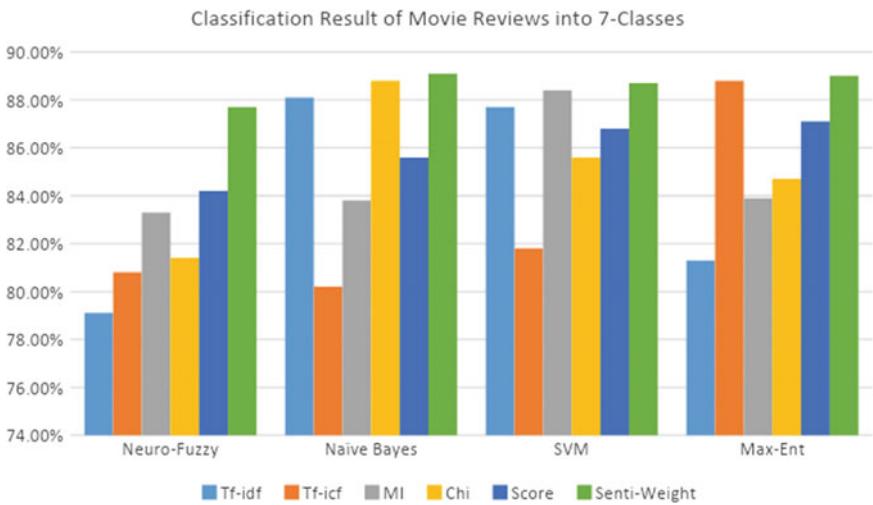
proposed weighting scheme also outperformed other traditional weighting schemes. Senti-weight has shown considerable improvement in the classification as it considers its sentiment score from HSWN along with the frequency. Score taken did not provide good results over other methods as it is based only on frequency. As the data set contains few sentences, this tends to improve in case the sentences are more. The results have been compared graphically as shown below (Figs. 5, 6, 7 and 8):



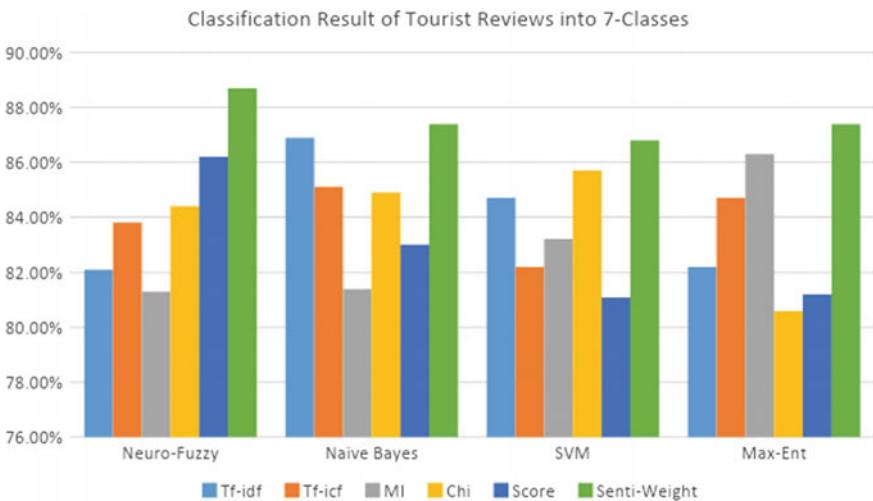
**Fig. 5** Comparative analysis of 5-class classification on movie review domain



**Fig. 6** Comparative analysis of 5-class classification on tourism domain



**Fig. 7** Comparative analysis of 7-class classification on movie review domain



**Fig. 8** Comparative analysis of 7-class classification on tourism domain

## 6 Conclusion and Future Work

In this work, multi-class classification based on intensities of sentiments has been done. On contrary to prevailing sentiment analysis, this is the first attempt for multi-class classification of Hindi sentences to the best of our knowledge. To accomplish this task, different methods have been investigated based on new weighting scheme. FNN has been applied which is based on human thinking and reasoning. We are able

to accomplish the task of multi-class classification of Hindi sentences with good results. This method is compared with different baselines. The proposed approach provided promising results. It outperformed some traditional term weighting schemes and classification techniques.

In future work, some other features can be explored to improve sentiment classification of Hindi sentences. Some language-specific constructs can be introduced to better understand the sentiments conveyed by Hindi language. Work needs to be done to expand the data set as it contains only few sentences. More sentences will lead to better learning of the methods.

**Acknowledgements** This work is supported by CSIR with file no. 09/263(1049)/2015-EMR-I

## Appendix

This section contains all the Hindi sentences included in this paper and their corresponding English translation.

Hindi sentence	Corresponding english sentence
शायद दर्शकों को उनसे उतना लगाव नहीं रहा	Perhaps the audience is not so attached to them
फिल्म के किंदार थोपे हुए नहीं लगते	Movie characters do not look imposing
फिल्म देश विदेश में जांदार कमाइ कर रही है	The film is making huge money abroad
कला की ऐसी दुर्दशा में पढ़ले नहीं देखी थीं	I had not seen such a tragedy of art

## References

1. Joshi, A., Balamurali, A. R., & Bhattacharyya, P. (2010). A fall-back strategy for sentiment analysis in Hindi : A case study. In *Proceedings of 8th International Conference on Natural Language Processing*.
2. Fu, G., Wang, X. (2010). Chinese sentence-level sentiment classification based on fuzzy sets. *Coling2010: Poster*, pp. 312–319.
3. Balamurali, A. R., Joshi, A., & Bhattacharyya, P. (2011). Robust sense-based sentiment classification. In *Proceedings of 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis, ACL-HLT*, pp. 132–138.
4. Bakliwal, A., Arora, P., & Varma, V. (2012). Hindi subjective lexicon: A lexical resource for hindi polarity classification. In *Proceedings of 8th International Conference on Language Resources and Evaluation*, pp. 1189–1196.
5. Jha, V., Manjunath, N., Shenoy, P. D., Venugopal, K. R., & Patnaik, L. M. (2015). HOMS: Hindi opinion mining system. In *Proceedings of 2nd International Conference on Recent Trends in Information Systems*, pp. 366–371.
6. Ramrakhiyani, N., Pawar, S., & Palshikar, G. (2015). Word2Vec or JoBimText? A comparison for lexical expansion of Hindi words. In *Proceedings of 7th Forum for Information Retrieval Evaluation (FIRE)*, pp. 39–42.
7. Pang, B., & Lee, L. (2005). Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of 43rd Annual Meeting of Association for Computational Linguistics*, vol. 3, no. 1, pp. 115–124.

8. Liu, S. M., & Chen, J. (2015). A multi-label classification based approach for sentiment classification. *Expert Systems with Applications*, 42, 1083–1093. <https://doi.org/10.1016/j.eswa.2014.08.036>.
9. Liu, S. M., & Chen, J. (2015). An empirical study of empty prediction of multi-label classification. *Expert Systems with Applications*, 42, 5567–5579. <http://dx.doi.org/10.1016/j.eswa.2015.01.024>.
10. Cui, Z., Shi, X., & Chen, Y. (2016). Sentiment analysis via integrating distributed representations of variable-length word sequence. *Neurocomputing*, 187, 126–132. <https://doi.org/10.1016/j.neucom.2015.07.129>.
11. Gaspar, R., Pedro, C., Panagiotopoulos, P., & Seibt, B. (2016). Beyond positive or negative: Qualitative sentiment analysis of social media reactions to unexpected stressful events. *Computers in Human Behaviour*, 56, 179–191. <https://doi.org/10.1016/j.chb.2015.11.040>.
12. Tripathy, A., Agrawal, A., & Rath, S. K. (2016). Classification of sentiment reviews using n-gram machine learning approach. *Expert Systems with Applications*, 57, 117–126. <https://doi.org/10.1016/j.eswa.2016.03.028>.
13. Li, J., Rao, Y., Jin, F., Chen, H., & Xiang, X. (2016). Multi-label maximum entropy model for social emotion classification over short text. *Neurocomputing*. <https://doi.org/10.1016/j.neucom.2016.03.088>.
14. Nadali, S., Murad, M. A. A., & Mining, A. O. (2012). Fuzzy semantic classifier to determine the strength levels of customer product reviews. In *Proceedings of International Conference on Advances in Computer Science and Applications*, pp. 60–63. 02.csa.2012.01.11.
15. Garg, K., & Lobiyal, D. K. (2018). Sentiment classification of hindi sentences using fuzzy logic. In *Proceedings of 5th International Conference on Computing for Sustainable Global Development*, pp. 3972–3976.
16. Martineau, J., & Finin, T. (2009). Delta TFIDF: An improved feature space for sentiment analysis. In *Proceedings of 3rd ICWSM*, pp. 258–261.
17. Rustamov, S., & Clements, M. (2013). Sentence-level subjectivity detection using neuro-fuzzy models. In *Proceedings of 4th WASSA, ACL*, pp. 108–114.
18. Khan, F. H., Qamar, U., & Bashir, S. (2016). SentiMI: Introducing point-wise mutual information with SentiWordNet to improve sentiment polarity detection. *Applied Soft Computing*, 39, 140–153.
19. Abdel Fattah, M. (2015). New term weighting schemes with combination of multiple classifiers for sentiment analysis. *Neurocomputing*, 167, 434–442. <https://doi.org/10.1016/j.neucom.2015.04.051>
20. Balamurali, A. R., Joshi, A., & Bhattacharyya, P. (2012). Cross-lingual sentiment analysis for Indian languages using linked wordnets. *Proceedings of Coling*, 2012, 73–82.
21. Jang, J. S. R., Sun, C. T., & Mizutani, E. (2014). *Neuro-fuzzy and soft computing*. Prentice-Hall.
22. Zadeh, L. A. (1972). A fuzzy set-theoretic interpretation of linguistic hedges. *Journal of Cybernetics*, 2(3), 4–34.

# Language Identification for Hindi Language Transliterated Text in Roman Script Using Generative Adversarial Networks



Deepak Kumar Sharma, Anurag Singh and Abhishek Saroha

## 1 Introduction

Language Identification (LID) is one of the key problems in any task corresponding to natural language processing (NLP). It being one of the most primitive problems in NLP. For any processing with the text in a unknown language, it is very important to classify which natural language does the text belong to. With the proliferation of Internet in our lives, we as humans have been surrounded by digital media and content in form of text. Content is written using of natural languages. The task, however, becomes complex when there is no knowledge of the script that is being used to convey the meaning of the language. Different languages are more or less complete in the set of phonemes by the virtue of sound their characters make or usually multiple syllables help realize all the phonetics present. Thus, a text can be transliterated meaning it can be written using script of some other language where using words and syllables phonetically similar to mean something totally different in another language. The first step to translate such text is to detect which language does this transliterated text intend to convey. In context of Indian languages and most specifically Hindi the work tries build a model for its identification in transliterated form. Meanwhile, smartphones and computers are becoming close to ubiquitous and majority of communication is done digitally. We use keyboards to generate content and express. Keyboards by default are in Roman script, i.e. the simple 26 English alphabets. Although there are keyboards in Devanagari, they are tedious and cumbersome to use. This has caused people to type their languages in Roman script.

---

D. K. Sharma · A. Singh (✉) · A. Saroha  
Netaji Subhas Institute of Technology, University of Delhi, New Delhi, India  
e-mail: anurags.it@nsit.net.in

D. K. Sharma  
e-mail: dk.sharma1982@yahoo.co.in

A. Saroha  
e-mail: abhisheks.it@nsit.net.in

Were the sentences make little or no sense in Roman script but are phonetically very similar to root language.

Thus, when read out the sentences sound like they are being spoken in native language. In our case of concern more specifically Hindi. Detection of such text and to label them is a challenging and novel area of research. Detection and labelling of text which sounds phonetically similar to the language that it was intended to be written in is challenging because many features that can be used to easily classify a language are lost due to limited scope of different scripts. The area offers an opportunity to explore and apply myriad of concepts ranging from novel information retrieval techniques used in natural language processing to deep learning techniques and architectures in machine learning. The research in this area will also shed light on novel ways to understand phonetics of texts and will help map languages based on phonetics also, thus serving as interesting problem to computational linguists and machine Learning community.

The current state-of-the-art models are either based on statistical calculations or either neural networks-based transliteration models which are strictly supervised learning in nature. This work proposes to use new method which is semi-supervised in nature. Despite the current models being used extensively, the current models still lack satisfaction and have room for improvement. The current neural machine transliteration systems use maximum likelihood estimation (MLE) principle for training the model which means to maximize the probability of a target ground truth sentence given the source sentence. There have been works to work around this. Instead generative adversarial networks can be used for minimizing the distance between the features extracted and features build using the generative Networks. The model aims to extract features from the actual data and also generate its own features using generative networks which are then compared with the original features using the discriminator part of GANs which labels the feature vectors as real or fake. Seeing the feature vectors they actually make a choice about which category they might have been generated from. The two parts of the generative adversarial network play a min-max game with each other, eventually making generator better at generating more real like data to fool the discriminator. The following construct is used to make the artificial neural network train themselves of the particular task in hand as same gradients that flow for error correction in the generator network can flow back into the ANN helping the network to backpropagate its errors. The subsequent sections describe related work. Propose the model and explain related concepts. Finally results are discussed, conclusion is made and paper sets scope for future works.

## 2 Related Works

In this section, the work closely related to our study is reviewed to give a idea of progress. Transliteration is a well-documented problem with many researchers attempting to solve the problem fully. The transliteration of Indian languages has been discussed by scholars since late nineteenth century. A lossless romanization

system for Indic languages was conceptualized when Charles Trevelyan and many other scholars, together in the Transliteration Committee of the Geneva Oriental Congress, in September 1894 agreed for transliteration of Sanskrit test to be done using International Alphabet for Sanskrit Transliteration (IAST) [9]. There have been multiple advancements in language transliteration since then. Devanagari transliteration that is Devanagari to Roman script often referred to as romanization is done using multiple approaches. The transliteration process uses language identification, named entity recognition and then the process of conversion into another language. There have been still no standard system for conversion of Devanagari to Roman script [10]. Malik et al. [8] attempted something that is very different from romanization. They tried to build a system for Urdu–Hindi machine transliteration using statistical models and finite state machines (FSM) which gives an novel idea for rule-based transliteration. Koehn et. al. [6] also give statistical machine translation techniques which deal with the formulation of efficient data formats for translation in general. The use case similar to ours was attempted by the Jia et. al. [5] for Chinese languages. They use n-grams Markov models to achieve the task of English–Chinese back-transliteration. Our model is designed for back-transliteration of English and Hindi and uses deep learning (ANNs) instead. Generative adversarial networks [2, 12] in Ian GoodFellow et al. were shown as novel idea that could easily underline the generator distribution for any task. Since then GANs have been used as foundations for building various architectures around them to solve multiple problems in different domains. They have also proved very successful in the task of face generative, text to image generation to name a few. Generative adversarial networks are also thought to be applied to language translation task [11]. There is very active research for machine translation going on in Google which resulted them being able to detect and translate Web pages making content on Web more accessible to all by putting down language barriers, for example, Britz et al. [1] propose novel work which aims to remove the limitations of neural machine translation (NMT) which is that they are very expensive to train which eventually gives very practical advise based on empirical results they found for English to German translation task. This work will make neural machine translation default standard and ubiquitous for language detection and translation. Therefore, this piece of research aims to contribute to achieving transliteration using generative adversarial networks. The focus is on first part of transliteration that is language detection using generative adversarial networks. Li et. al. [7] have shown that generative adversarial networks can be used in dialogue generation. They modelled this task as a reinforcement learning where two systems, a generative model is used produce response sequences and a discriminator is used to distinguish between the human-generated dialogues and the machine-generated ones, are jointly trained.

### 3 Methodology

The approaches used to tackle the problem solving are based on deep learning techniques primarily neural networks. It can be noted that no domain-specific resources and/or tools were used for development because it was intended that the work must maintain domain independence property. That means the architecture can be easily used for other language pairs provided novel features can be extracted within the text. The latter part of this section describes the approach that was followed in developing each part of the model.

#### 3.1 Language Identification

Language identification concerns itself with identifying the origin of a particular word or a group of words. The language identification can be thought of as a case of classification problem, where input text or a sentence has to be assigned one of the labels denoting the language it is. Hindi being a original to a non-Latin script, the method proposed for language identification is based on supervised machine learning. Here certain features present in the text are extracted to build a feature vector. This vector is propagated into a artificial neural network by means of which detection of the language is done.

#### 3.2 Feature Extraction

The sentences in the corpus are formatted, broken into tokens and checked for certain features which then help the model build a feature vector. There are multiple approaches that are being followed, and various features are used to identify languages. The features used for identification of language are described below:

1. *Character n-gram*: Character n-gram is a continuous sequence of alphabets extracted. Here N in N-gram stands for the number of characters which are taken into consideration for defining the probability of next character. That is the size of a moving window which defines previous characters taken into consideration. The common n-grams that are generated for length one (unigram), two (bigram) and three (trigram). These kinds of features were also used in previous works [10].
2. *First letter capital*: The feature is a binary one. It checks for presence of a capital letter at start of every word in sentence. Words starting with capitals can be good indication of named entities and thus can be independent of language.
3. *Alphanumeric*: While parsing the sentences, such tokens with syllables are looked for which start with a punctuation. For example, { : ; ; } do not belong to any language and must not be used to influence the feature vector.

4. *Finite rule*: This research work helped to develop a novel rule based on linguistic construct of Hindi language. The construct is although very specific to Hindi but can be used for all Indian Languages with similar features like Sanskrit, Oriya, Punjabi. Therefore, this research work also contributes to language detection by proposing a novel feature which is alone sufficient to classify transliterated Hindi text in Roman script as Hindi. The Hindi language is written in Devanagari script. Their being phonetic similarities between English vowels and Hindi vowels स्वर as they are made up of one or two English phonetically similar vowels clubbed together. The Devanagari consonants have a ‘inherent a’ sound. With each consonant the schwa sound must also be pronounced. And majority of words have a common pattern within themselves. The consonant sounds are followed by a vowel sound, which means in the transliterated text, consonants will be followed by vowels very frequently. The below two images explain the construct of Hindi vowels and consonants and also show their close transliterate text.

It is been discussed that transliteration and identification of transliterated text rely on the understanding of the native script the language uses. Hindi has twice as many vowels as English where vowels can be classified into two particular categories *long vowels* and *short vowels* which are articulated on basis of duration of time. The long vowels are usually transliterated to English using syllables as a combination of vowels. For example, (ईङ्ग) the ई is present in feet is ‘ee’, whereas इ is ‘i’ type in bin.

In Figs. 1 and 2, we can see that the phonetic mapping is attempted. Before NMT, one of the first steps was to create a mapping of Hindi phones and English phones for getting the transcriptions of Hindi characters. As an transitory step from the Hindi characters, one could generate English using older version of Google transliteration which used the International Alphabet of Sanskrit Transliteration

**Fig. 1** Hindi vowels and their use with consonants  
[https://www.omniglot.com/  
writing/hindi.htm](https://www.omniglot.com/writing/hindi.htm)

अ	आ	इ	ई	उ	ऊ	ए
a	ा	i	ी	u	ু	e
[A]	[a]	[i]	[i:]	[u]	[w:]	[e]
प	पा	पि	পী	পু	পূ	পে
pa	pā	pi	pī	pu	pū	pe
ঐ	আ	ঔ	ঁ	অ:	ঁ	ঞ
ai	o	au	ań	ah	āñ	r
[æ:]	[o]	[ɔ:]	[aŋ]	[əh]	[ā:]	[r]
ঐ	পো	পৌ	ঁ	পঃ	পঁ	পু
pai	po	pau	pań	paḥ	pāñ	pr

<b>क</b>	<b>ख</b>	<b>ग</b>	<b>घ</b>	<b>ड़</b>	<b>च</b>	<b>छ</b>	<b>ज</b>	<b>झ</b>	<b>ञ</b>
ka	kha	ga	gha	ñ̄a	ca	cha	ja	jha	ñ̄a
[kə]	[kʰə]	[gə]	[gʰə]	[ñ̄ə]	[tʃə]	[tʃʰə]	[də]	[dʒə]	[ñ̄ə]
<b>ट</b>	<b>ठ</b>	<b>ડ</b>	<b>ଢ</b>	<b>ଣ</b>	<b>ତ</b>	<b>ଥ</b>	<b>ଦ</b>	<b>ଧ</b>	<b>ନ</b>
ta	tha	da	dha	ñ̄a	ta	tha	da	dha	na
[tə]	[tʰə]	[də]	[dʰə]	[ñ̄ə]	[tə]	[tʰə]	[də]	[dʰə]	[nə]
<b>प</b>	<b>ଫ</b>	<b>ବ</b>	<b>ଭ</b>	<b>ମ</b>	<b>ଯ</b>	<b>ର</b>	<b>ଲ</b>	<b>ଵ</b>	
pa	pha	ba	bha	ma	ya	ra	la	va	
[pə]	[pʰə]	[bə]	[bʰə]	[mə]	[jə]	[rə]	[lə]	[və]	
<b>শ</b>	<b>ষ</b>	<b>স</b>	<b>হ</b>						
śa	ṣa	sa	ha						
[ʃə]	[ʃə]	[sə]	[ñ̄ə]						
Additional consonants									
<b>়</b>	<b>়খ</b>	<b>়গ</b>	<b>়জ</b>	<b>়়</b>	<b>়ফ</b>	<b>়ড</b>	<b>়ଢ</b>		
qa	ħa	ǵa	za	zha	fa	ṛa	ṛħa		
[qə]	[xə]	[yə]	[zə]	[ʒə]	[fə]	[rə]	[rħə]		
Common conjunct consonants									
<b>়ক্ষ</b>	<b>়়জ্ঞ</b>	<b>়ত্ক</b>	<b>়দ্ব</b>	<b>়য</b>	<b>়দ</b>	<b>়ত্ত</b>	<b>়়ঙ্গ</b>	<b>়়়</b>	
ksa	ʃʃa	tʃʃa	dva	dyɑ	ɖa	tta	ɖħħa	dbha	
<b>়ম</b>	<b>়হ</b>	<b>়ব্য</b>	<b>়শ্র</b>	<b>়ত্র</b>	<b>়প</b>	<b>়প্র</b>	<b>়ট</b>		
dma	ħma	hyɑ	śra	tra	rpa	pra	tra		

Fig. 2 Hindi consonants and inherent a with them <https://www.omniglot.com/writing/hindi.htm>

(IAST) scheme. English recognizer could then be used to search through Hindi for English phonemes that correspond to the Hindi syllables.

### 3.3 Review of Autoencoders and GAN

The main components of the model that are used as the building blocks are discussed below.

#### 3.3.1 Artificial Neural Networks

A neural network is a computation graphical model that is build using many smaller units of computation. These small units called neurons are “hooked” with each other to create a graph. Each neuron is a computational unit which takes input vector and applies a dot product with the weights of the graph to produce the output which is

then squashed between 0 to 1 using a sigmoid or some other activation function.

$$o(x_1, \dots, x_n) = \begin{cases} 1 & \text{if } w_0 + w_1x_1 + \dots + w_nx_n \geq 0 \\ -1 & \text{otherwise} \end{cases}$$

Above is a output of a perceptron unit. For a neuron, the output is usually in form of

$$\frac{1}{1 + \exp(\sum_j w_j x_j + b_j)} \quad (1)$$

Then error loss is calculated and weight update procedure takes place for the network using backpropagation algorithm.

### 3.3.2 Autoencoders

Autoencoders are a particular case of neural networks. The model consists of two parts that go into the autoencoder to complete it. The encoder part that encodes the given data into a latent compact representation. The next half of the autoencoder acts as the ‘decoder’ generating a close representation to data using the latent representation. Within the autoencoders is a variational autoencoder (VAE) which acts as a generative model and not a model that can memorize the data sets by mapping them to latent representations. The constraint on encoder is put, i.e. model defines a posterior distribution on the observed data given the latent representation. Let  $e \sim p(e)$  where  $p(e)$  is usually taken to be unit Gaussian distribution. Also  $x$  and  $q(e|x)$  are the observed data and probability of  $e$  given  $x$ , respectively. Now any latent representation when sampled from unit Gaussian may be decoded to generate images. The loss function which is being used is a two sum. It deals with the trade-off of reconstruction and how nicely does the latent representation match Gaussian distribution.

$$-\log\left(\frac{p(x|e)p(e)}{q(e|x)}\right) = \log(p(x|e)) + D_{KL}(q(e|x)||p(e)) \quad (2)$$

### 3.3.3 Generative Adversarial Networks

The generative adversarial networks or GANs are a set of two competing neural networks. The two halves being generator and the discriminator, respectively. The generator network generates the counterfeit data close to the actual data in its attempt to mimic the real data, whereas the discriminator network tries to find differences in the real and the fake data, that is the data from the data set and the data generated by the generator network. Thus, both networks play a ‘game’ with each other training themselves to get better at what they do. Eventually the generator becomes good enough to fool the discriminator network generating images very similar to the data set images. The learning is formulated by following min-max operation. The  $\hat{x}$  is

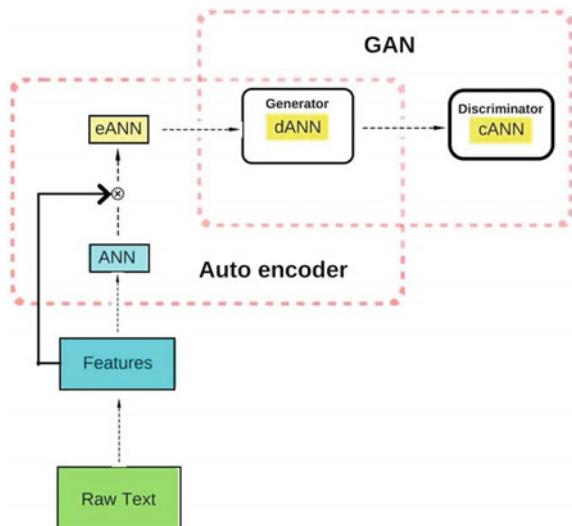
the generated data, where the D, that is, the discriminator is trained to maximize the probability of correct classification and G, that is the Generator is trained to minimize the same. Here  $D(\cdot)$  defines the final output of discriminator and  $G(\cdot)$  defines final output of the generator. So the first term in the equation is the term responsible for improving the accuracy of correct classification, and second term in equation is responsible for reducing the accuracy of incorrect classification; so for discriminator to be better trained, we need to maximize the inner expression, whereas the generator works adversarially to minimize the expression that is to fool the discriminator.

$$\min_G \max_D [E_x(\log(D(x)) + E_e(\log(1 - D(\hat{x})))] \quad (3)$$

### 3.4 Propagation in Model

This section aims to explain the model and also deals with brief discussion of main components in our model which are made using the encoder *eANN* and the decoder *dANN*. The GAN part consists of generator *dANN* and the discriminator *cANN*. As we can see that the decoder part of the autoencoder is same as the generator part of the GAN. Our goal is illustrated in Fig. 3 where we tell that we select features and then make a final layer feature vector which is basically a score vector based on those features. Now to train such network, what do we do is we generate the score vector that is made using the generator network. Then we compare the two score vectors which can then be used to train the network.

**Fig. 3** Block diagram of model



Now considering the raw text we extract the set of features  $x$ . The feature vector  $x$  is then forward propagated through a ANN giving us the score vector  $s$  which is basically a set of values of giving indication to which language does the text belong to  $s = \{s_i : s_i \in [0, 1]\}$ . The score vector is then multiplied to the feature vector  $x$  resulting in a matrix. The matrix is the resized into a long vector which is then fed to the encoder. The encoder produces a deep feature  $e$  specifically for every feature vector. The decoder ANN or the dANN takes the  $e$  as the input and reconstructs the feature vector  $\hat{x}$ . The discriminator then is aimed to distinguish between the original feature vector  $x$ , and the feature vector  $\hat{x}$  generated by the generator network, i.e. dANN. The classifier can be thought of as assigning different class labels to the feature vectors  $x$  and  $\hat{x}$  provided they can be distinguished. The generator and discriminator are trained adversarial until the discriminator is not able to distinguish between real and generated feature vectors.

### 3.5 Training the Model

This section specifies our training of our neural network and our autoencoder parameters  $\{w_s, w_e, w_d\}$ , defining the score ANN or the sANN, the encoder or the eANN and the decoder or the dANN, respectively. There is also training of the GAN parameters  $\{w_d, w_c\}$ , defining the generator network or the dANN and the discriminator network or the cANN. It can be observed that the same dANN is part of both autoencoder as the decoder and acts as the generator in GAN.

Our training is based on four loss functions loss of Gan  $\mathcal{L}_{GAN}$ , loss of autoencoder  $\mathcal{L}_{reconstruct}$ , prior loss  $\mathcal{L}_{prior}$  and regularization loss  $\mathcal{L}_{sparsity}$ . The idea behind training our model is to have a additional score vector  $s_p$  which is taken from some prior distribution  $s_p \sim p(s_p)$ . Now multiplying the input feature vector to the  $s_p$  gives us the input to eANN producing the encoded representation  $e_p$ . Given  $e_p$ , we reconstruct the vector similar to input feature vector as  $\hat{x}_p$ . Now  $\hat{x}_p$  can be used as a means to regularize the cANN such that it can very accurately label  $\hat{x}_p$  to be of the summary class. This also helps in giving a training threshold for the model. That the cANN can classify  $\hat{x}_p$  as summary but gives  $\hat{x}$  original label. The training algorithm, thus, iteratively updates the given three sets of parameters.

- for parameters  $\{w_s, w_e\}$ , the gradient descent is calculated using  $\mathcal{L}_{prior} + \mathcal{L}_{reconstruct}$
- for parameters  $w_d$ , the gradient descent is calculated using  $\mathcal{L}_{GAN} + \mathcal{L}_{reconstruct}$
- for parameters  $w_c$ , the gradient descent is calculated using  $\mathcal{L}_{GAN}$ .

#### 3.5.1 Reconstruction Loss $\mathcal{L}_{Reconstruct}$

The current standards for learning mechanism in autoencoders use Euclidean distances for calculating the difference in actual and reconstructed output. Nevertheless

there have been certain recent findings demonstrating the shortcomings of using Euclidean distance as a means to calculate loss [11]. Therefore, the reconstruction loss  $\mathcal{L}_{reconstruct}$  is defined using the output of the last hidden layer of the discriminator ANN. The loss is modelled as expectation of log likelihood of  $p(\rho(x)/e)$  where  $\rho(x)$  represents the feature vector in the hidden layer of discriminator ANN and  $e$  is the encoded output of encoder ANN.

$$\mathcal{L}_{reconstruct} = E[p(\rho(x)/e)] \quad (4)$$

### 3.5.2 Loss of Gan $\mathcal{L}_{GAN}$

Taking inspiration from [11], the goal to train the discriminator is that it can classify the reconstructed feature vector  $\hat{x}$  as ‘fake’ and original feature vector sequence  $x$  as ‘original’. For regularization of the model being trained, it is additionally enforced that the model learns to classify set of randomly generated feature vectors  $\hat{x}_p$  as ‘fake’. Therefore, the loss is expressed as:

$$\mathcal{L}_{GAN} = \log(cANN(x)) + \log(1 - cANN(\hat{x})) + \log(1 - cANN(\hat{x}_p)) \quad (5)$$

Here  $cANN(.)$  denotes the output of the discriminator ANN. Given the definitions of  $\mathcal{L}_{reconstruct}$  and  $\mathcal{L}_{GAN}$ , we update the parameters  $w_s, w_e, w_d, w_c$  using stochastic gradient descent. The algorithm given below summarizes each step for training the model. The capital letter notation is used to show that variable stands for minibatch of corresponding small alphabets used in the paper.

---

#### Algorithm 1 Training the model

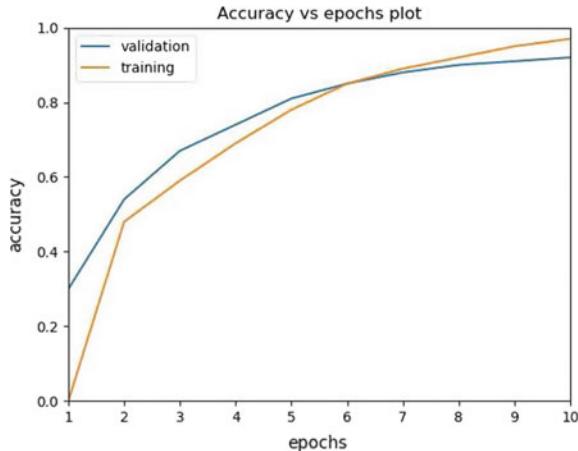
---

```

1: function UPDATE PARAMS  $\triangleright$  where input is the feature vector sequence and output is learned
   parameters  $w_s, w_e, w_d, w_c$ 
2:   for m do  $\alpha$  x number of iterations do
3:      $X = MiniBatchOfFeatureSequences$ 
4:      $S = sANN(X)$   $\triangleright$  select frames
5:      $E = eANN(X, S)$   $\triangleright$  encoding
6:      $\hat{X} = dANN(E)$   $\triangleright$  Reconstruction
7:      $S_p = DrawSamplesFromUniformDistribution$ 
8:      $E_p = eANN(X, S_p)$   $\triangleright$  encoding
9:      $X_p dANN(E_{S_p})$   $\triangleright$  Reconstruction
10:     $\{w_s, w_e\} = \{w_s, w_e\} - \nabla(\mathcal{L}_{reconstruct} + \mathcal{L}_{prior})$   $\triangleright$  weight updates
11:     $\{w_d\} = \{w_d\} - \nabla(\mathcal{L}_{reconstruct} + \mathcal{L}_{GAN})$   $\triangleright$  weight updates
12:     $\{w_c\} = \{w_c\} + \nabla(\mathcal{L}_{GAN})$   $\triangleright$  Maximization Update

```

---



**Fig. 4** Graph of accuracy over training data and validation data

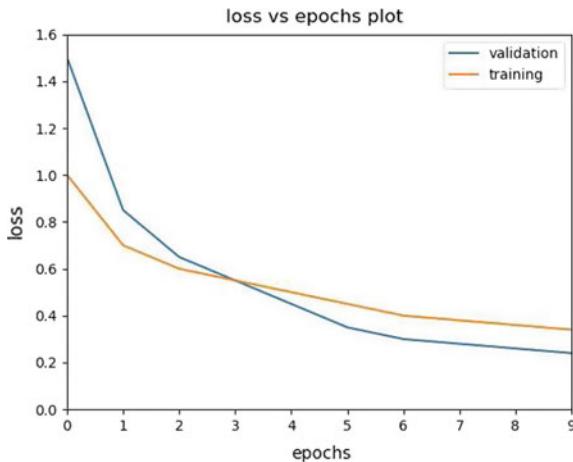
## 4 Results

### 4.1 Training Results

The data set is scraped from reliable sources and then was verified through Google detection API [3] by sending an HTTP request using a URL [4]. The verified data set was divided into training and testing data sets in the ratio 80:20. The training data set was further divided into training and validation data sets in the ratio 80:20. Our model is trained on a batch size of 10,000 words, and an accuracy versus epochs graph is plotted where x-axis represents the number of epochs and y-axis represents accuracy of the model. The blue line denotes the accuracy on validation data set, and red line denotes the accuracy on training data set. Accuracy of the model can be seen increasing on subsequent epochs from the graph on both training as well as on validation data (Fig. 4). In Fig. 5, we talk about loss, the loss taken is the sum of all individual losses and then scaled using a constant.

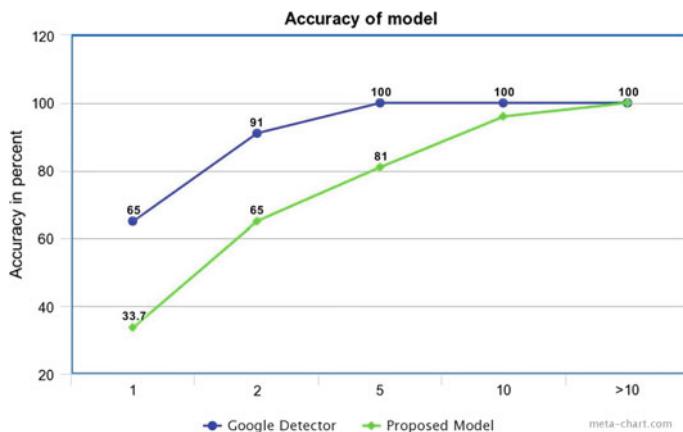
### 4.2 Comparative Results

Google detector is used for comparison of accuracy with our model. The test data was divided into different categories—1 word, 2 words, 5 words, 10 words and more than 10 words each having a batch size of 10,000 examples. Both ours as well as Google model was tested on these five categories, and results were plotted on the graph, where x-axis represents number of words and y-axis represents accuracy of model. The blue line denotes the accuracy of Google detector and green line denotes the accuracy of proposed model. For data set of 1 word category, both Google detector and our model give relatively low accuracy as it includes both unambiguous



**Fig. 5** Graph of loss over training data and validation data

and ambiguous words. Ambiguous words are those which have same spelling in different languages like ‘arre’ is detected as English while it is more commonly used in Hindi. Similarly words like ‘kahan’ and ‘jab’ are also detected as English. Hence, the relatively lower accuracy of 1 word data set is justified due to ambiguity. For data set of 2 words category, the accuracy of both Google and proposed model increases rapidly as ambiguity is much less than 1 word. On increasing the word count per example in data set to 5 words per testing example, the Google detector achieves 100% accuracy. It can be seen from the graph that our proposed model is giving sufficient accuracy when the number of words are increased above 5, and it almost achieves 100% accuracy when word count of test data per example increases above 10 words (Fig. 6).



**Fig. 6** Comparative results of accuracy over number of words in input

## 5 Conclusion

The model proposed performs with reliable accuracy when number of words in data set per example are more than 10 and with sufficient accuracy when number of words are less than 10. Our proposed model gives comparable results with Google detector. It can be concluded from the results that semi-supervised learning models such as ours can use generative adversarial networks for Hindi language identification and for transliterated text identification in general with great accuracy. Thus, the comparison with state-of-the-art Google language detection establishes that the model using feature extraction and then unsupervised learning of parameters is also at par with the current standards in language detection. It also opens up new frontiers were experiment with different types of deep learning models can be done which train in completely unsupervised fashion. Eventually even removing the need to engineer features in such models is an open challenge which future research can look into.

## References

1. Britz, D., et al. (2017) Massive exploration of neural machine translation architectures. [arXiv:1703.03906](https://arxiv.org/abs/1703.03906).
2. Goodfellow, I., et al. (2014). Generative adversarial nets. *Advances in Neural Information Processing Systems*.
3. <https://cloud.google.com/translate/docs/>
4. <https://translation.googleapis.com/language/translate/v2/detect>
5. Jia, Y., Zhu, D., & Yu, S. (2009). A noisy channel model for grapheme-based machine transliteration. In *Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration*. Association for Computational Linguistics.
6. Koehn, P., et al. (2007). Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*. Association for Computational Linguistics.
7. Li, J., Monroe, W., Shi, T., Ritter, A., & Jurafsky, D. (2017). Adversarial learning for neural dialogue generation. [arXiv:1701.06547](https://arxiv.org/abs/1701.06547).
8. Malik, A. et al. (2009). A hybrid model for Urdu Hindi transliteration. In *Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration*. Association for Computational Linguistics.
9. Monier-Williams, M. (1899). *A sanskrit-english dictionary (PDF)*. Oxford: Clarendon Press. pp. xxx.
10. Sharma, D.N. (1972). Transliteration into roman and Devangari of the languages of the Indian group. Survey of India.
11. Wu, L., et al. (2017). Adversarial neural machine translation. [arXiv:1704.06933](https://arxiv.org/abs/1704.06933)
12. Yoshua, C. Generative adversarial networks.

# An Improved Similarity Measure to Alleviate Sparsity Problem in Context-Aware Recommender Systems



Veer Sain Dixit and Parul Jain

## 1 Introduction

Context-aware recommender systems (CARS) serve as an informational filtering tool that filter through a large pool of items and recommend users those products that fits in their current contextual situations [1, 2]. People usually have different preferences in different contextual situations. To quote a few examples: (a) One would like to listen different music if the weather is rainy rather than a hot summer day and/or mood is happy instead sad and (b) the choice of purchasing a set of books would be different for kids and/or own friends.

Collaborative filtering (CF) is a result-oriented and popular approach in recommender system (RS) till date. It works on the fact that similar users have similar preferences in various domains such as entertainment, shopping, hardware items [1–5]. Typically, the core of CF which affects the accuracy of RS is to find the similarity between couple of users/items under sparse environments. However, context-aware datasets are highly sparse since items are rated under different contextual situations [1, 2, 6, 3, 7, 8]. So finding similarity in context-aware dataset is a challenging issue.

**Limitations of existing methods.** Sparsity is one of the major weaknesses in CF which becomes more severe when user preferences are diluted with contextual conditions because users have evaluated few items under different contextual situations from total accessible items. Therefore, it is difficult to compute similarity between users/items in highly sparse datasets, especially context-aware datasets. This affects the performance of RS. However, generic/conventional similarity measures such as

---

V. S. Dixit · P. Jain (✉)

Department of Computer Science, Atma Ram Sanatan Dharam College,  
University of Delhi, New Delhi, Delhi, India  
e-mail: paruljainpj@rediffmail.com

V. S. Dixit

e-mail: veersaindixit@rediffmail.com

the Pearson correlation coefficient (PCC), cosine (COS), mean squared difference (MSD) for similarity computation experience the below-mentioned drawbacks [2, 9–11]:

- Rare co-rated items: Not able to compute similarity between non-corated users/items.
- One co-rated item: PCC gives value either +1 or -1 and COS gives 1 in spite of the presence of rating.
- Local information: Only local information is used, and global information about ratings is ignored.
- Utilization of ratings: Do not utilize all rating information.
- Contextual condition: Contextual conditions about rating information are not considered.

Therefore, existing techniques with these measures suffer from low coverage and precision because of data sparsity and scalability problems [12, 6, 13, 14, 7, 8]. Furthermore, prefiltering [1], postfiltering [1, 2] and contextual modeling [1, 2, 15, 13, 8] in CARS cannot use all the preferences even if they are minimally contextually similar. Even some techniques employing newly emerging similarity measures such as NHSIM [9, 16], Bhattacharyya coefficient with correlation [14, 17] and PSS based [10, 18] which overcome one or another above-mentioned problem but do not consider contextual situation into account.

Motivated by these, we propose a similarity measure and its contextual variant that caters proportion of common ratings, global preferences, ratings of non-corated items, and context attributes. The proposed framework presents user- and item-based techniques employing the proposed similarity measure which suits well to highly sparse data. Moreover, the contribution of each rating in recommendation technique is weighted by contextual similarity to bring contextual effects in recommendations.

The primary contribution in this work is presented as follows:

- A new similarity measure and its contextual variant are proposed which caters local and global ratings, common rating proportions, user rating preferences, and contextual similarity. Overlap or Eskin measure is used to compute contextual similarity in two variants.
- The proposed similarity measures are utilized in user-based and item-based CF where different components are also contextually weighted.
- The proposal is also evaluated for group of users to expand RS research area. Predictive accuracy and classification accuracy metrics are used to analyze the results of recommendations on two contextually rich global datasets.

The remainder of the paper is presented as follows. Few related works are mentioned in Sect. 2. The proposed framework and the details of similarity measures with the algorithms are described in Sect. 3. Section 4 analyzes experimental results. The conclusions followed by some perspective of future research work are written in Sect. 5.

## 2 Related Work

CF is a personalized recommendation technique widely used in a variety of domains but also suffers from cold-start problem, data sparsity, and scalability issues [9, 19, 20, 10, 11, 21]. Usually, the databases are very sparse where users have evaluated small number of items. When the preferences are further diluted by the inclusion of contextual conditions, then the sparsity problem becomes more severe [12, 6, 3, 8, 22, 23]. Therefore, researchers have been focusing on predictive accuracy and proposed some solutions.

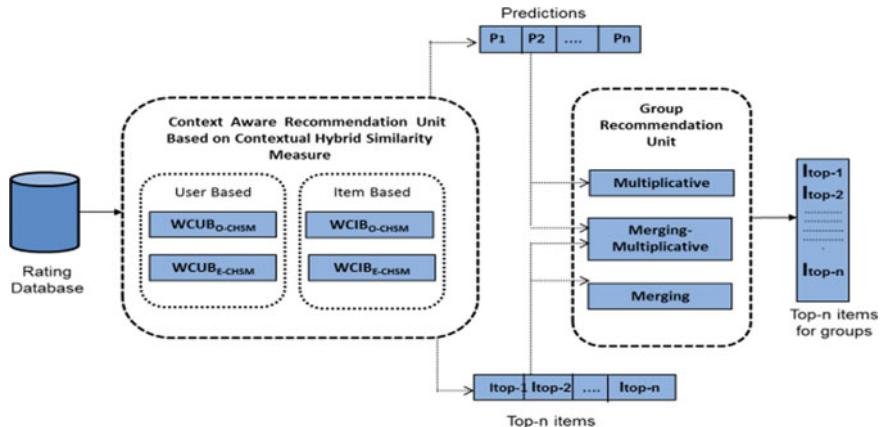
Many researchers have proposed some new similarity measures in order to increase accuracy. A similarity measure exploiting Bhattacharyya coefficient which overcomes the problem of usage of only co-rated items is suggested by [14], but it ignores context factors and use of correlation-based measures. Another metric, proximity–impact–popularity measure that considers the set of common ratings and their absolute value but ignores global preferences is proposed in [9]. Traditional similarity measure takes weighted average of all ratings, so to capture the confidence on each neighbor weighted similarity measure is used in [11]. Taking care of global preferences and local context, the author proposed a modified heuristic similarity measure in [18] but that is not suitable for context-aware datasets. Another similarity metric defined by [16] called as NHSM solves the new user cold-start problem but does not address contextual situations.

Moreover, addressing the sparsity issue in context-aware dataset, DCR algorithm is proposed but that suffers from co-rated item problem [13], whereas it is proved that two items can also be similar even if they are not co-rated [14]. Another approach called as DCW for CARS is described in [8] where contextual features are weighted and particle swarm optimization is used to efficiently determine the weights but used traditional PCC for calculating similarity that works on co-rated items only.

However, the above approaches either do not address contextual conditions or use traditional PCC, COS, MSD as similarity measure. Instead, we aim to capture the importance of contextual similarity, local and global preferences, and proportion of common ratings to increase the accuracy in CARS.

## 3 The Proposal

In this section, we propose a framework depicted in Fig. 1 that provides techniques to generate recommendations using the proposed similarity measures to ease data sparsity problem. In the framework, context-aware recommendation unit based on contextual hybrid similarity measure present several algorithms based on user and item model utilizing proposed similarity measures. The techniques are explored in the domains where items are typically recommended to individuals as well as group of users.



**Fig. 1** Proposed framework

### 3.1 Proposed Similarity Measure for Collaborative Filtering (O-CHSM and E-CHSM)

Different ratings are given by the users in different contextual situations in context-aware datasets which make them highly sparse. The conventional similarity measures such as PCC, COS, MSD are not suitable to context-aware datasets. In this section, we describe the proposed similarity measure Overlap-based contextual hybrid similarity measure (O-CHSM). It combines the Overlap measure for contextual similarity, Bhattacharyya coefficient for global rating information and NHSM. NHSM considers proportion of common ratings, global preferences of user behavior and can be easily combined with other similarity measure. Another variation of the newly proposed similarity measure is Eskin-based contextual hybrid similarity measure (E-CHSM) which uses Eskin measure instead Overlap for contextual similarity. The proposed similarity measures overpower the limitations of traditional similarity measures and suits well to sparse data especially context-aware datasets. The motivations for the improved similarity measure approach are as follows:

- Scarcity of co-rated items: The context-aware datasets where users rate items differently in different contextual conditions are highly sparse. The proposed similarity measure is suitable to those datasets and can find similarity between two users and items even when co-rated items are rare or none.
- Utilization of Global rating information: Bhattacharyya coefficient extracts global rating information as similarity value between a couple of items or users.
- Utilization of NHSM: This similarity measure is used to calculate local information. It considers not only the absolute ratings but also the proportion of common ratings. The similarity is decided by global preference of the user behavior and can be easily combined with other similarity measures.

- Contextual Similarity: Particularly, it gives importance to the contextual similarity of two vectors.

### 3.2 The Formalization of Similarity Measures

#### Overlap/Simple Matching Coefficient.

The simple matching coefficient (i.e., Overlap) is a simple, effective, and widely used method for categorical attributes which determines the similarity between two objects  $ob_i$  and  $ob_j$  [24] and is defined by

$$S(ob_i, ob_j) = \frac{\sum_{t=1}^n S_t(ob_{it} - ob_{jt})}{n}$$

where t represents the attribute and n depicts the number of attributes in total of the object.  $ob_{it}$  means t-th attribute of object  $ob_i$ . The value of

$$S_t(ob_{it}, ob_{jt}) = 1 \text{ if } ob_{it} = ob_{jt} \text{ and otherwise } 0.$$

#### Eskin Measure.

The Eskin measure was proposed by Eskin et al. (2002) which assigns comparatively more weightage to mismatches of attributes with the higher number of categories [24]. The similarity between objects  $ob_i$  and  $ob_j$  is defined by  $S(ob_i, ob_j) = \frac{\sum_{t=1}^n S_t(ob_{it} - ob_{jt})}{n}$  where  $ob_{it}$  means t—the attribute of the object  $ob_i$ .

$$S_t(ob_{it}, ob_{jt}) = 1 \text{ if } ob_{it} = ob_{jt} \text{ otherwise } \frac{m_t^2}{m_t^2 + 2}$$

where  $m_t$  represents number of categories of the t-th variable.

To give weightage to contextual situations, context similarity of two vectors is calculated using Overlap or Eskin measure. It assumes that those ratings become more valuable in making predictions whose contexts are more similar to that of active user. The following example describes the scenario where Table 1 shows ratings assigned by different users to a movie in different contextual situations.

From Table 1, the contextual similarity between u1 and u4 using Overlap measure is  $S(u1, u4) = \frac{2}{5} = 0.4$ , whereas Eskin measure compute it as

$$S(u1, u4) = \left( \frac{9}{11} + 1 + \frac{2}{3} + 1 + \frac{2}{3} \right) / 5 = 0.83$$

However, contexts with low similarity may enhance noisy ratings while making predictions, so a set of similarity thresholds are established for each component of

**Table 1** Scenario of using context similarity

User	Day type	Time	Location	Social	Mood	Rating
u1	Weekend	Night	Cinema hall	Colleagues	Positive	4
u2	Holiday	Evening	Friend's house	Girlfriend	Positive	5
u3	Weekday	Night	Cinema hall	Colleagues	Neutral	4
u4	Weekday	Night	Friend's house	Colleagues	Neutral	3

**Table 2** Rating vectors of items  $I$  and  $J$ 

$I$	1	0	2	0	1	0	2	0	3	0
$J$	0	1	0	2	0	1	0	2	0	3

prediction algorithm. This also means that ratings having context below the threshold are ignored.

### Bhattacharyya Coefficient.

The Bhattacharyya measure provides similarity between two probability distributions [14]. If  $\hat{p}_i$  and  $\hat{q}_j$  be the estimated discrete densities of the two items  $i$  and  $j$ , then the Bhattacharyya coefficient (BC) between items  $i$  and  $j$  is defined as

$$BC(i, j) = BC(\hat{p}_i, \hat{q}_j) = \sum_{t=1}^m \sqrt{(\hat{p}_{it})(\hat{q}_{jt})}$$

where  $m$  represents number of assigned ratings and  $\hat{p}_{it} = \frac{\#_t}{\#_i}$  where  $\#_i$  depicts the number of users rated item  $i$ ,  $\#_t$  is the number of users gave rating value  $p$  to item  $i$ .

### Illustrative Example.

The following example illustrates the computation of BC where Table 2 represents the rating vectors of items  $I$  and  $J$  by distinct users. The rating scale lies in the range {1, 2, 3}.

The BC is computed as:

$$BC(i, j) = \sum_{h=1}^3 \sqrt{(\hat{I}_h)(\hat{J}_h)} = \sqrt{\frac{2}{5} * \frac{2}{5}} + \sqrt{\frac{2}{5} * \frac{2}{5}} + \sqrt{\frac{1}{5} * \frac{1}{5}} = 1$$

### Proximity–Significance–Singularity (PSS).

To punish bad similarity and reward good similarity, we have used PSS similarity as local measure [9, 10, 18]. The user PSS similarity is defined as follows:

$$sim(u, v)^{PSS} = \sum_{a \in I_u, b \in I_v} PSS(r_{ua}, r_{vb}) \text{ where } I_u \text{ represents all the items rated by user } u \text{ and } r_{ua} \text{ means rating assigned by user } u \text{ to item } a.$$

$$PSS(r_{ua}, r_{vb}) = \text{Proximity}(r_{ua}, r_{vb}) \times \text{Significance}(r_{ua}, r_{vb}) \times \text{Singularity}(r_{ua}, r_{vb}).$$

Proximity computes whether two ratings are in agreement or not and assigns penalty to disagreement. It also considers absolute difference between two ratings.

Significance believe that those ratings more significant which are more away from the median rating.

Singularity represents difference of two ratings from the mean of their rating vector.

$$\text{Proximity}(r_{ua}, r_{vb}) = 1 - \frac{1}{1 + \exp(-|r_{u,a} - r_{v,b}|)}$$

$$\text{Significance}(r_{ua}, r_{vb}) = 1 - \frac{1}{1 + \exp(-|r_{u,a} - r_{med}| * |r_{v,b} - r_{med}|)}$$

$$\text{Singularity}(r_{ua}, r_{vb}) = 1 - \frac{1}{1 + \exp(-|((r_{u,a} - \bar{r}_a) + (r_{v,b} - \bar{r}_b))/2|)}$$

### **Jaccard Coefficient.**

The proportion of common rating is an important factor and included using Jaccard coefficient [14, 18, 11].

$$\text{sim}(u, v)^{\text{Jaccard}} = \frac{|I_u \cap I_v|}{|I_u| |I_v|}$$

### **User Rating Preference (URP).**

Distinct users rate items in distinct manner. Some users rate items highly and some gives low ratings. To reflect this rating behavior of users, the user rating preference model [10, 18, 21] is used which is based on mean and variance of ratings and defined as:

$$\text{sim}(u, v)^{\text{URP}} = 1 - \frac{1}{1 + \exp(-|\mu_u - \mu_v| * |\sigma_u - \sigma_v|)}$$

### **New Heuristic Similarity Measure (NHSM).**

The NHSM is a combination of Jaccard, URP and PSS as described in [10] and given by the following equation:

$$s(u, v)^{\text{NHSM}} = \text{sim}(u, v)^{\text{Jaccard}} * \text{sim}(u, v)^{\text{URP}} * \text{sim}(u, v)^{\text{PSS}}$$

### **Contextual Hybrid Similarity Measure (CHSM).**

Taking Overlap/Eskin, Bhattacharyya coefficient, Jaccard, PSS, and URP similarity into account, we combine these similarity measures to calculate similarity between a couple of users or items. The contextual hybrid similarity measures are defined by the following equations.

**Table 3** Notations and their meaning used in the algorithms

Notation	Meaning
$P_{a,i,c}$	Rating to be predicted for item $i$ by user $a$ with context vector $c$
$N_{i,\Phi}$	Neighborhood of those items which are rated by user $a$ and contextual similarity of two vectors is $\geq \Phi$
$N_{a,\Phi}$	Neighborhood of those users who have rated item $i$ and contextual similarity of two vectors is $\geq \Phi$
$\tilde{\omega}(a, \Phi)$	Average rating of user $a$ on those vectors where context similarity $\geq \Phi$
$\omega(t, \Phi)$	Weighted rating of neighbor $t$ where context similarity $\geq \Phi$
$su_O\text{-CHSM}(u, v)$	Overlap-based contextual hybrid similarity between users $u$ and $v$
$su_E\text{-CHSM}(u, v)$	Eskin-based contextual hybrid similarity between users $u$ and $v$
$si_O\text{-CHSM}(i, j)$	Overlap-based contextual hybrid similarity between items $i$ and $j$
$si_E\text{-CHSM}(i, j)$	Eskin-based contextual hybrid similarity between items $i$ and $j$

$$su_O\text{-CHSM}(u, v) = O(u_c, v_c) * \sum_{a \in I_u} \sum_{b \in I_v} BC(a, b) * O(ua_c, vb_c) * s(ua, vb)^{NHSM} \quad (1)$$

$$su_E\text{-CHSM}(u, v) = E(u_c, v_c) * \sum_{a \in I_u} \sum_{b \in I_v} BC(a, b) * E(ua_c, vb_c) * s(ua, vb)^{NHSM} \quad (2)$$

$$si_O\text{-CHSM}(i, j) = O(i_c, j_c) * \sum_{a \in U_i} \sum_{b \in V_j} BC(a, b) * O(ia_c, jb_c) * s(ia, jb)^{NHSM} \quad (3)$$

$$si_E\text{-CHSM}(i, j) = E(i_c, j_c) * \sum_{a \in U_i} \sum_{b \in V_j} BC(a, b) * E(ia_c, jb_c) * s(ia, jb)^{NHSM} \quad (4)$$

### 3.3 Context-Aware Recommendation Unit Based on Contextual Hybrid Similarity Measure

In this section, we elaborate on all the algorithms proposed in the framework (Fig. 1) in detail which are described by Eqs. 5, 6, 7, and 8. The various notations used in Eqs. 5, 6, 7, and 8 with their meaning are presented in Table 3. Equations 1, 2, 3, and 4 explain the computation of similarity measures used in Eqs. 5, 6, 7, and 8.

The equations to predict the rating for unrated item  $i$  by active user  $a$  are as follows:

**Weighted Context (using Overlap) User Based with CHSM ( $WCUB_{O-CHSM}$ ).**

$$P_{a,i,c} = \bar{\omega}(a, \Phi) + \frac{\sum_{t \in N_{a,\Phi}} su_{O-CHSM}(a, t)(\omega(t, \Phi) - \bar{\omega}(t, \Phi))}{\sum_{t \in N_{a,\Phi}} su_{O-CHSM}(a, t)} \quad (5)$$

**Weighted Context (using Eskin) User Based with CHSM ( $WCUB_{E-CHSM}$ ).**

$$P_{a,i,c} = \bar{\omega}(a, \Phi) + \frac{\sum_{t \in N_{a,\Phi}} su_{E-CHSM}(a, t)(\omega(t, \Phi) - \bar{\omega}(t, \Phi))}{\sum_{t \in N_{a,\Phi}} su_{E-CHSM}(a, t)} \quad (6)$$

**Weighted Context (using Overlap) Item Based with CHSM ( $WCIB_{O-CHSM}$ ).**

$$P_{a,i,c} = \bar{\omega}(i, \Phi) + \frac{\sum_{t \in N_{i,\bar{\omega}(i,\Phi)}} si_{O-CHSM}(i, t)(\omega(t, \Phi) - \bar{\omega}(t, \Phi))}{\sum_{t \in N_{i,\Phi}} si_{O-CHSM}(i, t)} \quad (7)$$

**Weighted Context (using Eskin) Item Based with CHSM ( $WCIB_{E-CHSM}$ ).**

$$P_{a,i,c} = \bar{\omega}(i, \Phi) + \frac{\sum_{t \in N_{i,\Phi}} si_{E-CHSM}(i, t)(\omega(t, \Phi) - \bar{\omega}(t, \Phi))}{\sum_{t \in N_{i,\Phi}} si_{E-CHSM}(i, t)} \quad (8)$$

### 3.4 Group Recommendation Unit

This unit is aimed to provide recommendations to group of users as described in Fig. 1 extending our research area. This unit implements three techniques for group recommendations: Merging, Multiplication, and Merging + Multiplication. Several works [25, 26] found Merging technique a simple, efficient, and widely used method as group techniques. Christensen and Schiaffino [25] suggests multiplication technique more effective in terms of individual satisfaction and Merging–Multiplication outperforms the other two. We analyzed the effectiveness of our proposed algorithms for group recommendations using the following techniques.

**Merging.** This method first collects top-n items recommended to each member of the group and then merges them into a single list. Then top-n items are recommended to the group as a whole [25, 26].

**Multiplication.** This method produces an aggregated rating after multiplication of the predicted ratings of each member of group and further recommends top-n items with highest predicted ratings [25].

**Merging–Multiplication.** First, this method combines top-n recommended items for each member of group into a single list, and then merging is used to filter items

**Table 4** Statistics of datasets

Datasets	# of users	# of items	# of ratings	# of contexts factors	# of user attributes	# of item attributes	Rating scale
IncarMusic	42	139	4012	8	1	8	1–5
LDOS-CoMoDa	121	1232	2296	12	4	11	1–5

with highest ratings. Afterward, an aggregated rating is obtained for top-n items on multiplying the predicted rating recommended to group as a whole [25].

## 4 Experimental Evaluation

To analyze the efficacy of the proposed framework, we have performed distinguished experiments where the following issues are addressed:

- How do the proposed similarity measure and its variant perform?
- To analyze several algorithms based on user model and item model using our proposed similarity measure in context-aware scenario.
- Do the proposed algorithms effective for group recommendations?

### 4.1 Data Preparation and Evaluation Metrics

We conduct the experiments on two contextually rich global datasets from the movie and music domains [27] and specially designed for context-aware personalization research and are collected from surveys. The summary of these datasets is given in Table 4.

We randomly partitioned the datasets into three parts after filtering out those users who have evaluated less than three ratings. Two parts are used as training set, and rest one is taken up as test set. The experiments are repeated five times, and average over five runs are presented for all measures. The predictive performance was measured by mean absolute error (MAE) and root mean squared error (RMSE). Furthermore, precision (P), recall (R), and F1-score (F1) measures for top 10 items are used for item recommendations. Here, an item that is considered relevant (a hit) only if it is rated more than or equal to 4 (among 5) by the active user in both the IncarMusic dataset and LDOS-CoMoDa dataset. We set the minimal coverage threshold  $\theta$  as 0.5 after doing rigorous parameter tuning. The same  $\theta$  threshold values are kept for all components of the algorithms to optimize the process using both datasets since rating scale is same (1–5).

For group recommendation, each approach is evaluated using small group (SG) (size 3–5) and large group (LG) (size 6–8) with five random groups. The average attained for five runs on F1 metric is presented for this case.

## 4.2 Compared Methods

To demonstrate the performance of the algorithms based on proposed similarity measures described in Sect. 3, we choose one context-aware recommendation approach DCW<sub>PEARSON</sub> where contextual feature are weighted [8] and another as a collaborative filtering method based on a similarity measure utilizing Bhattacharyya coefficient suitable for sparse dataset [14].

**DCW<sub>PEARSON</sub>.** This method is reported as a collaborative filtering approach where weighted contexts are implemented in different components of the algorithms which is another solution of data sparsity problem [8]. Though [8] implemented DCW<sub>PEARSON</sub> in only user-based model but we have used item-based model too. Here also, we used the same threshold  $\theta$  as 0.5 in different parts of the algorithm for the comparison purpose.

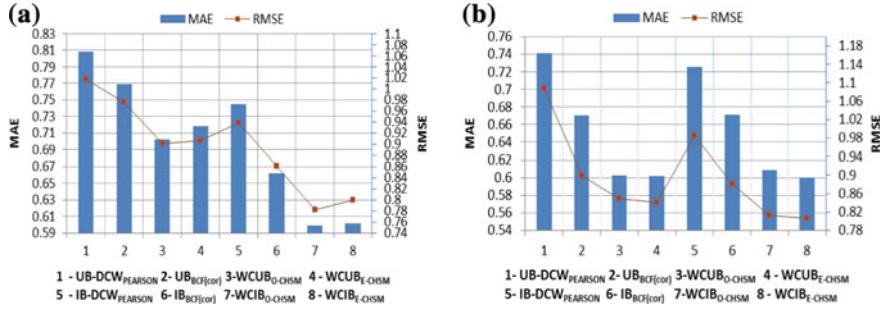
**BCF(cor).** It is a neighborhood-based CF approach employing a new similarity measure based on Bhattacharyya coefficient and gives importance to proportion of common ratings [14]. Though in [14] the similarity measure is applied on user-based CF but we have also implemented on item-based CF to compare the proposed item-based algorithms. Since BCF(cor) is proved better than BCF(med) in [14], so we choose to implement BCF(cor) on our datasets.

## 4.3 Results and Analysis

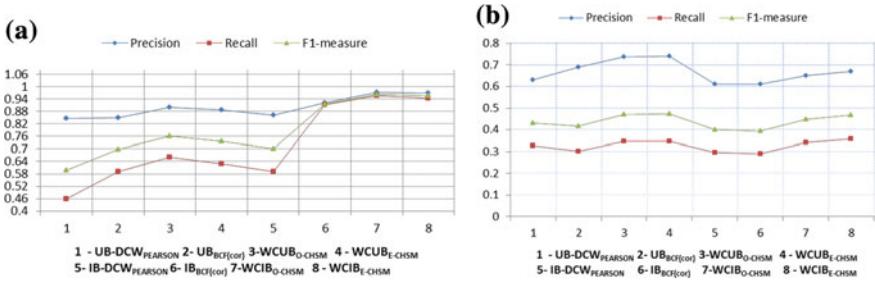
Here, we present and analyze the experimental results on LDOS-CoMoDa and Incar-Music datasets.

### Analysis of the proposed similarity measure and its variant.

Figure 2a, b depict that the proposed similarity measure-based algorithms show remarkable improvement over BCF(cor). The reason could be that the proposed Similarity measure covers contextual conditions of users/items and user rating preferences. The two variants which differ in computing contextual similarity are close to each other in terms of both predictive accuracy(MAE and F1-score). Proposed similarity measure-based algorithms reduces MAE more than 10% compared to other context-aware methods. Similar trend is seen in RMSE metric. E-CHSM in LDOS-CoMoDa dataset makes more error than O-CHSM since Eskin performs poorly in those datasets in which attributes take large number of values. On the other hand, E-CHSM performs better than O-CHSM with IncarMusic dataset. No single mea-



**Fig. 2** Performance comparison of different algorithms based on different similarity measures using MAE and RMSE on: **a** LDOS-CoMoDa dataset and **b** IncarMusic dataset



**Fig. 3** Performance comparison of different approaches using precision, recall and F1-score on two datasets. **a** LDOS-CoMoDa and **b** IncarMusic

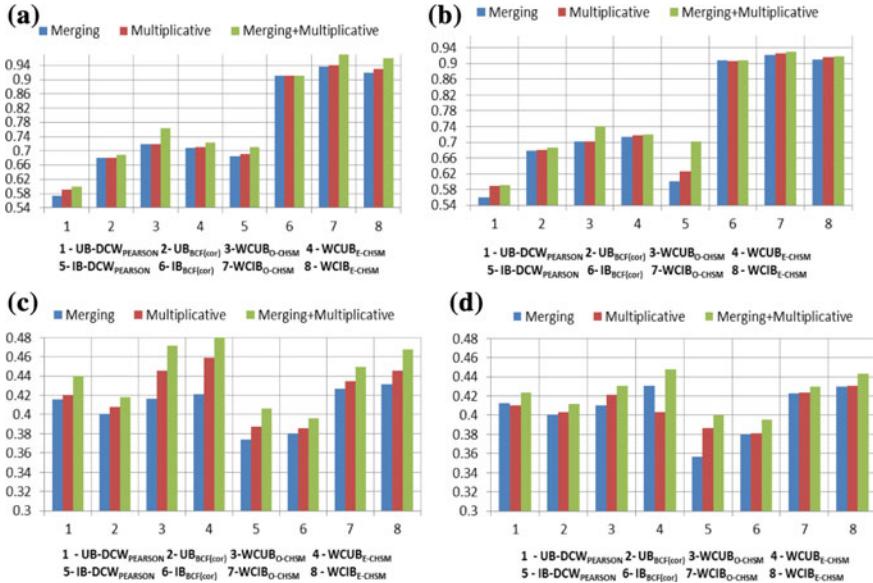
sure is always superior or inferior. Different similarity measures handle different characteristics of a dataset.

Hence, it can be concluded that proposed similarity measures shows good improvement over BCF(cor) and PEARSON similarity measure and are the best performing one. Also, Overlap and Eskin measures used for contextual similarity are dataset dependent.

#### Analysis of the performance of different algorithms.

Figure 3a, b represents the execution comparison of variants of proposed algorithms employing the proposed similarity measure w.r.t. DCW- and BCF(cor)-based algorithms. This is to be noted that algorithms using CHSM similarity measures improves accuracy (F1-score by more than 10%) while making predictions compared to BCF(cor)-based similarity measure which also works on non-correlated items. The algorithms ( $WCIB_{O-CHSM}$  in movie dataset and  $WCIB_{E-CHSM}$  in music dataset) based on item model are superior (reduces MAE by 10%) than user models ( $WCUB_{O-CHSM}$  in movie dataset and  $WCUB_{E-CHSM}$  in music dataset).

The  $WCIB_{O-CHSM}$  algorithm can obtain F1-score close to 0.96 and its closest competitor  $WCIB_{E-CHSM}$  can obtain F1-score close to 0.95. The difference in the accuracy (F1-score) of recommendations in user and item-based approaches is more



**Fig. 4** Comparison of different algorithms in group recommendations using F1 metric on two datasets. **a** Small groups (size 3–5) with LDOS-CoMoDa, **b** Large groups (size 6–8) with LDOS-CoMoDa, **c** Small groups (size 3–5) with IncarMusic, **d** Large groups (size 6–8) with IncarMusic

than 20% in movie dataset unlike the music dataset. The PCC-based algorithms (DCW) performed worst. This indicates that they could not retrieve items properly.

The Pearson correlation-based algorithms UB-DCW<sub>PEARSON</sub> and IB-DCW<sub>PEARSON</sub> have a F1-score of less than 0.7 in movie domain and 0.43 in music domain. This clearly indicates Pearson-based algorithms are not much reliable to produce recommendations in sparse datasets. Our proposed algorithms can handle highly sparse datasets in much more effective way, and algorithms using item neighborhood are better than user neighborhood-based algorithms.

### Effectiveness of algorithms for Group Recommendations.

In general, Fig. 4a–d clearly depict that the proposed algorithms can effectively work for groups of users like individuals. The predictive accuracy (MAE and F1-measure) is slightly improved with decrease in the group size. Merging–Multiplication shows remarkable improvement over other grouping techniques. It is to be noted that the improvement in F1-score reveals that the aggregation of the ranked lists of the group members is able to fix errors which otherwise were produced in the individual predictions.

Hence, the proposed algorithms are effective for group recommendations also.

## 5 Conclusions and Future Work

In this paper, we proposed an improved similarity measure which overcomes the traditional similarity measure drawbacks. The proposed similarity measure is based on PSS, URP, Bhattacharyya coefficient, Jaccard coefficient, and Overlap measure. Therefore, it caters local and global ratings, user rating preferences, proportion of common ratings and contextual conditions of the users. The variant of this measure uses Eskin instead Overlap for contextual similarity. Each factor in the proposed measure lies in the range {0, 1} and hence it is normalized. These measures suit well to highly sparse data and give importance to contextual similarity. The effectiveness of the proposed similarity measures are explored using user-/item-based collaborative filtering where each component is contextually weighted. The experiment results depict that the proposed similarity measure can overpower the limitations of traditional similarity measures. Also, the algorithms using proposed similarity measure perform better than other methods. The performance of Overlap and Eskin measures to find contextual similarity is dataset dependent. The reason might be Eskin depends on the number of values an attribute can get which is not true with Overlap. Finally, the proposed algorithms are also proved effective for Group Recommendations.

In future, we aim to unite these approaches with latent factor models such as matrix factorization.

## References

1. Adomavicius, G., & Tuzhilin, A. (2011). Context-aware recommender systems. In *Recommender systems handbook* (pp. 217–253).
2. Adomavicius, G., Sankaranarayanan, R., Sen, S., & Tuzhilin, A. (2005). Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Transactions on Information systems (TOIS)*, 23(1), 103–145.
3. Odić, A., Tkalčić, M., Tasić, J., & Košir, A. (2012). Relevant context in a movie recommender system: Users opinion versus Statistical detection. In *Proceedings of the 4th International Workshop on Context-Aware Recommender Systems*. Dublin, Ireland.
4. Papagelis, M., & Plexousakis, D. (2005). Qualitative analysis of user-based and item-based prediction algorithms for recommendations agents. *Engineering Applications of Artificial Intelligence*, 18, 781–789.
5. Baltrunas, L., Ludwig, B., Peer, S., & Ricci, F. (2012). Context relevance assessment and exploitation in mobile recommender systems. *Personal and Ubiquitous Computing*, 16(5), 507–526.
6. Zheng, Y., Mobasher, B., & Burke, R. (2014). Context recommendation using multilabel classification. In *IEEE/WIC/ACM International Joint Conference on Web Intelligence (WI) and Intelligent Agent Technologies (IAI), ACM Recsys* (pp. 301–304). ACM, Silicon Valley.
7. Zheng, Y., Burke, R., & Mobasher, B. (2012). Optimal feature selection for context-aware recommendation using differential relaxation. In Conference Proceedings of the 4th International Workshop on Context-Aware Recommender Systems. Dublin, Ireland: ACM RecSys.
8. Zheng, Y., Burke, R., & Mobasher, B. (2012). Recommendations with Differential context weighting. In *UMAP* (pp. 152–164), Springer.
9. Liu, H., Hu, Z., Mian, A., Tian, H., & Zhu, X. (2014). A new user model to improve the accuracy of collaborative filtering. *Knowledge-Based Systems*, 56, 156–166.

10. Katpara, H., & Vaghela, V. B. (2016). Similarity measures for collaborative filtering to alleviate the new user cold start problem. In *3rd International Conference on Multidisciplinary Research & Practice* (Vol. 4, No. 1, pp. 233–238).
11. Candillier, L., Meyer, F., & Fessant, F. (2008, July). Designing specific weighted similarity measures to improve collaborative filtering systems. In *Industrial Conference on Data Mining, LNAI* (pp. 242–255).
12. Zheng, Y., Mobasher, B., & Burke, R. D. (2013). The Role of emotions in context-aware recommendation. In *Decisions@ RecSys* (pp. 21–28).
13. Zheng, Y., Burke, R., & Mobasher, B. (2012). Differential context relaxation for context-aware travel recommendation. In *International Conference on Electronic Commerce and Web Technologies* (pp. 88–99). Springer, Berlin, Heidelberg.
14. Patra, B. K., Launonen, R., Ollikainen, V., & Nandi, S. (2015). A new similarity measure using Bhattacharyya coefficient for collaborative filtering in sparse data. *Knowledge-Based Systems*, 82, 163–177.
15. Kim, K. R., & Moon, N. (2011). Recommender system design using movie genre similarity and preferred genres in smart phone. *Multimedia Tools Applications*, 61(1), 87–104.
16. KG, S., & Sadashivam, G. S. (2017). Modified heuristic similarity measure for personalization using collaborative filtering technique. *Applied Mathematics and Information Sciences* 11(1), 307–315.
17. Miao, Z., Zhao, L., Huang, P., Yu, Y., Qiao, Y. (2016). Song: Methods for improving the similarity measure of sparse scoring based on the Bhattacharyya measure. In *International Conference on Artificial Intelligence: Techniques and Applications*.
18. KG, S., & Sadashivam, G. S. (2017). Modified heuristic similarity measure for personalization using collaborative filtering technique. *Applied Mathematics and Information Science*, 1, 307–315.
19. Adamopoulos, P., & Tuzhilin, A. (2013). Recommendation opportunities: improving item prediction using weighted percentile methods in collaborative filtering systems. In *Proceedings of the 7th ACM Conference on Recommender Systems*, Hong Kong, China (pp. 351–354). ACM.
20. Panniello, U., Tuzhilin, A., & Gorgoglion, M. (2014). Comparing context-aware recommender systems in terms of accuracy and diversity. *User Modeling and User-Adapted Interaction*, 24(1–2), 35–65.
21. Mahara, T. (2016). A new similarity measure based on mean measure of divergence for collaborative filtering in sparse environment. In Twelfth International Multi-conference on Information Processing; *Procedia Computer Science*, 89, 450–456.
22. Zheng, Y., Mobasher, B., & Burke, R. (2014). CSLIM: Contextual SLIM recommendation algorithms. In *Proceedings of the 8th ACM Conference on Recommender Systems*, (pp. 301–304). ACM.
23. Zheng, Y. (2015). A revisit to the identification of contexts in recommender systems. In *20th International Conference on Intelligent Users Interfaces*, ACM IUI (pp. 109–115). Atlanta, GA, USA.
24. Sulc, Z., Rezankova, H. (2014). Evaluation of recent similarity measures for categorical data. In *17th Application of mathematics and statistics in economics, International Scientific Conference, Poland*.
25. Christensen, I. A., & Schiaffino, S. (2011). Entertainment recommender Systems for group of users. *Expert Systems with Applications*, 38, 14127–14135.
26. Baltrunas, L., Makcinskas, T., & Ricci, F. (2010). Group recommendations with rank aggregation and collaborative filtering. In *Proceedings of the Fourth ACM Conference on Recommender Systems* (pp. 119–126). ACM.
27. Baltrunas, L., Kaminskas, M., Ludwig, B., Moling, O., Ricci, F., Aydin, A., et al. (2011). InCarMusic: Context-Aware Music Recommendations in a Car. In C. Huemer & T. Setzer (Eds.), *EC-Web, LNBP 85* (pp. 89–100). Heidelberg: Springer.

# Trust and Reputation-Based Model to Prevent Denial-of-Service Attacks in Mobile Agent System



Praveen Mittal and Manas Kumar Mishra

## 1 Introduction

Distributed system provides resource availability at various geographical locations. Mobile agent system uses this concept of distributed system as a key point and executes its line of code at various locations of resources. An agent is a program that assists people and acts on their behalf. Agents function by allowing people to delegate work to them [1]. This program migrates from one platform to another for the partial or full execution of its line of code. Not only that, but it can migrate from one platform to another for accessing various resources. A platform is nothing but a computer system, which can create number of mobile agents for various tasks.

In mobile agent system, deploying an agent on various platforms involves the possibility of denial-of-service attacks on agent. Hence, the mobile agent environment should be secure and reliable for agent to execute. The proposed model maintains reputation of the platform to which the agent will get executed and trust of the mobile agent. In the following section, we analyze some of the mobile agent's development kits available in market.

### 1.1 A Mobile Agent Kit

Mobile agent is a program that can be developed with the help of mobile agent development kits available in market such as Concordia, Jacada, Aglets, Voyager,

---

P. Mittal (✉) · M. K. Mishra  
GLA University, Mathura, India  
e-mail: praveen.mittal@gla.ac.in

M. K. Mishra  
e-mail: manas.mishra@gla.ac.in

Mole, and Odyssey [1]. Following section describes more about these mobile agent development kits.

Concordia is a full-featured framework made up of Java virtual machine for the development and management of network-efficient mobile agent applications, which extend to any device supporting Java. Concordia consists of various components, which are written in Java and combine together to provide a complete, robust environment for applications.

Jacada is another mobile agent software development kit which is freely available for download with a license key on a trial basis. It is designed only for Windows, size of a file is 160 MB, and system requirements are 5 GB hard disk space, 1 GB RAM, operating system must be higher than Windows XP SP2 [1].

Aglet is a very popular and famous mobile software development kits and freely available over the Internet. Tryllian is a company which developed various mobile agent development kits for different platform such as PDA and smart phones. A management server monitors the deployed agents and able to act accordingly. Tryllian offers services of license and subscription. Deploying an agent on the Tryllian network and installing agent development kit on server are the opportunities comes with subscription and license, respectively.

Voyager from object space is a Java-based platform. It facilitates objects to migrate as agents in the network. Voyager provides the use of object request broker with those of a mobile agent system.

Mole is a Java-based framework designed in University of Stuttgart for mobile agents. Major functionalities of include messages, RPC, and sessions. The research areas include security issues, communication between agents and agent groups and transactional by agents.

Another manufacturer of mobile agent is General Magic which creates Telescript, the first commercial mobile agent system. As Java is very popular language for Web-based applications. So General Magic planned to deploy a mobile agent paradigm in its Java-based Odyssey. Java class library was developed to create mobile agent applications.

## ***1.2 Advantages of Mobile Agents***

Mobile agent system should not be accepted on the basis of technology but should be chosen on the basis of benefits. Few benefits are as follows:

The main motive of mobile agent is to migrate computation to data rather than migrating data at the place of computation. In distributed system, lots of communication take place between autonomous systems, which not only consumes significant amount of time but also requires security. In the case of mobile agent, code gets execute locally. Since it can migrate itself to the executing environment, hence the significant amount of latencies also get reduces. As it does not require continuous connection for communication, it can work in disconnected mode.

Mobile agents can behave dynamically to an unexpected situation, if a platform is being failed, all agents executing on that platform will be alert to continue their execution on other platform.

### **1.3 Mobile Agent Applications**

Various applications of mobile agents are as follows:

Mobile agents are suitable for e-business. Mobile agent can negotiate the rate on online shopping or can compare rates with different quotes.

Mobile agent is suitable to work as a scheduler to schedule meeting on behalf of a platform. It can travel from one platform to another to schedule meeting according to the majority availability.

A secure transactional group can be made with only trusted mobile agents. The trusted platforms could let their mobile agents to meet on a secure platform. These mobile agents can purchase or sell various commodities.

Web crawler is an example of mobile agent. Information retrieval is a domain where mobile agent travels over multiple Web sites and collects the various information based on keywords

Mobile agent can work like a broadcasting and multicasting. All kind of push messages and software update can be done by mobile agent system.

Parallel processing can be done with the help of mobile agent. Any task can be divided into many parts, and each part may assign to various mobile agent. Once each subdivided task gets completed, it will be handover to the task originating platform.

With so many applications and advantages, mobile agent has security as an issue. Before deploying any agent on the platform, the originating platform should ensure about the reliability of the visiting platform. Next section discusses the security issues of mobile agent system.

### **1.4 Security Issues**

Issues in mobile agent systems can be broadly categorized as issues originating from an agent attacking an agent platform, an agent platform attacking an agent, and an agent attacking another agent [2]. They are further classified into many attacks, but proposed work deals only with denial-of-service attack. Hence, all other classifications are not discussed here.

#### **Agent Platform to Agent Attacks.**

In these attacks, platform acquires the agent's code and data. This domain includes masquerading, denial of service, eavesdropping, and alteration. These kinds of attacks are very easy as platform can access code and data of the agent. Denial-of-service attack in this category is very easy. A malicious mobile agent could trap

the visiting mobile agent and denial to provide service for which mobile agent came. It may produce intolerable delay while execution of the line of code or even forcefully terminate the execution without intimation.

### **Agent to Platform.**

In these attacks, agent acquires the execution environment of agent platform. This domain includes masquerading, denial of service, and unauthorized access. Denial-of-service attacks seem like a virus attacking over the system. A malicious agent can trigger a denial-of-service attack by acquiring excessive amounts of the platform's services and resources. A malicious agent may want to consume all computing resources of the executing platform so that the services provided by the platform cannot be used by other agents.

### **Agent to Agent.**

In these attacks, a malicious agent acquires the code and data of other agent. This domain includes masquerading, denial of service, unauthorized access, and repudiation. Denial-of-service attack is very common in this category. Malicious agent can attack over other agent with the intention of denial of its services. For example, sending multiple requests again and again will cause unwanted pressure on the request handling routines of the agent.

## **2 Related Work**

Trust and reputation have become important in security field. Many trust and/or reputation models are available recently. In this section, the most popular ones are elaborated.

In [3], authors suggest TRUMMAR as a reputation model in mobile agent systems where the reputation is computed on the basis of loss of reputation with time, prior-derived reputation, first impression, hierarchy of platforms, and reported reputations from other platforms that wish to volunteer for information. The reputation value is calculated as

$$\begin{aligned} \text{RepA/B}(0) = & \text{repA/B} + q \left( \sum \alpha_i \text{repA/B}_i \right) / \left( \sum \alpha_i \right) \\ & + r \left( \sum \beta_i \text{repA/B}_i \right) / \left( \sum \beta_i \right) + s \left( \sum \delta_i \text{repA/B}_i \right) / \left( \sum \delta_i \right) \quad (1) \end{aligned}$$

where

- $\text{RepA/B}(0)$  is reputation being computed now of A with respect to B.
- $\text{RepA/B}$  is last computed reputation of A in view of B.
- $\sum \alpha_i \text{repA/B}_i$  is the weighted sum of reputations of A as informed by the one hop distance of B ( $B_i$ ).
- $\sum \beta_i \text{repY/X}_i$  is the weighted sum of reputations of A as informed by the nodes that are more than two hops distance of B ( $B_j$ ).

- $\sum \delta_i \text{rep}_A/B_i$  is the weighted sum of reputations of A as informed by unknown node in the network that voluntarily to give reputation of B.
- $\delta, \beta, \alpha$  are weighing factors computed according to the reputation of the individual one-hop distance nodes, more than two-hop distance node, and unknown in the host network, respectively.
- p, q, r, and s are weighing factors corresponding to the previous reputation of A in view of B, one-hop distance nodes of B, nodes greater than two-hop distances of B, and alien in the agent network.

Cubaleska and Schneider [4] propose a mathematical model for a posteriori identification of malicious platform to develop a trusted system. On the basis of trust value of originating host to other, it can either prepare an itinerary in which reliable platform to be visited are mentioned, or it can choose the platform.

In [5], authors proposed a method for computing reputation on the basis of position of every person of a community within the social network. Local information is used to compute reputation value. A different algorithm, node ranking, was proposed to obtain such computations.

In [6], authors proposed Sporas as a reputation model in mobile agent system, where the reputation is calculated recursively and latest rating gets more weight. Hence, reputation rating at time i,  $R_i$ , is computed recursively from the previous reputation  $R_{i-1}$  and the purchase rating  $P_i$  as:

$$R_i = R_{i-1} + (1/\theta) \cdot \Phi(R_{i-1}) \cdot (P_i - R_{i-1}) \quad (2)$$

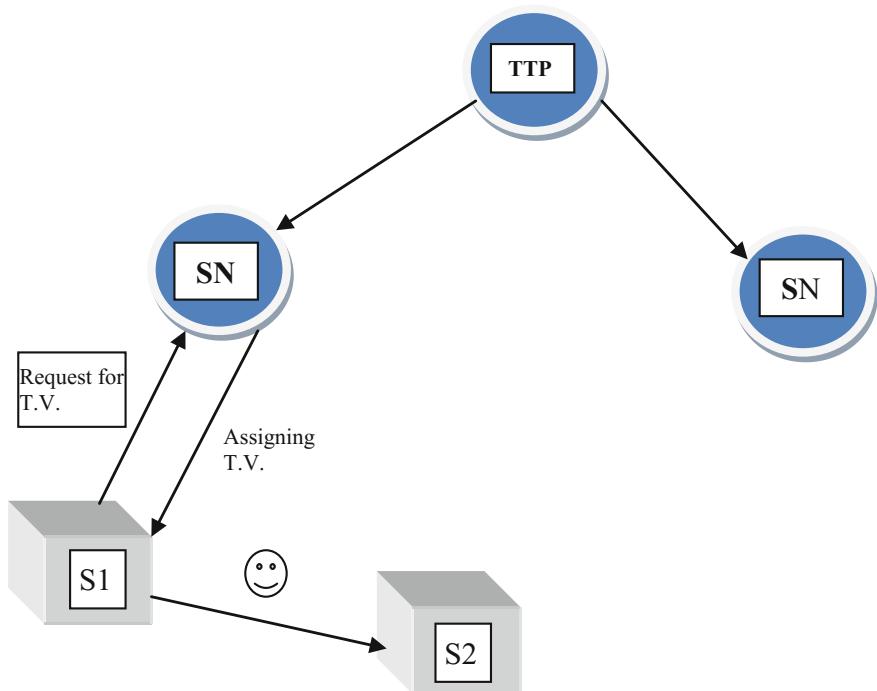
where  $\theta$  is the ratings,  $P_i$  is a rating taken from user i, and function  $\Phi$  is defined to gradually decrease the value for reputable users as:

$$\Phi(R_{i-1}) = 1 - 1/(1 + e^{-(R_{i-1} - M)/\sigma}) \quad (3)$$

where  $M$  is the maximum reputation and  $\sigma$  is the accelerating factor of the function  $\Phi$ . So, lesser the value of  $\sigma$ , the steeper the damping factor  $\Phi(R)$ .

### 3 Proposed Model

Designing a reliable and secure trust and/or reputation model for the execution of line of code without denial-of-service attack is a new research area. In this section, a model for computing the trust and reputation value is presented. Let us consider the situation in Fig. 1, where an agent platform S1 wants to send a mobile agent MA1 on another platform say S2 in order to execute line of code for the task. Platform S1 will not send the agent unless it is sure that platform S2 is trusted one, i.e., that platform S2 will provide a trusted environment which will not masquerade, alter or edit the agent's code, data, or status. In the mechanism of finding the trusted platform,



**Fig. 1** The proposed model (trust and reputation model)

platform S1 will calculate a reputation value for the platform S2. This reputation is calculated by taking the average of all its mobile agent's trust values.

All trust and reputation values will be maintained by some centralized supporting nodes (SN) which are behaving like a trusted third party (TTP). Since the mobile agent has to work in distributed environment, hence the number of supporting nodes (SN) varies according to the geographical area. Figure 2 shows the process in trust and reputation model.

In this model, information about the trust and reputation can be retrieved by any SN. These collected trust and reputation values then aggregated for choosing the right platform for the execution of the mobile agent. Once the execution gets over, the values of trust and reputation get modified according to the feedback given by executing platform.

### Algorithm

Input: TTP, SN, S1, S2, MA1

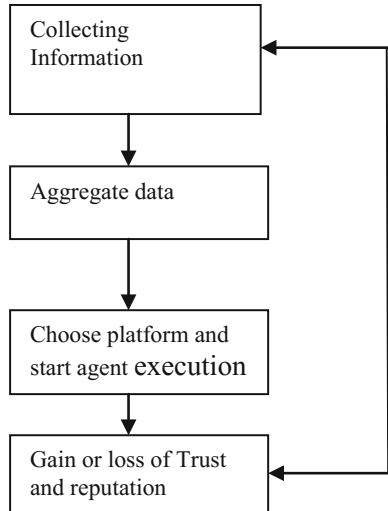
Output: Reputation and trust values to S1, S2, and MA1

Step 1: S1 is creating a mobile agent MA1

Step 2: S1 will send its unique address (MAC) to Supporting Node SN

Step 3: SN will check S1's unique address (MAC) and compute trust and reputation values for S1

**Fig. 2** The process in the model (flow chart)



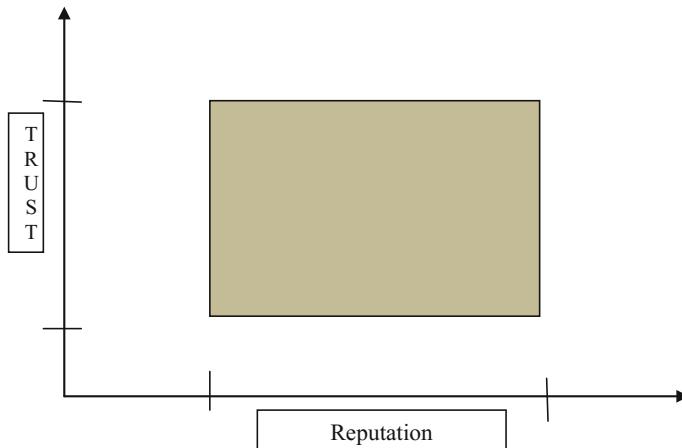
- (a) Assigning initial trust value to platform (If S1 is gaining trust and reputation values first time)
  - Initial trust value will lie between  $\alpha$  and  $\beta$ , where  $\alpha$  is initial value of the trust value, which is considered to be minimum and  $\beta$  is maximum value of the trust value a mobile agent can have.
  - Initial reputation can be calculated by taking the average of the trust values of all mobile agents under that originating platform.
- (b) Increment and decrement of trust and reputation values on the basis of type of services performed, i.e., read and write.
  - Increment in trust value on successful execution of services (reported by executor)
    - I. If service performed is read, then its trust value get increased by  $\gamma$
    - II. If service performed is write, its trust value get increased by  $\delta$ 
      - Decrement in trust value on execution of services more than the number of times allowed (reported by executor)
  - I. If service performed is read, then its trust value get decreased by  $\gamma$
  - II. If service performed is write, its trust value get decreased by  $\delta$

Step 4: Computing the maximum number of service request for preventing DoS. Maximum number of request Maxrq can be calculated as follows:

$$\text{Maxrq} = M \times \left( \frac{\text{TV}(S1)}{\sum \text{TV}_{Si}} \right) \quad (4)$$

where M is the total number of services and a platform S1 can perform over a time interval.

Step 5: Declaring the denial-of-service attack



**Fig. 3** Trust and reputation thresholds

Once the trust value of any mobile agent gets less than  $\alpha$ , the mobile agent will be declared as a malicious agent and its contribution in reputation of its originator will be canceled. Ultimately, the reputation of the originator gets reduced.

In Fig. 3, shaded area shows the trusted and reputed platform over which the mobile agent is secure from denial-of-service attack. Reputation is maintained by taking the average of trust values of all mobile agents of a platform.

$$\text{Rep}(S1) = \sum TV_{MAi}/n \quad (5)$$

where  $n$  is total number of mobile agents generated by  $S1$ .

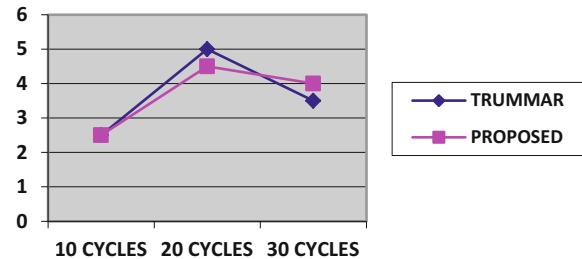
If the trust value of any mobile agent gets increased, then reputation of its originator automatically increases. Similarly, the decreased in trust value of any mobile agent reduces the reputation of its originating platform.

## 4 Results and Analysis

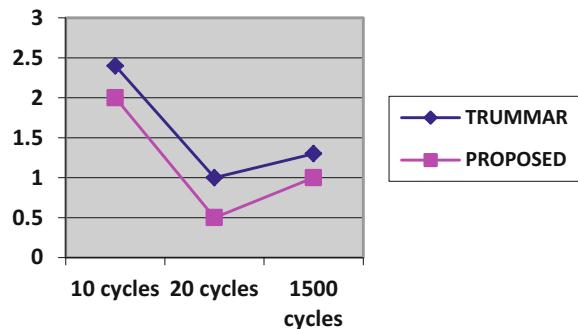
In order to compare proposed model with other trust and reputation model, consider the environment shown in Fig. 1.  $S1$  and  $S2$  are autonomous platform capable enough to create mobile agents and provide execution environment to them. Supporting nodes (SN) provide replica of database stored at trusted third party. As mobile agent works in distributed network; hence, the supporting nodes (SN) are situated at various geographical regions.

The table for maintaining the data for agent's trust and platform's reputation is maintained by SN, and it gets updated on the basis of agent's platform interaction.

**Fig. 4** Behaviour of reliable platform's reputation



**Fig. 5** Behavior of malicious platform



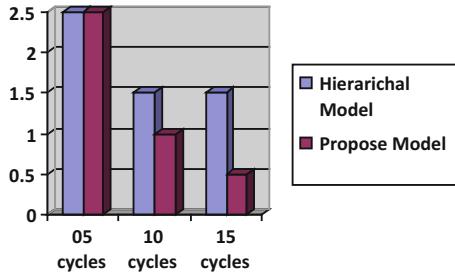
To verify proposed model, implementation is done in JADE. The system consists of four platforms S1, S2, S3, and S4. Only one communication can occur at any time. The minimum trust value for any mobile agent is 0.5, and the maximum trust value for an agent is 5. Similarly if a platform is found “untrustworthy,” its corresponding reputation value gets reduced by the trust value of the agent, otherwise its value gets increased. The trust value of a reliable agent increases gradually from an initial value from 0.5 to a value around 5. An agent’s trust value decays exponentially with respect to the number of execution. However, the reputation of the platform depends on average trust value of all its mobile agents.

A comparison between the TRUMMAR [3] and proposed model is shown in Fig. 4, where x-axis shows the number of times the services are demanded from platform and y-axis shows the reputation values of the platform.

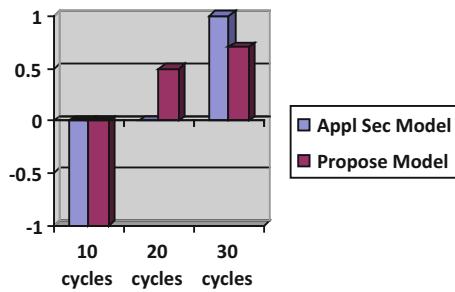
Model proposed in [3] increases and decreases the reputation value more drastically but proposed model, changes reputation value gradually as shown in Fig. 4.

The reputation of malicious platform decays from initial value to a value less than 0.5. The same effect that was analyzed for reliable platform is also analyzed for malicious platform. Figure 5 shows that the reputation value increases gradually, once it reaches to its minimum.

Model proposed in [7] uses group in-charge and itinerary to detect malicious host. The proposed model uses the reputation value for detecting malicious host. The proposed model consumes less time in comparison with model in [4] as shown in Fig. 6.



**Fig. 6** Comparison between two models



**Fig. 7** Comparing reputation values for two models

Model proposed in [8] uses trust value as (HT, T, ND, U, HU) from  $[-1, 1]$  scale. It builds a trust relationship with security requirement and security mechanism. The proposed model is more accurate as it computes trust in discrete values, while model in [8] uses range to check trustworthiness of an agent. Figure 7 shows the results.

## 5 Conclusion

Proposed model is a trust and reputation-based model that mobile agent system can implement to avoid denial-of-service attacks from agent-to-platform as well as platform to agent. Numerous of trust and/or reputation models have been previously implemented in research, but proposed model can handle both agent-to-platform and platform to agent denial-of-service attacks.

## References

1. Lange, D.B., & Oshima, M. *Introduction to mobile agents*. General Magic Inc., Sunnyvale, California, USA.
2. Jansen, W., & Karygiannis, T. (1998). Mobile Agent Security, U.S. Department of commerce, NIST, Spec. Publ. 800-19, p. 44 (Oct. 1999), Gaithersburg, MD 20899-8930.
3. Derbas, G., Kayssi, A., Artail, H., & Chehab, A. (2004). Trummar-a trust model for mobile agent systems based on reputation. In *Proceedings of the IEEE/ACS International Conference on Pervasive Services, 2004 (ICPS 2004)* (pp. 113–120). IEEE.
4. Cubaleska, B., & Schneider, M. (2002). Applying trust policies for protecting mobile agents against DoS. In *Third International Workshop on Policies for Distributed Systems and Networks (POLICY)*, Moterey, California.
5. Pujol, J. M., Sangüesa, R., & Delgado, J. (2002). Extracting reputation in multi agent systems by means of social network topology. In *Proceedings of the First International Joint Conference on Autonomous Agents and MultiAgent Systems*, Bologna, Italy.
6. Zacharia, G., & Maes, P. (2000). Trust management through reputation mechanisms. *Applied Artificial Intelligence*, 14, 881–907.
7. Aggarwal, M., & Nipur, P. (2012). Hierachal model to prevent DoS attack in mobile agents. In *International Conference in Recent Trends in Information Technology and Computer Science (ICRTITCS-2012) Proceedings published in International Journal of Computer Applications®(IJCA)* (pp. 0975–8887).
8. McDonald, J. T., & Yasinsac, A. (2006). Application security models for mobile agent systems. *Electronic Notes in Theoretical Computer Science*, 157(3), 43–59; In *Proceedings of the First International Workshop on Security and Trust Management (STM 2005)*.
9. [www.tryllian.com/technology-advancement-with-mobile-agentssoftware/12/05.2013](http://www.tryllian.com/technology-advancement-with-mobile-agentssoftware/12/05.2013).
10. Márrom, F. G., & Pérez, G. M. (2010). Towards pre-standardization of trust and reputation models for distributed and heterogeneous systems. *Computer Standards & Interfaces*, 32(4), 185–196.

# Fraud Detection in Online Transactions Using Supervised Learning Techniques



Akshi Kumar and Garima Gupta

## 1 Introduction

In today's world, as new technologies are emerging, the threat of online exploitation is also increasing. This exploitation can be monetary, psychological, sexual, etc. In almost every facet of development, there is a constant threat of getting duped in one way or the other. There can be many reasons for an adversary to threaten an individual, or an organization, like personal animosity, monetary gain, defamation, demeaning reputation. Such events relate to the term "fraud." Fraud is any illicit activity by an entity that causes loss to an individual or an organization [1]. Fraud can occur in various domains and in various ways. An efficient way to deal with fraud and the fraudsters is to understand and implement fraud analytics [2]. It helps in detecting, protecting, avoiding, and mitigating fraud. This paper discusses detecting the fraud in the domain of online transactions. Fraud detection is an activity wherein, fraud can be proactively identified and detected for any malicious activity that has taken place causing any kind of loss to the target entity [1].

Millions of online transactions take place every day, and all these transactions are subjected to various kinds of frauds. Such transactions encompass any monetary exchange done online. There are various kinds of fraudulent online transactions like credit card transaction fraud, bank statement fraud, insurance fraud, automated transaction fraud in banks [3]. This paper aims to expound fraud detection using supervised machine learning techniques in publicly available credit card dataset.

Whenever a real-time pattern is assessed and analyzed, manual dependency becomes impossible due to large sizes of the databases. Hence, the concept of

---

A. Kumar (✉) · G. Gupta  
Delhi Technological University, New Delhi 110042, Delhi, India  
e-mail: akshikumar@dce.ac.in

G. Gupta  
e-mail: garimacsdtu@gmail.com

machine learning [4] was introduced. Here, the dataset is given as input into the system and it intelligently computes, iterates, improves, and presents the desired result almost instantaneously, depending on the size of the dataset. In the work done, supervised learning technique [5] has been applied on the input data. In this technique, the system is presented with training data and the desired output. It then intelligently, after performing computations, reaches the optimum solution. This technique has been used to calculate the results because, since the desired output is previously known, the effectiveness of every technique can be computed and compared with each other.

In the presented work, the dataset has been processed by implementing the supervised learning techniques. The obtained results have showcased the accuracy of each technique depending on whether the resultant fraudulent transactions map to the fraudulent transactions provided in the dataset. These techniques include logistic regression, nearest neighbors, linear and RBF support vector machines, decision trees, random forest, and naive Bayes techniques. These machine learning algorithms, after being implemented on the dataset, have yielded expected results with accuracy >90%.

The paper is organized as follows: Sect. 2 discusses the similar work done in the past, i.e., in the domain of credit card fraud detection. In Sect. 3, all the supervised learning techniques used have been defined in detail. In Sect. 4, the system architecture has been provided along with the description of the dataset. The results and analysis have been discussed in Sect. 5. Here, we have explained the dataset, obtained results and their analysis. Section 6 concludes the paper with the future scope of the research.

## 2 Related Work

Kamaruddin and Ravi [6] implemented a hybrid architecture of particle swarm optimization and auto-associative neural network for one-class classification in Spark computational framework to detect credit card fraud. Santiago et al. [7] used SVM classifier to detect whether a transaction is fraudulent or not in a credit card dataset. They were successfully able to identify forty to fifty percent of the fraudulent transactions in a month. Gómez et al. [8] used artificial neural networks for detecting fraudulent credit card transactions and reducing data unbalancedness. They also evaluated the cost metrics for the results obtained. Bhattacharyya et al. [9] detected credit card fraud by implementing linear regression, support vector machine (SVM), and random forest. They found out that random forest gave better overall accuracy compared to the other two techniques. Quah and Sriganesh [10] have worked in credit card fraud detection using clustering and filtering capabilities of self-organizing maps (SOM). Panigrahi et al. [11] worked on credit card fraud detection by amalgamating four techniques, i.e., rule-based filter, Dempster–Shafer adder, transaction history database, and Bayesian learning. They first segregated the suspicious transactions by checking their deviation from the expected pattern. Then such instances were

combined to compute an initial belief. Then the transactions are classified according to their degree of fraud and at last using Bayesian learning, the belief is accepted or discarded based on transaction history. Halvaee and Akbari [12] introduced a new model called artificial immune system-based fraud detection model (AFDM) for credit card fraud detection. Mahmud et al. [13] analyzed and computed various classifier algorithms on a credit card dataset and found out that meta and tree classifiers perform better than other groups of classifier.

A lot of work has been done in fraud detection but, little work has been done in implementing and comparing various supervised learning techniques to detect fraud in online transactions. In most of the research work done, authors implement a machine learning technique on a dataset and present its results, but not much has been done in comparing the heuristics on a single platform and presenting a comparative study on the same.

### 3 Models

Various supervised learning techniques have been applied, and the results have been compared to deduce the most optimum model. Table 1 briefly describes the models of supervised learning techniques that have been implemented. The key features, their advantages, and disadvantages have been presented. Each technique also describes the implementation with respect to fraud detection, i.e., how they are used to detect fraud.

### 4 System Architecture

Fraud detection primarily deals with identifying different kinds of frauds that may take place offline or in online social Web. In the work presented, fraud detection has been applied on a dataset of online transactions. Along with genuine transactions, there can be fraudulent transactions which may cause monetary and economic loss. Hence, the detection of such transactions is crucial. For carrying out this task, supervised learning techniques have been applied on the credit card dataset [14]. These techniques have successfully been able to identify the genuine transactions and the fraudulent transactions. The proposed architecture of the system has been presented in Fig. 1.

To implement the techniques, Python 3.6.3 platform is used. The libraries used are “pandas,” which is used for reading the dataset file, “matplotlib,” for plotting graphs and confusion matrices and “sklearn” to implement machine learning techniques.

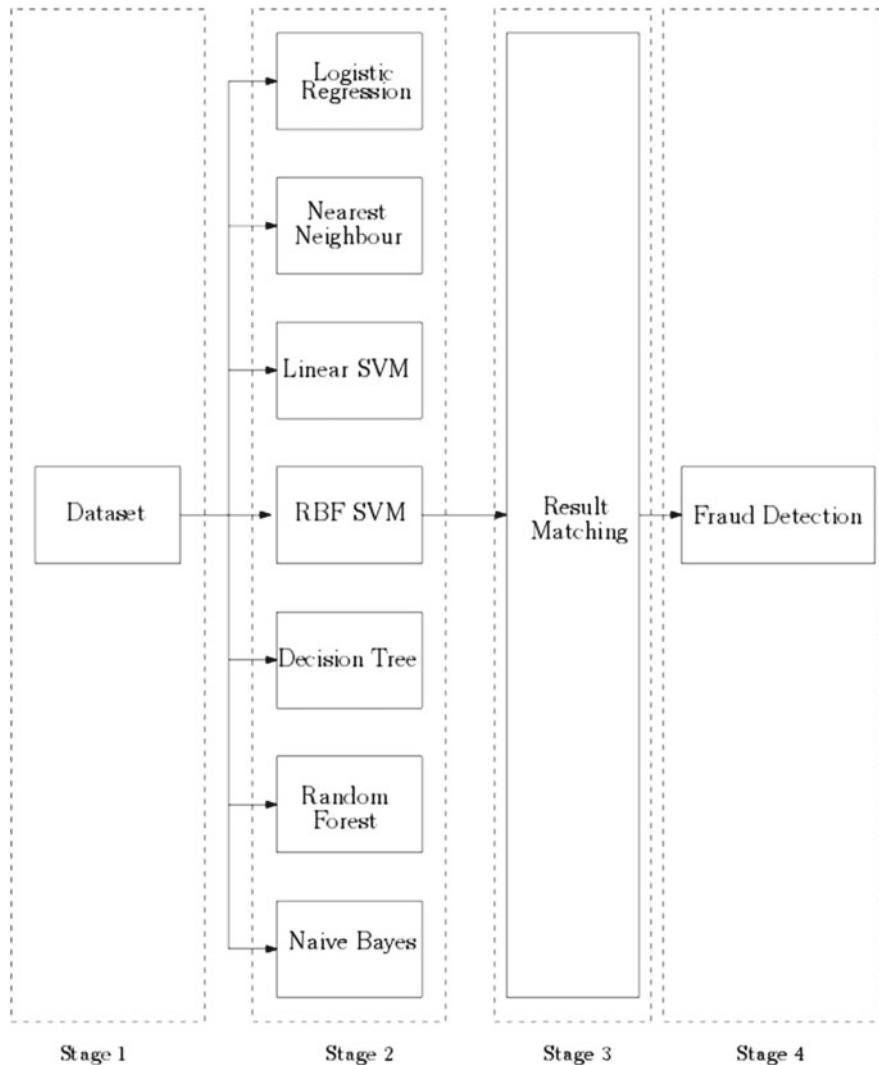
**Table 1** Description of the models used in the paper

Algorithm	Key features	Advantages	Disadvantages	Implementation in fraud analytics
Logistic regression	In logistic regression, the probability of the given instance either belonging to class “1” or belonging to class “0” is predicted. Afterward, the output is transformed to take discrete values 0 or 1 using the sigmoid function model	This method is used to predict discrete variables	Linear regression output values in a continuous range because of which we cannot use it to predict binary variables	In this paper, it is predicted if a given instance of data corresponds to fraud (represented by class “1”) or does not correspond to fraud (represented by class “0”)
Nearest neighbors	This method classifies an instance of the dataset based on the labels assigned to the k-nearest training examples (Euclidean distance is most commonly used metric)	It handles multi-class cases on its own	It has a high computation cost, and we must have a meaningful distance function	For every instance to test, k-nearest training instances in the feature space is found out using Euclidian distance as metric. If most of them are labeled as “1”, then the model will conclude that the given transaction is fraudulent. Similarly, if most of them are labeled as “0”, the model will conclude that the given transaction is not fraudulent
SVM	SVMs or support vector machines in the training process construct a hyperplane (surface of dimension n-1 in a n dimension space), which optimally classifies the training data. The hyperplane is then used to determine the labels of the test data. Kernels in SVM perform nonlinear transformation of the data. Kernels can transform the feature space in which SVM operates	It is useful in cases where data points are inseparable by a hyperplane in the original feature space. In that case, kernels can transform the feature space into a space of higher dimension where the data points are separated by a hyperplane. RBF kernel transforms the feature space into a space which is infinite dimensional	The selection of kernel function parameter is the biggest limitation of SVM [15]	The dataset has 28 features. SVM will construct a hyperplane of dimension 27. This hyperplane will divide the data into two parts optimally such that one part has mostly fraud instances and the other has non-fraud instances. When an instance has to be tested, it tests which side of the hyperplane that particular instance lies on. It then labels that instance accordingly

(continued)

**Table 1** (continued)

Algorithm	Key features	Advantages	Disadvantages	Implementation in fraud analytics
Decision trees	In decision trees, every internal node of the tree represents a decision and it splits the tree into branches based on a condition. Every decision involves two criteria which feature to choose and the condition on those features. The leaf node of the tree represents a decision on the label of the test instance	The key advantage of this method is that nonlinear relationships between parameters do not affect the tree performance	The decision tree becomes unstable if any data value in the dataset is modified	The leaf node labels the test instance as fraud “1” or not fraud “0”. There are many ways to split the tree in branches. The model optimizes the structure of the tree during the training process. In this paper, recursive binary splitting is used for training which chooses the split with the best cost
Random forest	Random forest is an ensemble learning method. During training, it randomly selects a subset of the training set and creates decision tree for that subset. This process is repeated multiple times creating multiple decision trees in the training process. While testing, the result of each decision tree is evaluated and then the random forest model outputs the mode of classes predicted by the decision tree	Random forest is the most accurate learning algorithm and produces highly accurate classifiers	This method does not work accurately if the independent and dependent variables share a linear relationship	In the paper, this algorithm randomly selects a subset of the dataset and creates multiple decision trees and the result decides whether a given transaction is fraudulent “0” or non-fraudulent “1”
Naïve Bayes	This method is a classification technique and works on the concept of Bayes’ theorem. Each data point in the dataset is considered independent. This method works on the technique that, given that the output of a data point is true, the probability of that data point being true is maximized	This method needs less training data as if the naïve Bayes independence assumption is true, then it will converge quickly	Native Bayes classifier makes a very strong assumption when data points in a dataset are independent of each other	In the dataset used, we are provided with the output and based on the correctness of that output, using naïve Bayes, it is predicted whether a given transaction is fraudulent or not



**Fig. 1** System architecture for performing the comparative study

The system architecture has been broken down into four stages pertaining to each component of the workflow:

### 1. Input Preprocessing

In this stage, the credit card fraud dataset is given as input into the system. Modifications are performed on the feature space of the dataset. This is done to optimize the results of the machine learning algorithms which are applied in the next stage.

## 2. Processing

The output of the input processing stage is taken, and it is used as input for this stage. For every data point in the input dataset, the label for every transaction is computed. The label defines every transaction as fraudulent or non-fraudulent. This is done for each machine learning technique.

## 3. Result Matching

In this stage, the labels are obtained from the previous processing stage. These labels are then, compared with the ground truth to evaluate the accuracy of each technique.

## 4. Fraud Detection

Based on the accuracy of the techniques found in the previous stage, the optimal technique is selected and the labels (fraudulent or non-fraudulent) are reported for every data point.

### **4.1 Dataset**

The credit card transaction dataset [14] contains the details of transactions made by credit cards in September 2013 by European cardholders. The occurrence of these transactions spans two days. There are a total of 284,807 transactions, out of which, 492 are fraudulent. This data has gone through principal component analysis (PCA) transformation. The dataset contains 28 features, where “time” is the time elapsed between each transaction and the first transaction in the dataset. The feature set class defines the output (0 or 1), whether the transaction is fraudulent or not.

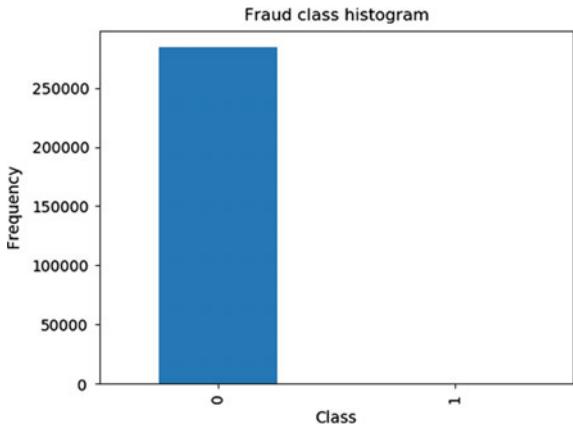
## **5 Results and Analysis**

In this section, the results obtained after applying various the supervised learning techniques on the dataset have been discussed. Furthermore, the efficiency of each technique has been calculated.

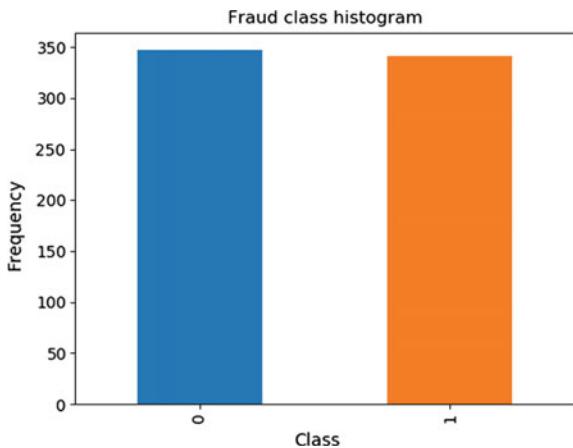
### **5.1 Results**

After applying the machine learning techniques discussed in the paper, the accuracy of each method and the values of various metrics, i.e., precision, recall, f1-score, and support have been calculated. Figure 2 represents the fraud class histogram of all the data points in the dataset. Here, the output has been classified as either 0 (non-fraudulent transaction) or 1 (fraudulent transaction). The graph has been plotted on a

**Fig. 2** Fraud class histogram of the entire dataset



**Fig. 3** Fraud class histogram of under-sampled data



frequency of transactions ranging from 0 to 284,807 and the classified output (0 or 1). Due to such a large number of transactions, the number of fraudulent transactions is scarcely visible. As a result, the dataset has been under-sampled. For this, 350 random points have been taken and the results have been calculated. As shown in Fig. 3, the transaction frequency ranges from 0 to 350.

Further, the accuracy of each supervised learning technique used has been calculated and listed in Table 2.

In addition to this, the metrics for the results obtained have also been computed. These metrics are precision, recall, f1-score, and support. Tables 3, 4, 5, 6, 7, 8, 9 enlist the values of aforementioned metrics for Logistic Regression, Nearest Neighbors, Linear SVM, RBF SVM, decision trees, random forest, and naïve Bayes, respectively, as shown in Table 3 for logistic regression. The confusion matrices have also been realized in Figs. 4, 5, 6, 7, 8, 9, 10 for logistic regression, nearest neighbors, linear SVM, RBF SVM, decision trees, random forest and naïve Bayes, respectively, to describe the performance of model on the given dataset, when the output is known.

**Table 2** Accuracy of techniques used

Supervised learning technique	Accuracy
Logistic regression	93.60% ( $\pm 4.28\%$ )
Nearest neighbors	92.58% ( $\pm 3.97\%$ )
Linear SVM	93.02% ( $\pm 4.40\%$ )
RBF SVM	92.73% ( $\pm 3.09\%$ )
Decision trees	90.26% ( $\pm 2.42\%$ )
Random forest	93.31% ( $\pm 2.23\%$ )
Naïve Bayes	91.71% ( $\pm 3.20\%$ )

**Table 3** Metrics computation for logistic regression

	Precision	Recall	f1-score	Support
0	0.93	0.98	0.96	144
1	0.98	0.93	0.96	152
Avg/Total	0.96	0.96	0.96	296

**Table 4** Metrics computation for nearest neighbors

	Precision	Recall	f1-score	Support
0	0.93	0.99	0.96	144
1	0.99	0.93	0.96	152
Avg/Total	0.96	0.96	0.96	296

**Table 5** Metrics computation for linear SVM

	Precision	Recall	f1-score	Support
0	0.94	0.97	0.95	144
1	0.97	0.94	0.95	152
Avg/Total	0.96	0.96	0.96	296

**Table 6** Metrics computation for RBF SVM

	Precision	Recall	f1-score	Support
0	0.94	0.98	0.96	144
1	0.98	0.94	0.96	152
Avg/Total	0.96	0.96	0.96	296

**Table 7** Metrics computation for decision trees

	Precision	Recall	f1-score	Support
0	0.92	0.90	0.91	144
1	0.90	0.93	0.92	152
Avg/Total	0.96	0.96	0.96	296

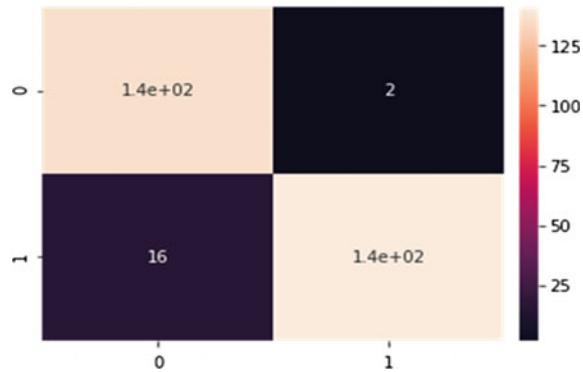
**Table 8** Metrics computation for random forest

	Precision	Recall	f1-score	Support
0	0.93	0.99	0.96	144
1	0.99	0.93	0.96	152
Avg/Total	0.96	0.96	0.96	296

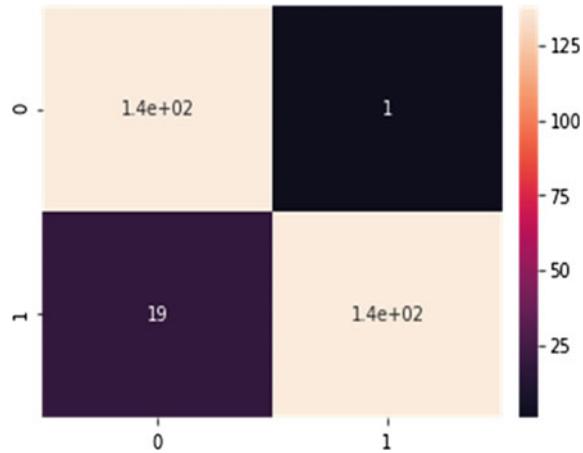
**Table 9** Metrics computation for Naïve Bayes

	Precision	Recall	f1-score	Support
0	0.90	0.99	0.94	144
1	0.99	0.90	0.94	152
Avg/Total	0.96	0.96	0.96	296

**Fig. 4** Confusion matrix for logistic regression



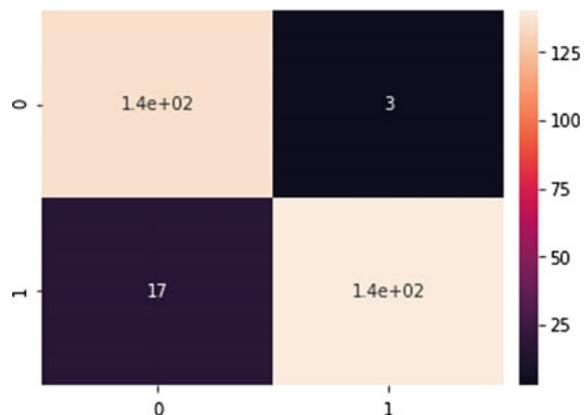
**Fig. 5** Confusion matrix for nearest neighbors



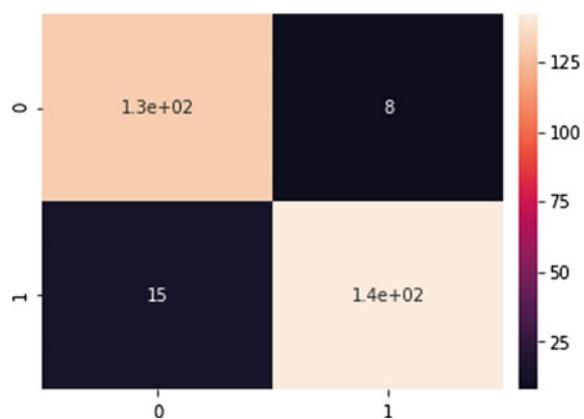
## 6 Conclusion

This paper aimed to automate prediction of fraud in an online credit card transaction dataset. Supervised learning techniques, namely logistic regression, nearest

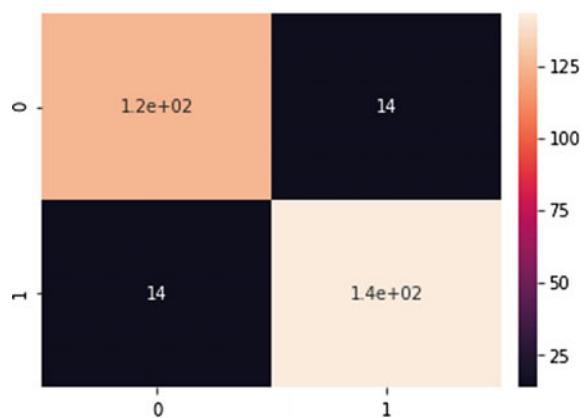
**Fig. 6** Confusion matrix for linear SVM



**Fig. 7** Confusion matrix for RBF SVM



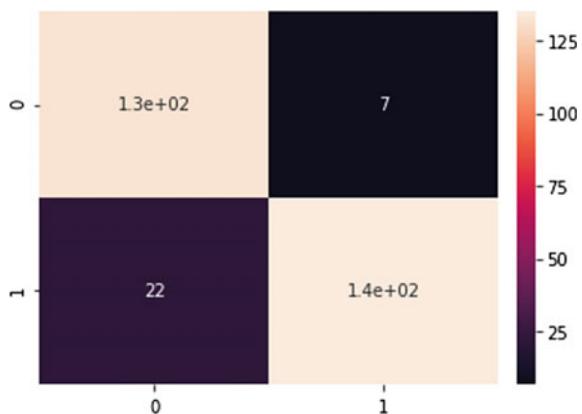
**Fig. 8** Confusion matrix for decision trees



**Fig. 9** Confusion matrix for random forest



**Fig. 10** Confusion matrix for naïve Bayes



neighbors, linear SVM, RBF SVM, decision trees, random forest, and naïve Bayes were implemented. The performance of each technique was analyzed and compared. Logistic regression proved to be the best among the others.

As a future scope of this research, the use of neural networks as well as other supervised, unsupervised, and reinforcement learning techniques can be explored.

## References

1. Abdallah, A., Maarof, M. A., & Zainal, A. (2016). Fraud detection system: A survey. *Journal of Network and Computer Applications*, 68, 90–113.
2. Makki, S., Haque, R., Taher, Y., Assaghir, Z., Ditzler, G., Hacid, M. S., & Zeineddine, H. (2017). Fraud analysis approaches in the age of big data-a review of state of the art. In *2017 IEEE 2nd International Workshops on Foundations and Applications of Self\* Systems (FAS\*) W* (pp. 243–250). IEEE.

3. West, Jarrod, & Bhattacharya, Maumita. (2016). Intelligent financial fraud detection: a comprehensive review. *Computers & Security*, 57, 47–66.
4. Christopher, B. M. (2006). *Pattern recognition and machine learning*. Springer.
5. Baesens, B., Van Vlasselaer, V., & Verbeke, W. (2015). *Fraud analytics using descriptive, predictive, and social network techniques: A guide to data science for fraud detection*. Wiley.
6. Kamaruddin, S., & Ravi, V. (2016). Credit card fraud detection using big data analytics: use of psoaann based one-class classification. In *Proceedings of the International Conference on Informatics and Analytics*, p. 33. ACM.
7. Santiago, G. P., Pereira, A., & Hirata Jr, R. (2015). A modeling approach for credit card fraud detection in electronic payment services. In *Proceedings of the 30th Annual ACM Symposium on Applied Computing* (pp. 2328–2331). ACM.
8. Gómez, J. A., Arévalo, J., Paredes, R., & Nin, J. (2018). End-to-end neural network architecture for fraud scoring in card payments. *Pattern Recognition Letters*, 105, 175–181.
9. Bhattacharyya, S., Jha, S., Tharakunnel, K., & Westland, J. C. (2011). Data mining for credit card fraud: A comparative study. *Decision Support Systems*, 50(3), 602–613.
10. Quah, J. T., & Sriganesh, M. (2008). Real-time credit card fraud detection using computational intelligence. *Expert Systems with Applications*, 35(4), 1721–1732.
11. Panigrahi, S., Kundu, A., Sural, S., & Majumdar, A. K. (2009). Credit card fraud detection: A fusion approach using Dempster-Shafer theory and Bayesian learning. *Information Fusion*, 10(4), 354–363.
12. Halvaiee, N. S., & Akbari, M. K. (2014). A novel model for credit card fraud detection using artificial immune systems. *Applied Soft Computing*, 24, 40–49.
13. Mahmud, M. S., Meesad, P., & Sodsee, S. (2016). An evaluation of computational intelligence in credit card fraud detection. In *2016 International Computer Science and Engineering Conference (ICSEC)* (pp. 1–6). IEEE.
14. <http://mlg.ulb.ac.be/>.
15. Auria, L., & Moro, R. A. (2008). *Support vector machines (SVM) as a technique for solvency analysis*.

# Part V

## Computing in Education

**Dr. Pinaki Chakrabarty Section Editor**

### **Editorial**

Computers were first used widely as an instructional aid during the 1960s. Over the years, their use in the field of education has increased manifold. The availability of interactive computer systems since the 1980s, the advent of the World Wide Web in the 1990s and the ushering in of Massive Open Online Courses during the 2000s revolutionized the academia. Today computers are an integral part of the life of a student anywhere in the world. This part presents some interesting research works that use extensible and adaptable methods to make education more effective, scalable and personalized.

The first chapter by Dadhich and Dutta describes a low-cost mobile app that can operate in the off-line mode to convert printed text to speech in real time and help visually impaired persons learn on the move with minimal manual intervention. Instructors now try to make the process of teaching and learning more efficient using appropriate software tools. Sharma *et al.* apply the notion of graph centrality to identify the most important central keywords in a document. They utilize WordNet as a knowledge source to build a semantic network of the terms appearing in a document. Their work demonstrates that graph centrality measures, such as degree, betweenness, PageRank and closeness, can be used as a simple yet powerful means of extracting keywords, but caution against bias towards broader terms. Vijayalakshmi and Hota propose a novel algorithm for judicious task assignment in a crowdsourcing environment based on the online trustworthiness of participants as assessed from their knowledge and belief. They demonstrate promising results of their scheme when applied to a social media application that the authors developed for task assignment among educationists.

Srinivas *et al.* extend the virtual machine for the Scratch programming language by logging real-time data streams of fine-grained transactions as a series of time-stamped learning steps. They develop a visualization tool for instructors to monitor students' progress and identify learning gaps. Computers and the Internet have virtually redefined the concept of collaborative learning. Narbutaite *et al.* report an educational approach where robots can be gainfully employed to teach

STEM and interdisciplinary subjects. They present an encouraging case study of teaching robot programming with their proposed collaborative teamwork approach over a period of 5 years in the Kaunas University of Technology, Lithuania.

Smartphones have enormous scope in education and education management. Jain and Kumar developed a freely downloadable mobile app to teach various sorting and searching algorithms to high school and undergraduate students. Assessment from actual usage and a questionnaire-based survey to evaluate presentation, performance and usage reveal positive feedback among the users. Mulatu *et al.* present the architecture for a dynamic mobile learning application that leverages both native and Web-based client–server infrastructures and further integrates with cloud computing services. The system was developed using agile software development processes and iterative prototype testing for Android platform aided by Firebase Cloud Messaging. The hybrid learning system empowers anyone with lifelong learning through an informal environment by delivering any educational content anywhere and anytime.

### **Section Reviewers:**

Anand Vijayalakshmi

Ahmad Firdaus Bin Zainal Abidin

Anu Saini

Anupama Jha

Anupama Kaushik

Cesar A. Tecson

Chesta Agarwal

Dipika Jain

Divya Chaudhary

Gaurav Indra

Kamlesh Dutta

Kanika

Kanika Bhatia

Mamta Mittal

Manpreet Kaur

Manpriya Kaur

Maria Cristina Rodriguez-Sanchez

Minni Jain

Mukesh Sahu

Nenad Jovanovic

Nupur Chugh

Pinaki Chakraborty

Poonam Rani

Priti Bansal

Pushp

Ranjit Rajak

Rashmi Dravid

Raza Abbas Haidri

Ruchi Sharma

Sanjay Tyagi

Savita Yadav

Shefali Arora

Shweta Taneja

Soyinka Nath

Srishti

Sulabh Tyagi

Sunny Rai

Twinkle Gupta

Veenu

Vidhi Khanduja

Vidur Katyal

Vikas Maheshkar

Viraj Kumar

Zorica Bogdanovic

# Real-Time Printed Text Reader for Visually Impaired



Ashutosh Dadhich and Kamlesh Dutta

## 1 Introduction

Learning plays a key role in an individual's overall development, and most of it comes from reading new books or documents. But these methodologies are designed to cater to the needs of sighted individuals only. Visually impaired person (VIP) needs these printed materials to be first translated into Braille format for them to read it, or at the very least, the printed material needs to be translated into digital format. The digital format is then further converted into audio format so that the information could be delivered to the VIP. Researchers have tried to tackle this problem faced by visual impaired people either by proposing real-time conversion of black–white printed text to blind readable format [1–3] or by taking the e-pdf version of the document and reproducing the result in tactile display designed for a VIP [4, 5]. Another method of presenting this information is by speaking it out to user as suggested in [6, 7]. But the main limitation with all those methods is that they all required input in a particular format to generate the output in a VIP readable format.

The following paper proposes a low-cost solution for reading for the members of the visually impaired community. The solution can be used to read any printed material written in any Latin languages like English in real time and another interesting feature of this solution is that the printed material does not have to be a page. It could be a text written on any surface, including walls and still the user could access the information through text to speech conversion as shown in Fig. 1.

---

A. Dadhich ()

Powerup Technology Inc., Río Tigris 66 int. 20 Col. Cuauhtémoc c.p., 06500 Ciudad de México,  
Mexico

e-mail: ashutosh030495@gmail.com

K. Dutta

NIT Hamirpur, Hamirpur 177005, Himachal Pradesh, India

e-mail: kdnith@gmail.com

**Fig. 1** VIP checking the timer of an electronic stove



The proposed solution is basically a smartphone application which reads aloud the contents of the document presented to it in real time. The processing delay for reading the whole page is of about 15 s only which is quite an acceptable duration. The solution could be implemented either as a handheld device (smartphone) or mounted over a stand for doing the text analysis. However, results have shown that if the person is reading a book using this solution, mounting the device over a stand as shown in Fig. 2 would lead to more accurate results. Since sometimes the user's hand might shake and clicked image might get blurred. Effort has been made to make the user interaction as accessible as possible like, for example, a physical device reset button (Volume up Button) is made available to user using which the user can stop the current task which the application is doing and ask it to do a new task (read new text).

The paper is organized as follows. Section 2 describes the related work and highlights various reading systems developed so far for the visually impaired community. In Sects. 3, the proposed system and various functionalities incorporated in the system are described. Section 4 summarizes the performance of the proposed system. Finally, Sect. 5 concludes the paper with future direction and the relevant impact of the proposed solution on the society.

## 2 Related Work

Many systems have been developed to covey the written information in books, document, etc., to VIP, either through audio or through touch, a brief analysis of them can be seen in Table 1. The first such invention came in the year 1829 [8], com-

**Fig. 2** A visually impaired individual reading a book using the system



monly known as the Braille system, developed by Louis Braille. The system uses 6 raised dots, in a  $2 \times 3$  fashion per cell, to represent a symbol. This symbol can be a punctuation mark, character or something else which gives meaning to a text. These dots as designed to be sensed using the index finger for the user to gain information. The right column is numbered from 1–3 and the rest is from 4–6. So, a total of 64 different patterns could be generated per cell. But the problem with this system was that the books which were converted into Braille were bulky, i.e., harder to carry around. Since intensive manual labor was required to convert a single page document in Braille, only a handful of books would be converted into Braille, thereby limiting the reading domain of the visually impaired user.

This problem received a technological boost in the year 1962 [1], when Davis proposed a method of tackling the problem by producing electrical signal equivalent to each character present in the printed text, although a great solution at that time, it had some issues like accuracy, storage space. Later on, in 1972, Meindl [2] worked upon increasing the accuracy by the addition of two custom MOS integrated circuits. The problem with this approach was that it only worked on black and white paper, and also since most of the VIP are not that financially strong, it required them to spend some money which is not possible for everyone to do so.

Later on, in the earlier twenty-first century, Velázquez et al. [4, 6] devised a mechanism which first converted e-books into Braille formats, stores the results in USB and later on populates the results on a Braille Tactile Display. This made the system portable and fairly accurate. But it had a limited domain due to the fact that the user could only read those books which are already available in the digital format.

**Table 1** Brief analysis of the previous and current work

Researcher name	Year	Work	Remark
Louis Braille	1829	Developed Braille system	Braille books are bulky and heavy to carry around
Davis/Meindl	1962, 1972	Generated electrical signal equivalent to each character in a printed text	Worked only for a specific set of printed text (black/white)
Velázquez	2008	Converted e-books into format compatible with Braille tactile display	Required input in specific format, i.e., digital format, plus it required the user to have access to a Braille tactile display
Jethjarurach	2014	Speak out information contained in a bar code regarding a product or place	Domain limited to products or places having bar code carefully placed for to the information to be accessible to a VIP
Sabab	2016	Speak out the words which the user touches while reading an e-book through his/her smartphone	Requires input in specific format, i.e., digital text
Proposed solution	2017	Read out any Latin-printed text on any surface in real time through smartphone device	Unable to recognize handwritten text

It fails to address the user needs when the user has to read to a printed document or has to read the labels of the products of his day-to-day use products. A somewhat similar approach was devised by Sabab and Ashmafee [7], who proposed that instead of reproducing the results on a tactile display why not use the smartphone which is available to every user and speak out the word which the user touches instead. A hybrid of both the above was proposed by Xiaoli et al. [5]. Another interesting work in this regards was done by Jethjarurach and Limpiyakorn [9] which showcased on how the quality of living can be improved and increased independence of visually impaired people through reading the contents of a product by reading its bar code.

In the present paper, the author devices a system which acts on the limitations of all the previously discussed system, i.e., the system could compile any Latin-printed text in real time and present the audio version in real time. The printed material does not necessarily have to be on a piece of paper; it can be of any size and contrast. The system also possesses zoom in/out functionality. The zoom feature enhances its

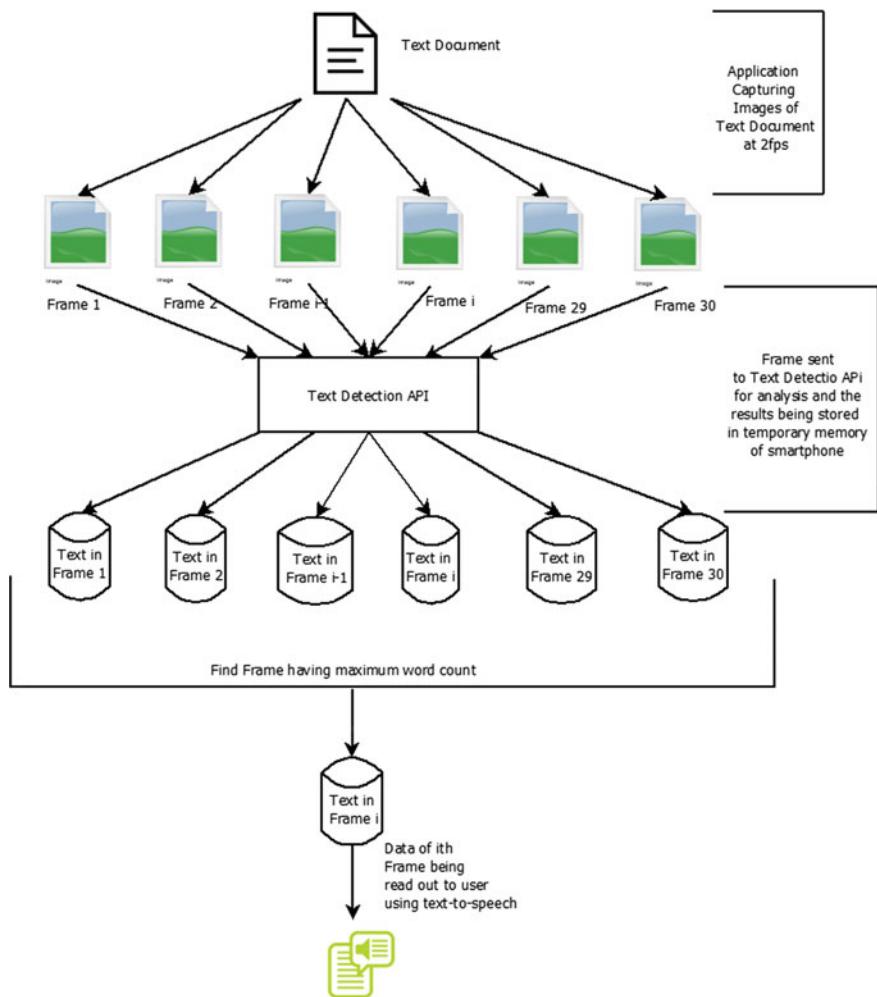
applicability and could even be used by elderly people who face difficulty in reading text written in small font. Another important feature of this solution is that it does not require any Internet connection to operate. It is designed keeping in mind the financial constraints of the visually impaired community, and the user does not have to spend money for active Internet connection to use.

### 3 Proposed System

It is said that a VIP does not see black color because he/she has never experienced black color before colors are nonexistent for them. Simply saying they do not interpret this word visually instead, they use their other senses like touch, smell, and hearing in-order to carry their day-to-day tasks. And due to their heavy reliance on these, they train them to such a level that heightened sense of touch, smell, and hearing than an average individual. The proposed solution utilizes one of these heightened senses, i.e., hearing for conveying the reproduced digital information in form of audio.

The proposed solution, as can be seen in Fig. 3, is an android application which helps the VIP to read any Latin text material printed on any surface through audio without requiring an active Internet connection. The application is designed for android smartphone devices and is designed so as to make it as user-friendly as possible for a VIP to use it. The application can be triggered using voice, i.e., user just have to say “OK Google open Blind Reader,” and the application interface is made as minimalistic as possible, i.e., there are no visual buttons present inside the application, and only two kinds of user interaction are allowed, first one is if the user single tap the smartphone screen, the application would read out whatever is presented in front of the smartphone camera and the other feature is hard reset using volume down button which may be required in case the user wants to read something else and wants to stop the processing of text currently being processed or being read out.

Although a vast number of other features could be added in the proposed system, doing would do more harm than good since this would make the application clattery and thereby rendering the application useless for VIP as can be seen in the case of (KNFB Reader App) [10, 11]. This application features similar like the solution proposed in this paper but it is a premium product and costs around Rs. 8000 [11] which could be considered too much given the fact that the average income of VIP in India is around Rs. 102,742 [12]. Other places where the proposed solutions stands out is that in KNFB Reader, the user has to manually click the image which could turn out to be quite a challenge for VIP which is not the case with the proposed solution since the user if not required to click images and the text from best out of 30 frames is picked for reading out loud to the user. Lastly, KNFB Reader as can be seen in Fig. 4 relies on Internet for its processing which could be quite a problem since Internet is still considered a luxury in many parts of the developing countries of the world which is not the case with the proposed system since it required only



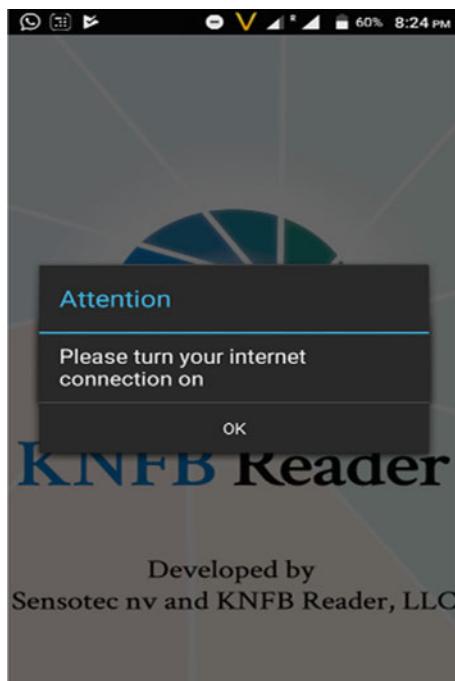
**Fig. 3** Proposed system

a one-time Internet connection to download the necessary trained models which are an integral part of the Text Detection API.

### 3.1 System Overview

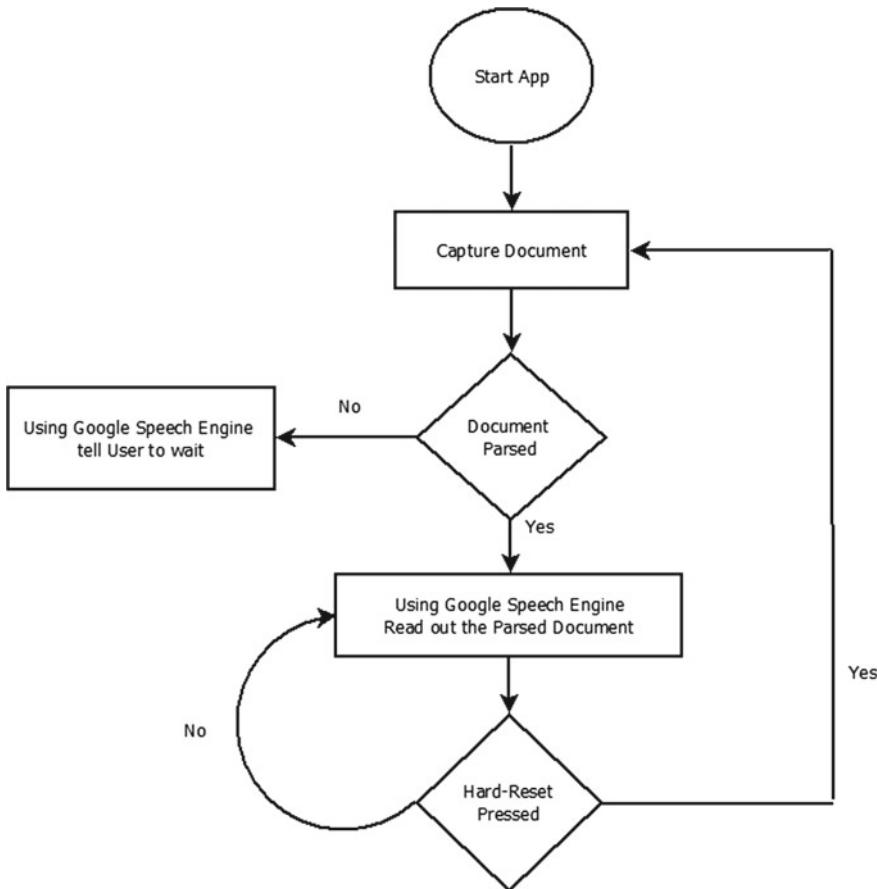
In research, a low-end android device is used for testing the smartphone application based on the proposed methodology. The android device had 1 gigabyte of main memory and runs on a 1.3 GHz quad-core Spreadtrum SC9832A processor. The

**Fig. 4** KNFB reader requesting internet access



minimum space required for running the application has been kept low keeping in mind the weak financial condition of a VIP. The system workflow can be visualized as shown in Fig. 3 where images of the text document are continuously captured and sent to Text Detection API for processing after that the best result is picked up using frame FindingAlgorithm() and after that, it is spoken out to the user using Google Speech Engine.

Once the user opens the application, it automatically starts capturing whatever reading material is presented in front of it through the camera. Images are captured at 2 fps. From there, each frame is passed onto the Text Detection API, which is stored locally inside the system. After that, results from the Text Detection API are stored in temporary memory. This process is continued till the sufficient number of frames of a particular scene is processed (the research showed that 30 frames, i.e., 15 s of a particular scene are sufficient to best judge its contents). Once we have sufficient data, the best frame is picked up and the contents are recited to the user when he taps on the screen. If he wants to stop the current recitation, then it can be done through the hard reset button (Volume Up) and can continue with some other document. One important thing to note is since it is real-time system, i.e., one that is continuously translating printed text to digital form so it has some delay associated with it. So meanwhile the application is processing the results and during that time if the user taps the screen, the system would ask user to wait sometime until the processing is completed.



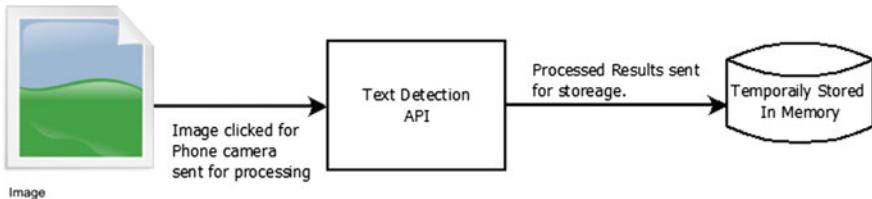
**Fig. 5** System overview

### 3.2 System Functionalities

The system, as seen in Fig. 5, comprises of three major components. Those are (1) Text Detection API, (2) Speech Engine, and (3) Best Frame Match.

#### 3.2.1 Text Detection API/Mobile Vision

Text Detection API is an offline API provided by Google [13] for detecting text in images. In video streams, once detected it uses machine perception to figure out the actual text in each block of the image and then gives the output in the form of words and lines as shown in Fig. 6. It has the ability to detect any Latin written language. Text Detection API is used in the proposed system. However, it is completely offline,



**Fig. 6** Block diagram of working of text detection API in the proposed system

i.e., won't require any active Internet connection to use it. Experiment with this API has shown that for a smartphone device having an eight-megapixel camera, and having sufficiently good resolution, it is possible to detect those printed texts which otherwise would require for a normal person a magnifying glass to read.

### 3.2.2 Text to Speech (TTS)

The system uses Google Speech Engine for doing its text to speech analysis. Since the prototype application is running on an android phone, using the by default provided Google speech for reading out loud the generated text would seem an obvious choice. The use of default speech engine won't increase the size of the application thereby not affecting the performance of the system in lower-end devices [9].

### 3.2.3 Best Frame Match

The authors believe that it would be unwise for the VIP to first capture an image of the document and then process it. There is a likelihood that the image may not have been captured correctly. So, the novel idea has been implemented in which the system does the heavy lifting for the user. The user points his smartphone toward the screen, and in the meanwhile, the system clicks approximate 30 images at the rate of 2 fps. Out of these 30 images, the frame having maximum word count is considered for further processing, i.e., for the conversion of text to speech, as shown in Fig. 3.

## 4 Evaluation and Result Analysis

For the evaluation, the algorithm in the proposed solution is compared with the digital text, generated through Text Detection API. Following metrics are used for comparison:

1. Percentage match of the generated text with actual text.
2. Time taken for text conversion in both the above cases.

**Table 2** Accuracy comparison between mobile vision and proposed solution

Total words	Mobile vision (% accuracy)	Proposed solution (% accuracy)
194	52.55	94.61
375	59.22	96.75
380	33.90	98.39
383	44.57	94.52
388	49.67	91.37
384	77.05	88.74
381	32.14	95.51
<b>Average accuracy (%)</b>	<b>49.87</b>	<b>94.27</b>

**Table 3** Comparison of result generation time taken on mobile vision and proposed solution

Total words	Mobile vision (time taken for parsing in seconds)	Proposed solution (time taken for parsing in seconds)
194	2.56	17.08
375	4.43	15.25
380	3.22	17.69
383	4.30	16.09
388	3.60	17.12
384	2.00	18.85
381	2.47	18.38
<b>Average Time (s)</b>	<b>3.22</b>	<b>17.20</b>

Both the above-mentioned metrics were tested by reading around 7 text pages from the book “Think and Grow Rich,” and the results are as summarized in Table 2 and 3. Accuracy is measured as per Eq. (1).

$$\text{Accuracy} = (\text{Number of words detected} * 100) / \text{Total words} \quad (1)$$

As can be seen in Table 2, the printed text to digital text conversion accuracy calculated using Eq. (1) is increased by 44.4%, in the case of the proposed solution, but the proposed solution performed slower by 13.98 s during time analysis as can be seen in Table 3. Therefore, it could be safely said that the proposed solution may be slower but it gives more accurate results in a more user-friendly and efficient way.

While collecting feedbacks, many suggestions came forward about how this system can be improved in future and the most common ones were like that include the functionality to be able to store the digital versions of the already read text documents so that they could be used for future references, other suggestions included the ability to comprehend image present inside a book so the person using the application would have a more deeper understanding of the user is reading.

## 5 Conclusion

According to WHO [14] an estimated 253 million people live with vision impairment: 36 million are blind and 217 million have moderate to severe vision impairment; this paper is an effort to put forward a proof of concept of a low-cost system which could in real-time read out any written material in any color, sizes, or text without requiring any Internet connection. The system is so far presented before a blind community in Bangalore, India. The response of the community was quite positive and would prove to be an effective “on the go” reading aid for a blind or VIP.

## References

1. Davis, J. H. (1962). Print recognition apparatus for blind readers. *Journal of the British Institution of Radio Engineers*, 24(2), 103–110
2. Plummer, J. D., Meindl, J. D. (1972). MOS electronics for a portable reading aid for the blind. *IEEE Journal of Solid-State Circuits* 7(2), 111–119.
3. Fourmer d'Albe, E. E. (1914). On a type-reading Optophone, *Proceedings of Royal Society* 90, 373–375.
4. Velázquez, R., Preza, E., & Hernández, H. (2008). Making eBooks accessible to blind Braille readers. In *IEEE International Workshop on Haptic Audio visual Environments and Games* (pp. 25–29).
5. Xiaoli, H., Tao, L., Bing, H., Qiang, C., Qiang, X., & Qiang, H. (2010). Electronic reader for the blind based on MCU. In *International Conference on Electrical and Control Engineering* (pp. 888–890).
6. Velazquez, R., Hernandez, H., & Preza, E. (2010). A portable eBook reader for the blind. In *Annual International Conference of the IEEE Engineering in Medicine and Biology* (pp. 2107–2110).
7. Sabab, S. A., & Ashmafee, M. H. (2016). Blind reader: An intelligent assistant for blind. In *19th International Conference on Computer and Information Technology (ICCIT)*, (pp. 229–234).
8. Braille, L. (1829) *Method of Writing Words, Music, and Plain Songs by Means of Dots, for Use by the Blind and Arranged for Them*.
9. Jethjarurach, N., & Limpiyakorn, Y. (2014). Mobile product barcode reader for Thai blinds. In *2014 International Conference on Information Science and Applications (ICISA)*, (pp. 1–4).
10. Google Play Store (KNFB Reader). Retrieved October 10, 2017, from <https://play.google.com/store/apps/details?id=com.sensotec.knfbreader&hl=en>.
11. KNFB Reader Home Page. Retrieved July 20, 2017, from <http://www.knfbreader.com/>.
12. GNI per capita, Atlas method (current US\$). *World Bank*. 2014-05-01. Retrieved July 16, 2014.
13. Mobile Vision API. Retrieved October 10, 2017, from <https://developers.google.com/vision/text-overview>.
14. Visual impairment and blindness 2017 [Online]. Retrieved August 10, 2017, from <http://www.who.int/mediacentre/factsheets/fs282/en/>.

# Intelligent Task Assignment in a Crowdsourcing Platform



A. Vijayalakshmi and Chittaranjan Hota

## 1 Introduction

Recently, crowdsourcing has emerged as an effective tool for solving extensive problems where humans are effective and computers fail to. The person who posts the task on crowdsourcing platform is called “requester” and the online users who applied for the task are called “Workers.” Then, the task will be assigned to workers and workers will send the solution to the requester through the crowdsourcing platform [1]. Crowdsourcing benefits to tackle issues that any single individual cannot resolve, by aggregating information and work power of heterogeneous users. Crowd wisdom, crowd creation, crowdfunding, and crowd voting are the important classification of crowdsourcing platform. It is applicable to execute a variety of tasks such as evaluations (restaurants, books, Web sites), expert tasks (translations, mathematical problems), voting (teleporting, face book’s “like” button), gather funds (crowdfunding). For example, Amazon Mechanical Turk is an Internet marketplace in which companies and computer programmers outsource simple tasks, and workers are free to choose which ones they want to perform. Workers are paid based on their performance [2] in completing the work.

General idea of crowdsourcing is a requester/crowdsourcer will post a task in a crowdsourcing platform and the participating user will apply/register for that task and then collectively will come with the solution for the posted task. Since we are in digital era, everyone is connected to the Internet which leads to the popularity of crowdsourcing platforms. For many tasks, a crowd might be more efficient and effective than an expert. For example, Galton showed that a crowd could guess the

---

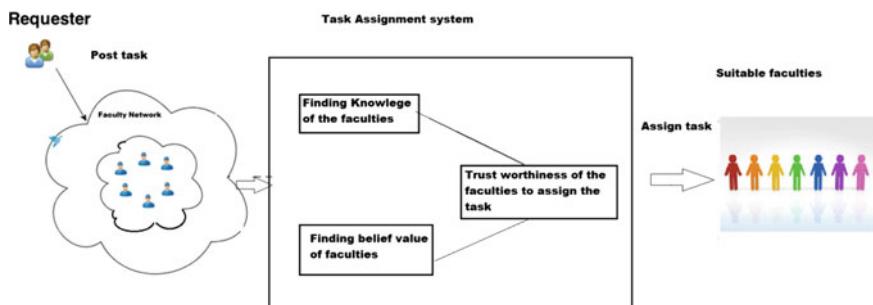
A. Vijayalakshmi (✉)  
BITS – Pilani off Campus Centre, Pune, India  
e-mail: vijayalakshmi\_anand@pilani.bits-pilani.ac.in

C. Hota  
BITS – Pilani, Hyderabad Campus, Hyderabad, Telangana, India  
e-mail: hota@hyderabad.bits-pilani.ac.in

weight of an ox better than a farmer [3]. Other works showed that a group of people performed better than individuals on tasks such as traversing a maze if their decisions were aggregated [4]. Such techniques are effective probably because aggregation cancels out individual errors and reinforces correct solutions [5].

Application of crowdsourcing implies more in business domain, and also we can effectively use it for academics and research application. Due to the extensive growth of Internet technology, in educational institutions teaching, learning, and research are not limited within campus. Crowdsourcing technique is suitable for such multi-campus universities. Since crowdsourcing platform motivates the individuals to coordinate the effort and intelligence to solve major issues, it is not only used for teaching but also helps in performing various tasks in educational institutions. Use of social networks as a crowdsourcing platform is a recent trend in educational institutions. Benefits of social media network as a crowdsourcing platform are to bring online people together, broadcast the information, participate in ongoing conversation, and gather information posted by user. Hence, we have used social media as a crowdsourcing platform. This paper focuses on application of crowdsourcing technique in social media network for faculties so that they can share their ideas cooperatively and can create the course material or lecture material, gather opinions, seek relevant/interesting references/incidents/experiences for a particular topic/subject and also can help in finding the subject matter expertise.

In this paper, we have used trust algorithm for the task assignment. Trust plays a very important factor in the crowdsourcing platform since it works with online users to complete a task. The human might be hoax or fraud and can send wrong solution due to two reasons. First, the impious worker can send answers/solutions randomly to obtain rewards, so the quality of the solution/answer will be degraded. Second, for a complex job, the worker may not have sufficient knowledge for completing the task. As a result, an incorrect answer may be provided [6]. Hence, assigning task based on trustworthiness of the person and skill set is very important to handle in such cases. This paper focuses on implementation of vector space model [7] to find the knowledge value and trust algorithm to assess trustworthiness of the person.



**Fig. 1** Proposed task assignment system

Trustworthiness of a person can be calculated using the belief value and obtained knowledge value. The proposed task assignment system is shown in Fig. 1.

## 2 Related Work

Continual research is happening in crowdsourcing platform specifically for task assignment to users. In some crowdsourcing platform [8], the requester/crowdsourcer gives the required skill set for the particular task and the worker who is having those skills can apply for that task. Once the task is completed, the quality check will be done. Task assignments have been previously studied in the context of mechanism design for crowdsourcing markets. CrowdDB answers queries by micro-task crowdsourcing platform which neither database systems nor search engines can answer effectively [9]. Ho et al. [10] demonstrated in their research work that adaptively allotting workers to assignments can lead more precise output at a lower cost when there are various types of workers.

Task assignment can also be done based on finding expertise or skill set of the particular person. Expert finding method is common in social media application and service-oriented systems [11–14] and vector space model can be used to find out knowledge of the person [7]. Although the person is knowledgeable, we cannot rely on them that they will complete the job successfully. To overcome this issue, some researchers introduced the trust algorithm for task assignment. Trust is a psychological state comprising the intention to accept vulnerability based upon positive expectations of the intentions or behaviour of another [16]. Various algorithms have been implemented to evaluate the trustworthy of people in ecommerce sites, in social media application, and in the area of network security. Page rank algorithms and trust rank algorithms are used in ecommerce sites to find trustworthy customers [17]. Rating method, feedback method is common for some ecommerce sites. The famous Google uses page rank algorithm for searching documents based on the keywords.

There is countable application of crowdsourcing which considers people's trust for a task assignment. A new ranking algorithm named CEF [18] was used to verify the proficiency of users in a particular task by employing social network analysis techniques and WordNet dictionary. Ye and Wang [19] suggested transaction-based trust model to find out dishonest of people. Yang et al. [20] introduced a context-aware trust model, i.e. crowd trust based on the classifications, for calculating the task type-based trust referred to as TaTrust and task reward amount-based trust, i.e. RaTrust. Liu et al. [21] designed a trust-aware task allocation algorithm that takes inputs the estimated trust of workers and pre-set budget and outputs the optimal assignment of tasks to workers and unified graphic trust model is used by Liu [22]. Yin et al. [23] calculated trust between adjacent user using similarity, familiarity, and social reputation measures. Similarly, Adal et al. [24] exhibited a new approach to measure trust, i.e. behavioural metrics, which is calculated through noticeable communication behaviour in social networks. In particular, behavioural trust was measured by the communication between sender and receiver and does not look at

meaning of the messages. As very less amount of work is done on trustworthiness of the users in crowdsourcing environment, also the existing methods rely on the trust of information and information sources to find out the trustworthiness of the user. Motivation for implementing proposed solution on trust algorithm is to help in overcoming challenges of building trust between task producer and requester at crowdsourcing platform. Trust on data sources assumes an indispensable part in numerous areas of cooperation between operators, specifically when data sources are either human specialists or programming specialists.

In stock market, an agent may receive information from a given source about the evolution of a stock's price. In this case, the agent's trust in the source has an influence on the dynamics of the belief about the evolution of the stock's price. The latter belief is fundamental for the agent to decide whether to buy or sell stocks [25]. There are various models of beliefs which have been proposed. One of the belief theories mentioned earlier was by Dempster–Shafer (DS). In this paper, belief is calculated by the interaction between users in the social media. Mostly people assume that belief and trust are interchangeable; however, trust is stronger entity than belief and we can use belief to get trust between the requester and the user. This idea inspired us to use "belief" to find out the trust. Although to get the clearer calculation of trust amongst several crowdsourcers and users, we need "knowledge" as well. This paper proposes a trust algorithm using "belief" and "knowledge" of the user.

### 3 Trust Algorithm for Task Assignment

Our proposed belief- and knowledge-based trust algorithm is to find out the trustworthiness of the person who is trying to apply for the task on the crowdsourcing platform. The job requester allocates the job to the person based on the trust value.

Let  $U = (u_1, u_2, \dots, u_m)$  are the set of users. The task requester is denoted as  $R = (r_1, r_2, \dots, r_s)$ , and the set of task is denoted as  $T = (t_1, t_2, \dots, t_n)$ . There are different types of crowdsourcing task which will be available in crowdsourcing system. In our model, two attributes are considered to find out trust value: belief  $B(u_i)$  of the user and user's knowledge for a task  $K$  ( $u_i \rightarrow t_i$ ). For clarification, the symbol  $Tr(u \rightarrow t)$  is used to represent the trust value from user  $u$  to task  $t$ .

#### 3.1 Calculation of Belief

##### 3.1.1 Familiarity of the User

It is calculated based on number of times the person visited the crowdsourcing site and number of times the person communicated with the task producer

$$\text{Familiarity } \mathbf{F} = \mu_1 * \mathbf{V} + \mu_2 * \mathbf{C} \quad (1)$$

where

- V Number of times visited, and
- C Number of times communicated with user

$$0 << \mu_1, \mu_2 << 1, \mu_1 + \mu_2 \leq 1$$

### 3.1.2 Reputation of the User

It is calculated based on the number of upvotes given to user's comments and number of followers for that particular person.

$$\text{Reputation } \mathbf{R} = x * \mathbf{NC} + y * \mathbf{NF} \quad (2)$$

where NC is the number of upvotes for comments, and  
NF is the number of followers.

$$0 << x, y << 1, x + y \leq 1$$

Here,  $\mu_1$ ,  $\mu_2$ ,  $x$ , and  $y$  are relative weights.

Finally, belief is calculated using familiarity and reputation of the user who has applied for the job.

$$\mathbf{B}(\mathbf{u}_i) = \alpha * \mathbf{F} + \beta * \mathbf{R} \quad (3)$$

where F is the familiarity,  
R is the reputation, and  
 $\alpha$  and  $\beta$  are the relative weights.

## 3.2 Calculation of Knowledge

### 3.2.1 Knowledge of the User

The knowledge of the user is calculated by using vector space model [7]. Vector space model is the heart of information retrieval systems. In which, the messages and queries are considered as vectors. To retrieve the knowledge value in the proposed system, faculty profiles and tasks are considered as vectors for vector space model [26]. We have used the same method which was used in [7] to find out the knowledge of the faculties. We have calculated IDF, i.e. inverse document frequency, and TF,

i.e. term frequency, values for each faculty, and for the given task, cosine similarity measure was used then to sort out the faculties in rank-wise.

Following are the steps used to find out the knowledge of the user:

1. Calculation of inverse document frequency IDF. IDF provides the value, how frequent the current term appears across the entire faculty profiles, using the following formula:

$$\text{IDF}(t) = \log(N/n_t) \quad (4)$$

where N is the total number of faculties profiles we have, and

$n_t$  is the number of faculties profiles containing the word t.

2. Calculation of term frequency TF. Term frequency is related to how often the specific term appears within a document and is calculated so to be independent of the length of the document.

$$\text{TF}(t) = (n_t/N) \quad (5)$$

where  $n_t$  is the number of times term t appears in a faculty profile, and

N is the total number of terms in the document.

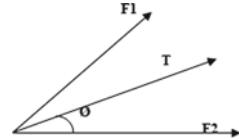
3. Calculation of  $\text{idf} * \text{tf}$  for each user.
4. Calculation of  $\text{tf} * \text{idf}$  values for the given task.
5. Calculation of the length of each user's profile and given task. The term will appear in length document more compared to small documents. So, we need to normalize the document length.
6. Calculation of the cosine similarity for each user. The cosine similarity between task and applied faculties is a measure that calculates the cosine of the angle between them. This metric is a measurement of orientation and not magnitude; it can be seen as a comparison between task and the facilities on a normalized space because we are not taking into the consideration only the magnitude, but the angle between them [27]. Formula to find cosine similarity is:

$$\begin{aligned} \cos(\theta) &= \frac{A * B}{||A|| ||B||} \\ &= \sum_{i=1}^n A_i * B_i / \sqrt{\sum_{i=1}^n A_i^2 * B_i^2} \end{aligned} \quad (6)$$

where  $A_i$  and  $B_i$  are components of vectors A and B, i.e. faculty profile vector and task vector. Figure 2 shows the approach of finding the similarity between faculties and task.

7. Arrange users in descending order as per the knowledge.

**Fig. 2** Illustration of cosine similarity between faculties and task



The final trust formula is as follows:

$$\boxed{\begin{aligned} \text{Tr } (u_i \rightarrow t_i) &= w_1 * B(u_i) + w_2 * K(u_i \rightarrow t_i) \\ 0 << w_1, w_2 << 1, w_1 + w_2 &\leq 1 \\ w_1 \& w_2 \text{ are relative weights} \end{aligned}} \quad (7)$$

To obtain the relative weight values of Eqs. 1, 2, and 7, six-valued Lukasiewicz logic is employed where the  $i$ th grade of importance Imp is given the weight [28]:

$$\begin{aligned} \text{Imp } i &= i/N - 1 \\ &= i/5 \end{aligned}$$

Set imp = {Disregarded/0.0, Unimportant/0.2, Slightly Unimportant/0.4, slightly Important/0.6 Important/0.8, Crucial/1.0}.

## 4 Experimental Result and Analysis

The social media application i.e., faculty network was created to test the performance of our algorithm. Faculty network contains registration details, articles, forum, and chat module. Around 600 faculties have registered in faculty network. To evaluate the trust algorithm, we have taken around 10 faculties for each task. Here, we are selecting only one faculty who is suitable to perform the task. The faculties can post the question and respond to the existing post. They can also write articles about any topics. The faculty network also contains options like send friend request, follow any person and rate the answers/articles posted by other faculty. We have used the open-source tool to extract the keywords from faculties' registration details, articles, chat, and forum. The keywords are put in the faculties profile database which is used as the document in vector space model. The faculties profile database will look like shown in Table 1.

The task is also divided into number of vectors. Then, we have calculated Idf, Tf values for faculties as well as given task to calculate their knowledge value on a particular task. The obtained knowledge is used for the calculation of trustworthiness of the user along with belief. The belief value contains reputation and familiarity of

**Table 1** Faculties' profile database

Faculties ID	Text
Faculty 1	Database (3), Database design (2).....
Faculty 2	Cloud computing (1), private cloud (3)....
Faculty 3	Operating system (2), system calls (2)...
Faculty 4	Computer networks (1), routing (2) ...

**Table 2** Task1—Cloud computing

Task name	Cloud computing		
Task requester	Faculty A		
Faculty ID	Knowledge	Belief	Trust
Faculty 1	1	0.15	0.83
Faculty 2	0.37	0	0.37
Faculty 3	0.3	0.06	0.25
Faculty 4	0.16	Less than threshold	NA
Faculty 5	0.16	Less than threshold	NA
Faculty 6	0.16	Less than threshold	NA
Faculty 7	0.16	Less than threshold	NA
Faculty 8	0.06	Less than threshold	NA
Faculty 9	0	Less than threshold	NA

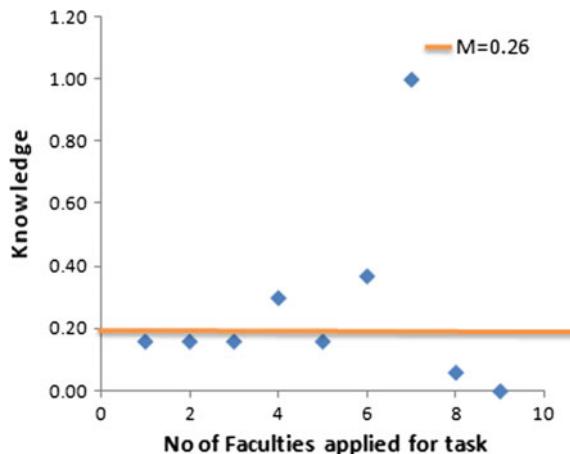
the faculty. Reputation and familiarity will be calculated from the social media. The formula is mentioned in Sect. 3.

To assess the performance of our proposed trust algorithm framework, we conducted experiments on two tasks, i.e. operating system-related task and cloud computing-related task. Task will be posted by faculties on faculty network. For example, cloud computing-related task is posted by Faculty A which is mentioned in task name/task requester rows of Table 2. Faculties who applied for the task are mentioned in Faculties ID column in the table. Knowledge column represents the knowledge value of each faculty which is calculated using vector space model and belief value is calculated only for the faculties who have knowledge value more than the threshold value. The last column represents the trust value which is calculated using formula 7. The trust value calculated using belief and knowledge of faculties for cloud computing is shown in Table 2.

Table 3 shows the trust value calculation of another task, operating system. Around 10 faculties applied for this task and knowledge value is calculated for 10 faculties. Only six faculties have sufficient data related to operating system task, so belief value is calculated only for the faculties who have knowledge value more than threshold value. Here, only three faculties hold knowledge more than the threshold value. Finally, trust value is calculated only for those three faculties.

**Table 3** Task2—operating system

Task name	Operating system		
Task requester	Faculty A		
Faculty ID	Knowledge	Belief	Trust
Faculty 1	1	0.04	0.81
Faculty 2	0.5	0.19	0.44
Faculty 3	0.9	0.1	0.74
Faculty 4	0.4	Less than threshold	NA
Faculty 5	0.3	Less than threshold	NA
Faculty 6	0.1	Less than threshold	NA
Faculty 7	0	Less than threshold	NA

**Fig. 3** Task1—cloud computing

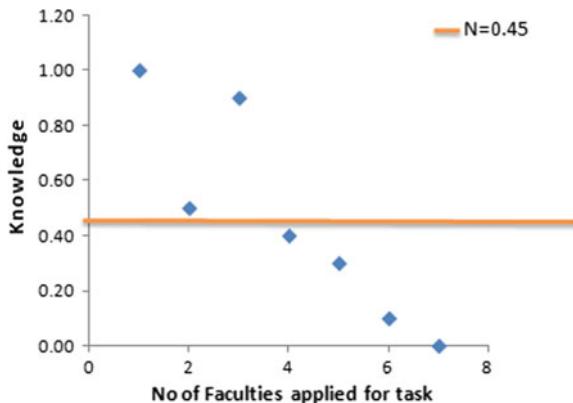
We have plotted the knowledge value of the faculties for the two tasks which is shown in Figs. 3 and 4.

The knowledge values are not distributed evenly, so we have taken the threshold value  $M$  is 0.26 for cloud computing task and  $N$  is 0.45 for operating system task, which is the average of all users' who had been applied for the tasks.

Here, Figs. 3, 4 represent the knowledge value of the faculties applied for the Task 1, operating system and Task 2, cloud computing. We have selected faculties whose knowledge value is more than the threshold value.

Then, we calculated the belief value for three faculties whose knowledge value is above the threshold value. Finally, we got the trust value for three faculties. We have assigned the task to the faculty whose trust value is highest. If we want more faculties, we have to keep threshold value for trust and select faculties above threshold.

**Fig. 4** Task2—operating system



## 5 Conclusions

Discovering reliability of the individual is in all fields; however, here we have ascertained the reliability of the individual to relegate the assignment. The proposed trustworthiness algorithm considers both knowledge and belief value of the person to find out the trust. From the experimental result, it has been proved that by using proposed new system, the person who is knowledgeable and trustable will unfailingly make the quality result for assigned task. This algorithm is most suitable to the situation where the task requester does not know the correct output. Our experimental results also show that the crowdsourcing techniques can be used to solve the issues in universities as well. When applying this technique to expert finding in our research, we obtain a significant improvement on performances.

Here, we have assigned the task to the person who holds highest trust value. In future, we will implement the algorithm to select multiple workers with more data and also trust-based algorithm will be compared with probabilistic approach which is mostly used algorithm for task assignment in crowdsourcing platform.

**Acknowledgements** We would like to thank the off-campus faculty colleagues of our university who have volunteered their data in the social media application that we used to experiment the algorithms proposed in this research.

## References

- Allahbakhsh, Mohammad, Benatallah, Boualem, & Ignjatovic, Aleksandar. (2013). Quality control in crowdsourcing systems: Issues and directions. *IEEE Internet Computing*, 17(2), 76–81.
- Tokarchuk, O., Cuel, R., & Zamarian, M. (2012). Analyzing crowd labor and designing incentives for humans in the loop. *IEEE Internet Computing*, 5.
- Galton, F. (1907). Vox populi (The wisdom of crowds). *Nature*, 75(7), 450–451.

4. Gurnee, Herbert. (1937). Maze learning in the collective situation. *The Journal of Psychology*, 3(2), 437–443.
5. Estes, W. K., & Maddoxf, W. T. (2005). Risks of drawing inferences about cognitive Processes from model fits to individual versus average performance. *Psychonomic Bulletin & Review*, 12(3), 403–408.
6. Liu, X., Lu, M., Ooi, B. C., Shen, Y., Wu, S., & Zhang, M. (2012). CDAS: A crowdsourcing data analytics system. *Proceedings of the VLDB Endowment*, 5(10), 1040–1051.
7. Vijayalakshmi, A., Chittaranjan, H. (2017). Task assignment in Crowdsourcing using vector space model. *International Journal of Pure and Applied Mathematics* 116(23), 605–610.
8. Ipeirotis, P. G. (2010). Analyzing the amazon mechanical turk marketplace. *XRDS: Crossroads, The ACM Magazine for Students*, 17(2), 16–21.
9. Franklin, M. J., Kossmann, D., Kraska, T., Ramesh, S., & Xin, R. (2011). CrowdDB: answering queries with crowdsourcing. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data* (pp. 61–72). ACM.
10. Ho, C. J., Jabbari, S., & Vaughan, J. W. (2013). Adaptive task assignment for crowdsourced classification. In *International Conference on Machine Learning (ICML'13)*.
11. Bozzon, A., Brambilla, M., Ceri, S., Silvestri, M., & Vesci, G. (2013). Choosing the right crowd: expert finding in social networks. In *Proceedings of the 16th International Conference on Extending Database Technology* (pp. 637–648). ACM.
12. Kardan, A., Omidvar, A., & Behzadi, M. (2012). Context based expert finding in online communities using social network analysis. *International J of Computer Science Research and Application*, 2(1), 79–88.
13. Schall, D., Skopik, F., & Dustdar, S. (2012). Expert discovery and interactions in mixed service-oriented systems. *IEEE Transactions on services computing*, 5(2), 233–245.
14. Li, X., Ma, J., Yang, Y., & Wang, D. (2013). A Service Mode of Expert Finding in Social Network. In *2013 International Conference on Service Sciences (ICSS)*, (pp. 220–223). IEEE.
15. Welinder, P., & Perona, P. (2010). Online crowdsourcing: rating annotators and obtaining cost-effective labels. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (pp. 25–32). IEEE.
16. Seco, V. M. M., & Lopes, M. P. (2014). Between compassionateness and assertiveness: A trust matrix for leaders. *Journal of Industrial Engineering and Management*, 7(3), 622–644.
17. Chandratre, P., & Kulkarni, U. (2015). Implementation of trust rank algorithm on web pages. *International Journal of Science Technology &Management*, 04(1).
18. Kardan, A., Omidvar, A., & Behzadi, M. (2012). Context based expert finding in online communities using social network analysis. *International Journal of Computer Science Research and Application*, 2(1), 79–88.
19. Ye, B., & Wang, Y. (2016). Crowdrec: Trust-aware worker recommendation in crowdsourcing environments. In *2016 IEEE International Conference on Web Services (ICWS)*, (pp. 1–8). IEEE.
20. Ye, B., Wang, Y., & Liu, L. (2015). Crowd trust: A context-aware trust model for worker selection in crowdsourcing environments. In *2015 IEEE International Conference on Web Services (ICWS)* (pp. 121–128). IEEE.
21. Liu, X., He, H., & Baras, J. S. (2015). Trust-aware optimal crowdsourcing with budget constraint. In *2015 IEEE International Conference on Communications (ICC)*, (pp. 1176–1181). IEEE.
22. Liu, X., & Baras, J. S. (2015). Trust-aware crowdsourcing with domain knowledge. In *2015 IEEE 54th Annual Conference on Decision and Control (CDC)*, (pp. 2913–2918). IEEE.
23. Yin, G., Jiang, F., Cheng, S., Li, X., & He, X. (2012). Autrust: A practical trust measurement for adjacent users in social networks. In *2012 Second International Conference on Cloud and Green Computing (CGC)*, (pp. 360–367). IEEE.
24. Adali, S., Escrivá, R., Goldberg, M. K., Hayvanovych, M., Magdon-Ismail, M., Szymanski, B. K., & Williams, G. (2010). Measuring behavioral trust in social networks. In *2010 IEEE International Conference on Intelligence and Security Informatics (ISI)*, (pp. 150–152). IEEE.

25. Lorini, E., Jiang, G., & Perrussel, L. (2014). Trust-based belief change. In *European Conference on Artificial Intelligence-ECAI 2014* (p. 549).
26. Bansal, S. (2012). Comparison between the probabilistic and vector space model for spam filtering. *International Journal of Computational Intelligence Techniques*, 3(2).
27. Perone, C. S. (2016). Machine learning: Cosine similarity for vector space models (Part III) (September 12, 2013). <http://blog.christianperone.com/2013/09/machine-learning-cosine-similarity-for-vector-space-models-part-iii/>.
28. Alavi, M., & Leidner, D. E. (2001). Knowledge management and knowledge management systems: Conceptual foundations and research issues. *MIS Quarterly*, 107–136.

# Teaching Algorithms Using an Android Application



Dipika Jain and Pawan Kumar

## 1 Introduction

Education is a process of giving and receiving systematic information or instruction, especially at an educational institute like school, college or university. Prof. Dr. John Dewey related education as a process of experience which is not restricted to age. A few years ago, when the term education was introduced, it was said as imparting face to face instruction. This was termed as passive learning where all the efforts inclusive of the learning activities, imparting knowledge, etc., were made by the teacher. Thereafter came up an active learning system where students used to listen rather than just hearing. Rote learning came into practice, and best crammers were rewarded by our education system. Skills, creativity, innovation, problem-solving, risk-taking were not considered. But now both teachers and students find new ways of teaching and learning, respectively. Various pedagogical methods like seminars, presentations were used as tools for teaching at different educational institutions. There are seminars, guest speakers, workshops, etc., assignment which are evaluated in academic task of both students and teachers. Now with the introduction of new technology, there came smartphones using windows platform, Android platform and iOS.

Android is the most likely mobile operating system which was developed by Google in 2005 but was used commercially in 2008. Students, youngsters and adults all are so familiar and used to this latest technology and Internet that learning took its way towards mobile learning which is portable and can be accessed anywhere. Different Android applications were developed for various purposes. Some of them

---

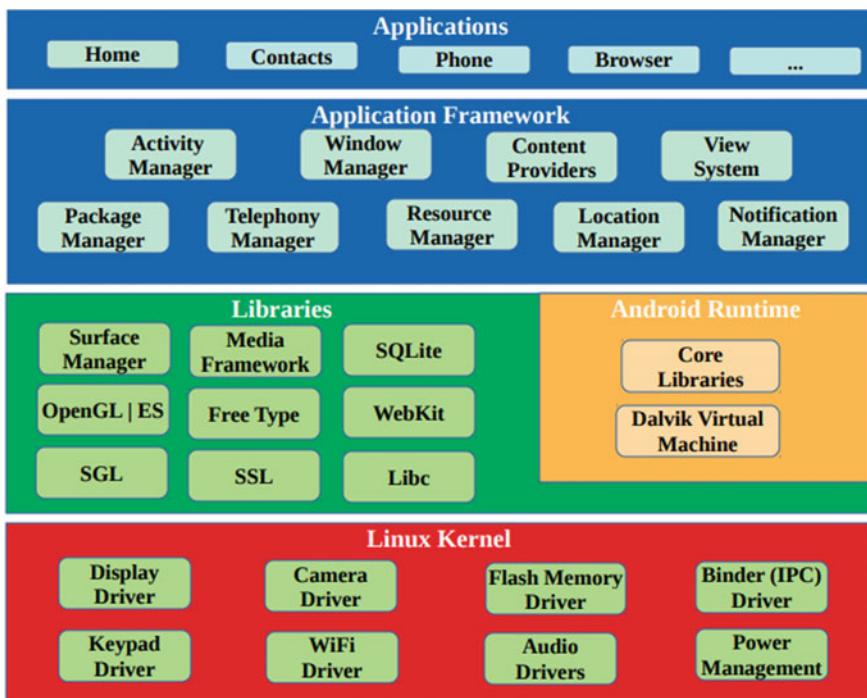
D. Jain (✉) · P. Kumar  
Bharati Vidyapeeth Institute of Management and Research, Paschim Vihar, New Delhi, India  
e-mail: dipikajain12.24@gmail.com

P. Kumar  
e-mail: pparashar657@gmail.com

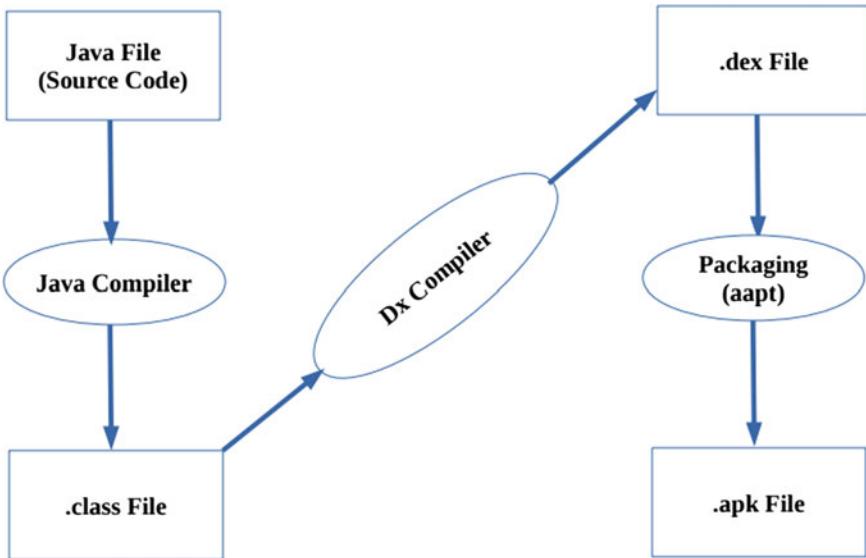
came up to serve in the category of education. These educational applications were related to a particular topic or subject or for general purpose. Apart from it, Android applications for books, dictionary, etc., which serve as tools for education were also introduced. To enhance education, our educators designed various applications for different courses at universities. Some of the applications have been discussed in the later section of the paper.

In this paper, an Android application, SearSort, has been developed which has algorithms of both sorting and searching techniques which are an important part of computer applications and engineering course. Every year, around 160 students intake is available in the computer engineering while 120 for computer application course. There are more than 200 colleges and universities across the globe which offers courses in computer applications and engineering. Before moving to the evaluation results of the application developed, here we present the basic architecture of Android which would give readers an idea about Android OS (Fig. 1).

Figure describes that Android is mainly a four-layer architecture which has five sections. The kernel is responsible for the level of abstraction between the hardware device and the hardware drivers like camera, Wi-fi, keypad, audio, power, display and flash memory as Linux is good for networking. Dalvik virtual machine which is a kind of Java virtual machine, especially designed and optimized for Android, is a key



**Fig. 1** Architecture of android



**Fig. 2** Representation the development of .apk file

component of Android run-time. It uses core features of Linux like multithreading and memory management (Fig. 2).

This application covers an important part of undergraduate course data structures and algorithms. Algorithms can be explained as a procedure of solving problems using specific actions. It is basically a sequence of steps executed in some given time and space. There are various algorithms learnt during the course of data structures and design and analysis of algorithms. Searching and sorting algorithms cover a basic section in the course. This course is taught not only to computer science engineering students but also to the students of other engineering streams.

## 2 Experimental Studies

This section covers a summary of different case studies of various authors under literature survey and details of our application under experimental results.

## 2.1 Literature Survey

The case studies of various authors over the decade have been recorded in the context below. It covers some Android applications used at various universities and opinion of various authors.

Boticki et al. [1] developed a mobile application named Sortko to teach five important sorting algorithms. This application enables a student to choose an algorithm. It does not provide students with the code part of the algorithm. Sortko has been used to teach a course on algorithms and data structures to undergraduate students at the University of Zagreb. This application is renamed as Sortko Sorts in Google Play store. The review on play store records the malfunctioning of quick sort algorithm.

An Android application in the field of medicine for 257 students and 131 junior doctors at school of nursing in University of Nottingham, UK, was developed by Payne et al. [2]. Two hundred and three out of 257 and 98 out of 131 owned smartphones while 115 students and 67 doctors among them owned iPhone. It was a paid application. The application cover features like student timetable, lecture notes, log tables, hospital application and hospital disease guidelines, doctor's leave record, contact on call details for the use of students and doctors, respectively. It was recorded that over the period of 24 h, application was used for around 30 min and 20 min at most by students and doctors, respectively. The negative response of 26 students and 15 doctors was observed due to the cost of smartphones and the clinical application.

Godwin-Jones [3] talked about mobiles applications in Windows, Android and iPhone which are useful in language learning. The paper presents a collective study on all the language learning applications like Talk to Me, Town Musician of Bremen. It also presents basic application development where for Android platform; Java is running on a version of Linux and using objective C for Apple iPhone. The author has reviewed various applications both Web and mobile from 2007 and has summarized various audio, video, game, vocabulary applications for different languages like Pleco, an application for learning Chinese and for a complete language course, and an application living language for French is also considered. It also covers the Web applications combined with mobile devices in form of applications which can be used anywhere by the language learners.

Shah et al. [4] presents an application SmartGlass. This application is based on visual search. Visual search is a technique of searching which involves images instead of text and is based on database technology, crawlers and OCR. The author had reviewed Google Goggles and CamFind for this application. CamFind's accuracy level is 85% and can locate at any angle, while Google Goggles' accuracy is between 15 and 20%. It was considered as a conversion of text to pdf format. The paper concluded that SmartGlass application is a unique blend of visual search and e-commerce which also have add-on features of location-based navigation, voice-based search and sudoku solver.

Kidi et al. [5] developed an educational game Merah Putih for Android platform. This application was designed to educate and provide information regarding Indonesian culture. The game consists of various quizzes related to geography, map world puzzle, etc., which helped players to increase their interest in education. Waterfall life cycle model was used to develop the application with respect to software development. The application was distributed to 100 respondents out of which 66% were male while 34% were women. Forty-seven per cent of them were students who played this application. Some of them used the application on daily basis for 1–3 h. According to the author, most of the players considered gameplay as the critical part of the application. It comprised of four games. Among which geo-challenge for kids is for less than 10 years players, while the other three games were for players of all ages. Similarly, minigame paradise is a multiplayer game, while the rest are single player games.

Jain et al. [6] developed a predictive parsing visualization tool (PPVT) which was used by the engineering students for learning compiler construction course which was taught in the fall semester. The tool helps students calculate first and follow for the grammar in predictive parsing. It was developed in C++ and printed leftmost derivation as the result. Authors deployed it over 62 students out of which 83% of the students recorded positive responses towards the performance and usage of PPVT.

## 2.2 *Experimental Results*

This part of our paper covers the figures and graphical representation related to our work, results and reviews obtained so far. This application is developed on Android, the architecture of which has been briefed in the earlier section of the paper. Android Studio is software and for which an Android phone is used as an emulator. It is available on Google Play for all Android versions in market. Our Android application is developed using four layers in the package version 1.0. Some of the quick screenshots of the application are here. The figures represent the icon and the splash banner (Fig. 3).

**Fig. 3** Icon of application—SearSort



Working of app SearSort is presented via Fig. 4 through 10 where sorting and searching features are shown. This application has a main menu having two sections—sorting and searching which have number of algorithms. College logo is the starter of the application which lasts for a few seconds. This application was shared with the undergraduate computer application students at Bharatiya Vidyapeeth Institute of Management and Research for their courses data structure algorithms. The survey of our application also targeted high school students who had opted for computer science and information practices as their main subject (Figs. 5, 6, 7, 8, 9 and 10).



**Fig. 4** Banner of application—SearSort



**Fig. 5** Screen 1

### Selection Sort

```
void sort(int arr[])
{
    int n = arr.length;

    // One by one move boundary of unsorted
    subarray
    for (int i = 0; i < n-1; i++)
    {
        // Find the minimum element in unsorted
        array
        int min_idx = i;
        for (int j = i+1; j < n; j++)
            if (arr[j] < arr[min_idx])
                min_idx = j;

        // Swap the found minimum element with
        the first
        // element
        int temp = arr[min_idx];
        arr[min_idx] = arr[i];
        arr[i] = temp;
    }
}
```

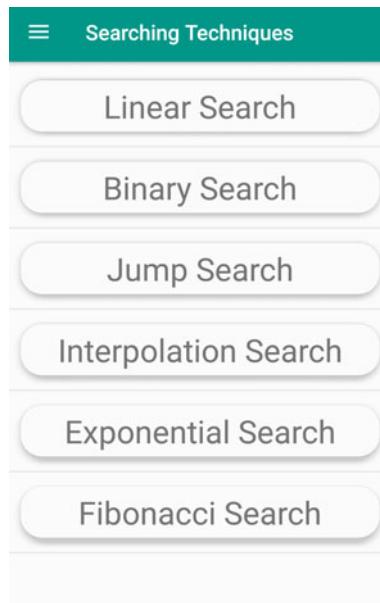
 Source Code       Try Code

**Fig. 6** Screen 2

Google Play recorded several reviews and rating of the application. The result of the issues in form of feedback is discussed in the next section of the paper.

### 3 Review and Suggestion

This section of the paper consists of the facts observed from cases above in the literature survey and opinion from our understanding. Apart from it, the section also carries the reviews observed by the app users based on a set questionnaire.



**Fig. 7** Screen 3

The questionnaire carried seven questions in total against which 43 responses were recorded. The earlier analysis depicted that out of 43 students, 76.7% of were undergraduate students pursuing bachelor of computer application program, 7% were students of computer applications, while rest were the students of high school. Statistics verifies 33 engineering students used the application. Now on considering the present analysis observed during the boards 2018, it was noted that number of high school users has increased by 27% of the initial users. The survey is recorded based upon certain factors, namely presentation, performance and usage of the application.

- Presentation—Design of the application

According to the feedback of the users, it is observed that the presentation of the application was not liked by a few, while 65.1% of the users liked it. A neutral response from 23.3% was also recorded.

- Performance of the application

Out of the total responses, positive views were given by 72.1% of the students, while 25.9% gave a neutral answer for the same. 76.6% of the students agreed to the fact that the application is useful for the undergraduate students.

**Fig. 8** Screen 4

The screenshot shows a mobile application titled "Linear Search". The main content area displays the following Java code:

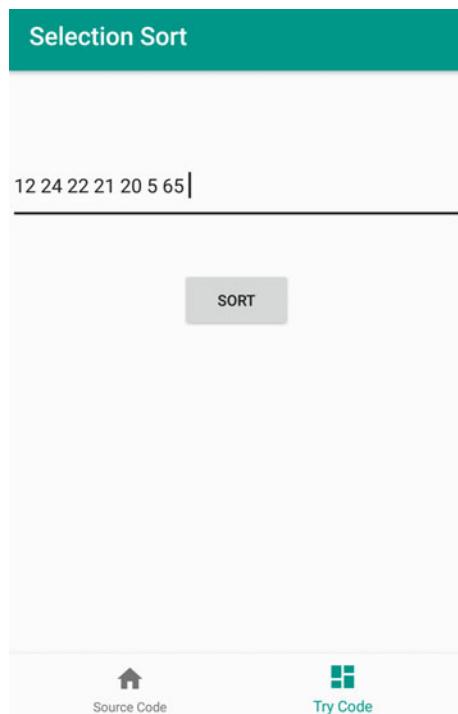
```
static int search(int arr[], int n, int x)
{
    for (int i = 0; i < n; i++)
    {
        // Return the index of the element if the
        // element
        // is found
        if (arr[i] == x)
            return i;
    }
    // return -1 if the element is not found
    return -1;
}
```

At the bottom of the screen, there are two buttons: "Source Code" with a house icon and "Try Code" with a square icon.

- Usage of the application

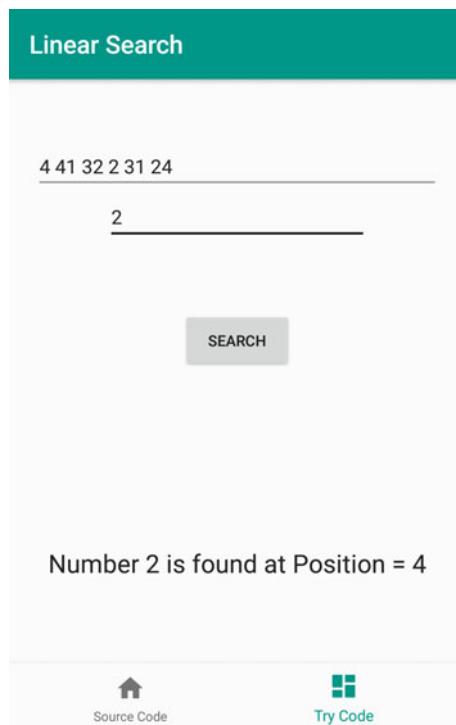
The question for usage was framed so as to categorize users into three groups mainly as the users who use the application daily, the users using the application once a week and the users using the application once in a month. Responses were recorded, and it is found that maximum users use the application once in a week, while 18.6% of the total uses the application daily.

Considering all the aspects of the application, it is observed that most of the users were undergraduates and approved it to be a useful application, while a few were high school students who also willingly used the application.

**Fig. 9** Screen 5

## 4 Conclusion

The Android applications were developed with the motive of introducing a new way of learning among students. It is also observed that responses for paid applications were not in favour of the developers by the users who cannot afford. Therefore, educational applications which are developed, especially for some purpose to serve the institution and students, must be a free version. The application described in this paper is available free on Google Play, and its contents cover a unit of two courses of computer science, namely data structures and design and analysis of algorithms. A positive feedback is observed from maximum users from high school and under-graduate courses. Hence, it can be used as a teaching and learning application for searching and sorting by faculty and students, respectively.

**Fig. 10** Screen 6

## References

1. Boticki, I., Barusic, A., Martin, S., & Drljevic, N. (2013). Teaching and learning computer science sorting algorithms with mobile devices: A case study. *Computer Applications in Engineering Education*, 21(S1), E41–E50.
2. Payne, K. F. B., Wharrad, H., & Watts, K. (2012). Smartphone and medical related App use among medical students and junior doctors in the United Kingdom (UK): A regional survey. *BMC Medical Informatics and Decision Making*, 12, 121.
3. Godwin-Jones, R. (2011). Emerging technologies-mobile apps for language learning. *Language Learning and Technology*, 15(2), 2–11.
4. Shah, J., Desai, T., & Shah, P. (2015) SmartGlass: Visual commerce application (android). In *International Conference on Advanced Computing Technologies and Applications.; Procedia Computer Science*, 45, 236–243.
5. Kidi, N., Kanigoro, B., Salman, A.G., Prasetio, Y.L., lokaadinugroho, I., & Sukhmandani, A. A. (2017) Android based Indonesian information culture education game. In *2nd International Conference on Computer Science and Computational Intelligence; Procedia Computer Science*, 116, 99–106.
6. Jain, A., Goyal, A., & Chakraborty, P. (2017). PPVT: A tool to visualize predictive parsing. *ACM Inroads*, 8(1), 43–47.

# Keyword Extraction Using Graph Centrality and WordNet



Chhavi Sharma, Minni Jain and Ayush Aggarwal

## 1 Introduction

Keywords are a short representation of a document which summarizes the important content of the document into certain selective words which are either picked from the document or constructed using the underlying sense of the document. Keyword extraction deals with automatically picking up the underlying text from the document which can best describe the senses of it.

Keyword extraction is a very important field of study for NLP purposes because it serves great many functionality. The amount of text being produced daily is increasing exponentially and it becomes impossible for the reader to sift through it, and hence, keyword presents an alternative of understanding for the user to determine whether the document deserves the time or not. Keywords are also used in information retrieval systems [1] such as search engines to better index the documents and provide document management and categorization.

A lot of research has been done in the field of keyword extraction from supervised to graph-based approach, but not much work has been done exploring the semantic relatedness of the term and ranking them on the language tree to determine the central words of the documents. In this paper, we present an unsupervised learning approach to extract keywords from the document which are based on the semantic strength of words in a particular document. We use WordNet [2] as our knowledge base to

---

C. Sharma · M. Jain (✉) · A. Aggarwal  
Delhi Technological University, Shahbad Daulatpur, Bawana Road,  
New Delhi 110042, Delhi, India  
e-mail: minnijain91@gmail.com; minnijain@dtu.ac.in

C. Sharma  
e-mail: sharma.chhavi96@gmail.com

A. Aggarwal  
e-mail: aggarwal96ayush@gmail.com

construct our graph and various centrality measures to determine the most important words which are thus termed as keywords.

## 2 Related Work

Keyword extraction has been the topic for extensive research since a long time with advancement being made to improve it using a lot of natural language processing tools. Both supervised and unsupervised approaches have been discovered and used over the years for keyword extraction.

Initially, keyword extraction was done only for “global context information”. “Local context information” was determined by using techniques like support vector machines and used to classify the documents [3].

Automatic keyword extraction for documents by generating lists of stop words for specific corpora and domains [4] had improved the benchmark for keyword extraction. They used news articles and defined metrics for characterizing the exclusivity, essentiality, and generality of extracted keywords within a corpus. WordNet would serve a similar purpose in eliminating the stop words and focusing on syntactically related words.

Supervised algorithms such as KEA [5] and GenEx [6] explored the frequency of a word in the document and their location to determine the importance of word to be chosen as a keyword. Their algorithm was further improved in 2006 by Witten and Medelyan [7] by using semantic information for the phrases and the words of the document using a domain-specific knowledge base.

Moreover, various algorithms on lexical databases were time consuming and inefficient for extracting keywords or text summarization. WordNet lexical database eliminated the connection words and built a tree using generic terms like hypernym or lexicographer file. By weighing the various tree levels and using other statistical methods, a restricted number of keywords for the entire document were calculated [8].

While supervised algorithm provided a standard and bench-marked algorithm for the process of keyword extraction, it had a few shortcomings. Apart from needing hand-tagged documents which are not readily available, they are also biased toward the domain in which they are trained on and, hence, are not suitable for extensive usage [9]. Thus, the research shifted on unsupervised approaches. HaCohen-Kerner proposed a uni-gram, 2-gram, and 3-gram-based approach to extract keywords which use abstract and titles and presents the group with maximum value as the keyword [10]. This was further improved by Pudota et al. by using POS tags into the n-grams extracted and thus made it domain independent [11].

The idea of using WordNet is demonstrated by Wei et al. [12] where semantically related words are identified with the help of WordNet lexicon. This is extremely useful in keyword extraction as a paragraph or document that revolves around a central idea contains semantically related words or synonyms for the main phrase or “keywords”.

Graph-based approaches have also been researched extensively for their ease of calculating domain-independent keywords from the documents. SemanticRank [13] uses a graph-based technique exploiting the semantic relatedness of the document using knowledge-based measures of WordNet and Wikipedia and graph centrality measures of PageRank [14] and Hits [15].

Boudin [16] carried out experiments on three standard datasets having different languages and showed that simple graph centrality measures like degree centrality could achieve results comparable to widely used TextRank algorithm. This is especially useful for shorter documents.

Unsupervised graph-based keyword extraction is useful in scientific documents as there is generally no preexisting dataset for the same. In a similar method, Schluter [17] extracted keywords from scientific articles using the document model obtained from text graphs as an input to various centrality measures.

Selectivity-based keyword extraction has been proposed as a new unsupervised graph-based keyword extraction method by Beliga et al. [9], along with various methods such as Croatian keyword extraction and other graph-based approaches.

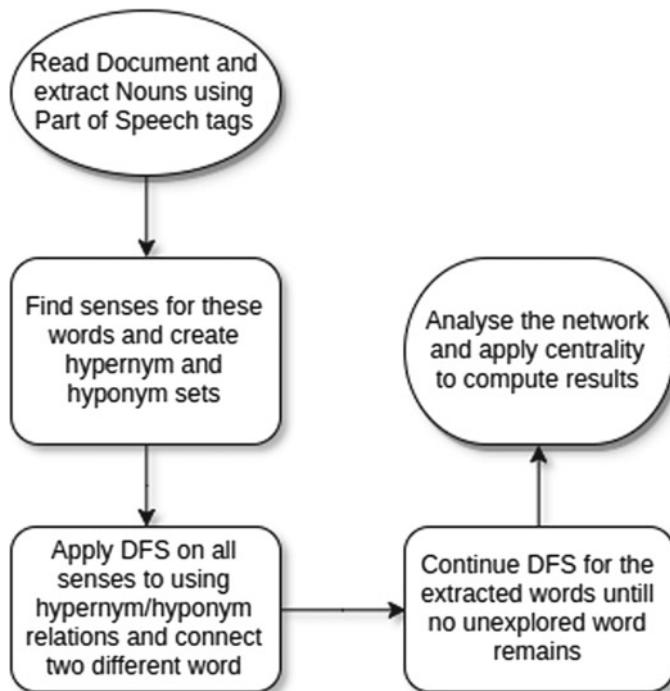
Keywords are more likely to be found among influential nodes of graph of words rather than among its nodes high on eigenvector-related centrality [18]. Using graph degeneracy, this method calculates keywords for documents (short and medium) and produces the best results for longer documents. Document clustering using keywords by representing documents as graphs and their words as nodes has been explored by Nagarajan and Nair. Their algorithm produces more than 90% accuracy.

While the above-mentioned methods hint on the usage of semantic network using graph network, they have not explored the aspect of semantic relatedness for keyword extraction. We derive our hypothesis from an algorithm used in word sense disambiguation [19] and evaluate the graph using various centrality measures and compare results.

This paper is organized as follows. In Sect. 3, we talk about the proposed algorithm along with a discussion on various centrality methods used. It is followed by experimental results done on a dataset of abstract followed by a discussion on future work before concluding the paper.

### 3 Proposed Scheme

Our keyword extraction algorithm is created on a graph connectivity-based measure. In order to isolate the impact of graph connectivity measures on keyword extraction, we use a fairly general disambiguation algorithm that has few parameters and relies almost exclusively on graph structure for inferring word senses [19]. This approach has been extended to other text-based purposes as well [20, 21], but has not been much explored in keyword extraction. We use WordNet sense inventory but neither our graph-based algorithm nor our connectivity measures are limited to this lexicon. Resources with alternative sense distinctions and structure can also serve as a knowledge base for our method.



**Fig. 1** Flowchart of the algorithm

We use the relations of *Hyponymy* and *Hypernymy* of English language to find a relation—a path in the graph, from one word to another. Hyponymy stands for a subordinate relationship that is *part-of* relationship, whereas Hypernymy stands to cover the broad meaning of any term, i.e., *is-a* relationship. The senses or the *synsets* of a word from the text are expanded and analyzed to match with any sense of the synset of any other word from the document, and hence, we create a network structure with connection denoted by edges (Fig. 1).

Our algorithm first parses the document to extract all the words from the document, removing the stop words and finding all nouns. This step is essential to remove all unnecessary words which can never act as a keyword of our document, thus saving on the computation time. We use the aforementioned relationship of Hypernymy and Hyponymy to do a depth-first search of the constructed wordlist to create the graph. The graph is created by choosing a word and then keeps deriving its synsets till it connects with any pre-discovered node of the graph.

To analyze the structure of graph and find out the important nodes, we use various centrality measures. Using the centrality measure, we try to determine the most important node(s) of the system, and on the basis of this, we term that node as the most connected and hence the most important node, thus being chosen as the keyword.

### 3.1 Centrality Measures

Centrality measures are used to define the importance of a node  $x$  in a graph. They are used to denote the influence of a node in the construction of another node, that is, how central that node is for the graph. For our algorithm, we use four centrality measures—*Degree*, *Betweenness*, *PageRank*, and *Closeness (KPP)*.

**Degree Centrality** of a node determines the importance of a node by calculating the number of edges incident on the node. In an un-directed graph, we can think of degree centrality as the sum of all the edges on the graph. For a node  $x$  in a graph  $S$ , degree centrality can be defined as:

$$\text{Degree}(x) = |(x, y) \in E : x, y \in S|$$

**Betweenness** is the ratio of the shortest path from the current vertex to all the other vertices to the sum of shortest path between any two vertices. It determines the importance of a node in a network by determining whether it is “central” to all the other nodes or not.

$$\text{Betweenness}(x) = \sum \frac{\sigma_{i,j}(x)}{\sigma_{i,j}}$$

**Page Rank** is used to identify the node with maximum number of connections with other nodes in the network [14]. It also takes into consideration the degree of node that our current node is connected to and gives importance to nodes with more connections to determine the score.

$$\text{PageRank}(x) = \frac{1 - k}{|S|} + d \sum_{x,y \in E} \frac{\text{PR}(y)}{\text{outdegree}(y)}$$

**Closeness** is used to determine the degree of importance of a node by evaluating its closeness to every other node. It depends on the sum of shortest path distances between the node and all the other nodes it is connected to.

$$\text{Closeness}(x) = \frac{\sum_{y \in S: x \neq y} \frac{1}{d(x,y)}}{|S| - 1}$$

**Hits** finds the importance of a node in a network by analyzing the number of connections incoming and outgoing to the node. It works on the recursive dependency of the hubs and authority nodes [15]. Hubs are the nodes which have many outgoing links whereas authority is the nodes which have various incoming links.

$$\text{HITS}_A(x) = \sum_{y \in \text{In}(x)} \text{HITS}_H(y)$$

$$HITS_H(x) = \sum_{y \in Out(x)} HITS_A(y)$$

### 3.2 Algorithm

- Step 1: Parse the document and extract all the words from the document, and remove stop words<sup>1</sup> and select all the words which have a noun meaning in their *synset*. Also, initialize the list of words as *NULL*
- Step 2: For every word  $x \in S$ , calculate the hypernyms and hyponyms for the sense  $x$  and do a DFS with all the discovered words. If a match is found, a relation is said to exist between the two words.
- Step 3: Explore the edges from most recently discovered word  $y$  till the point there are no new edges to be discovered.
- Step 4: After the words are processed, analyze the network using the following centrality methods.

If  $x, y$  denote the nodes of a graph and  $E$  the set of edges and  $S$  the set of nodes, then

$$Degree(x) = |(x, y) \in E : x, y \in S|$$

$$Betweenness(x) = \sum \frac{\sigma_{i,j}(x)}{\sigma_{i,j}}$$

$$PageRank(x) = \frac{1 - k}{|S|} + d \sum_{x,y \in E} \frac{PR(y)}{outdegree(y)}$$

$$Closeness(x) = \frac{\sum_{y \in S, x \neq y} \frac{1}{d(x,y)}}{|S| - 1}$$

$$HITS_A(x) = \sum_{y \in In(x)} HITS_H(y)$$

$$HITS_H(x) = \sum_{y \in Out(x)} HITS_A(y)$$

and evaluate the top keywords from every centrality to propose result.

---

<sup>1</sup> Stop-words are the words in a language that do not serve much meaning but are used for construction of a sentence and binding it together.

## 4 Experimentation and Results

We run our algorithm on a dataset of journal articles picked up from the Internet a sample of which is shown above. Only the abstracts of the articles are processed as discussed and words with noun POS tags are extracted and fed to the algorithm.

The author suggested keywords which are used for matching and analyzing the result. One such abstract of a sample document is shown below (Fig. 2). The keywords extracted for a sample document are shown in Table 1, and the actual keywords for the sample document as given by the author are shown in Table 2.

Since our extraction algorithm works only on keywords, we use all the words of the expected keyword set to match the extracted keywords to evaluate the result. We compare the extracted keywords and the expected keywords by analyzing the root word and not the specific word forms, i.e., *living* and *live* will give a match.

As abstract provides less data to work upon as compared to full articles, we hypothesize the accuracy of the algorithm thus derived here will be the bare minimum of another such algorithm. The calculated precision score of the algorithm after running on the dataset for various centrality methods are.

Thus as shown in Table 3, we observe that the algorithm performs fairly well despite being run on just the abstract of the articles and centrality measures of *Betweenness* and *Closeness* perform better than any other centrality method.

Increases in overweight and obesity have been observed globally in both developed and developing countries. The authors assessed the relation between lifestyle factors and body mass index (BMI) (weight (kg)/height (m)<sup>2</sup>) in a population-based longitudinal study, using BMI and its subsequent change as responses in a multilevel model. The authors included 11,115 men and women aged 20-61 years at baseline who were living in the municipality of Tromsø, Norway, and who participated in three or four consecutive health surveys between 1979-1980 and 2001. Baseline age, physical activity at work, coffee consumption, and desired BMI (i.e., the BMI that the subjects reported they would like to have) were positively associated with baseline BMI, whereas height, alcohol consumption, leisure-time physical activity, and level of education were inversely associated. Most relations were found to be stronger in women than in men. Clinically relevant effect sizes were observed for most of the significant associations, especially in women. For instance, on an ordinal scale, a one-category increase in educational level would decrease the mean baseline BMI among women by 0.30 kg/m<sup>2</sup>. Significant associations between several lifestyle factors and subsequent BMI change revealed that observed baseline associations were strengthened over time, especially in women.

**Fig. 2** Sample document

**Table 1** Extracted keywords from the document from various centrality measures

Centrality	Keywords
Closeness	Time, mass, model, live, age, response, effect, increase, size, body
Hits	Time, mass, living, age, size, increase, study, body, response, model
PageRank	Relation, association, baseline, level, woman, activity, scale, index, body, increase
Degree	Body, activity, relation, level, time, mass, association, size, live, men
Between	Body, education, men, association, index, study, effect, scale, size, level

**Table 2** Keywords provided by the author

Keywords
Adult, aged, body mass index
Female, humans, lifestyle
Linear models, longitudinal studies, male
Middle aged, norway, obesity
Questionnaires, risk factors, urban population

**Table 3** Precision score for different number of keywords

Centrality measures	Three keywords (%)	Five keywords (%)	Seven keywords (%)
Closeness	22	18	14
Hits	12	15	14
PageRank	10	9	9
Degree	16	12	10
Betweenness	22	18	15

## 5 Observation and Future Work

We were able to deduce the fact that graph centrality provides a simple yet sophisticated way to extract keywords from a document. While the algorithm right now works only on English, it is not restricted to this language and can be extended depending upon the choice of the knowledge base. The limitation of this algorithm lies in the fact that we see a lot of clustering of words arising from similar meaning words which appear repeatedly in a paragraph from the same sense, hence biasing the algorithm toward that. The algorithm also shows a tendency to get swayed toward a paragraph with broader meanings as they tend to have more connections with words down the language tree. We also are unable to extract key phrases due to the limita-

tions of WordNet lexicon, which we want to alleviate using other lexicons such as ConceptNet or Wikipedia.

## References

1. Jones, S., & Paynter, G. (2002). Automatic extraction of document keyphrases for use in digital libraries: Evaluation and applications. *Journal of the American Society for Information Science and Technology*.
2. Fellbaum, C. (1998). *WordNet: An electronic lexical database*. Cambridge, MA: MIT Press.
3. Zhang, K., Hui, X., Tang, J., Li, J., Yu, J. X., Kitsuregawa, M., Leong, H. V. (2006). Keyword extraction using support vector machine advances in web-age information management.
4. Rose, S., & Engel, D., Cramer, N., Cowley, W. (2010). Automatic keyword extraction from individual documents. *Text Mining: Applications and Theory*, 1–20. <https://doi.org/10.1002/9780470689646.ch1>.
5. Witten, I. H., Paynter, G. W., Frank, E., Gutwin, C., & Nevill-Manning, C. G. (1999). KEA: Practical automatic keyphrase extraction. ACM DL.
6. Turney, P. (1999). Learning to extract keyphrases from text. *Information Retrieval*.
7. Witten, I. H. & Medelyan, O. (2006). Thesaurus based automatic keyphrase indexing. In *Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL'06)* (pp. 296–297), Chapel Hill, NC.
8. Cerbulescu, C., & Leotescu, G. S. (2017). Extracting text keywords using WordNet (pp. 1–4). <https://doi.org/10.1145/3136273.3136280>.
9. Beliga, S., Ana, M., & Martinčić-Ipšić, S. (2015). An overview of graph-based keyword extraction methods and approaches. *Journal of Information and Organizational Sciences*, 39(1), 1–20.
10. HaCohen-Kerner, Y. (2003). Automatic extraction of keywords from abstracts.
11. Pudotta, A., Dattolo, A., & Baruzzo, A. (2010). New domain independent keyphrase extraction system digital libraries. In *6th Italian Research Conference, IRCDL*. Padua, Italy.
12. Wei, T., Lu, Y., Chang, H., Zhou, Q., & Bao, X. (2015). A semantic approach for text clustering using WordNet and lexical chains. *Expert Systems with Applications*, 42(4), 2264–2275, ISSN 0957-4174.
13. Tsatsaronis, G., Varlamis, I., & Nørvåg, K. (2010). SemanticRank: ranking keywords and sentences using semantic graphs. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING'10)*.
14. Page, L., Brin, S., Motwani, R., & Winograd, T. (1999). The pagerank citation ranking: bringing order to the web. Technical Report, Stanford InfoLab.
15. Kleinberg, J. M. (1998). Authoritative sources in a hyperlinked environment. In *Proceedings of the Ninth Symposium Discrete Algorithms* (pp. 668–677).
16. Boudin, F. (2013). A comparison of centrality measures for graph-based keyphrase extraction. In *International Joint Conference on Natural Language Processing (IJCNLP)* (pp. 834–838), Nagoya, Japan.
17. Schluter, N. (2014). Centrality measures for non-contextual graph-based unsupervised single document keyword extraction. In *Proceedings of TALN Association for Computational Linguistics*.
18. Tixier, A., Malliaros, F., & Vazirgiannis, M. (2016). A graph degeneracy-based approach to keyword extraction. In *EMNLP*.
19. Navigli, R., & Lapata, M. (2010). An experimental study of graph connectivity for unsupervised word sense disambiguation. *IEEE Transaction on Pattern Analysis and Machine Learning*, 32(4).
20. Jain, A., Mittal, K., & Tayal, D. K. (2014). Automatically incorporating context meaning for query expansion using graph connectivity measures. *Progress in Artificial Intelligence*, 2, 129–139.

21. Jain, A., & Lobiyal, D. K. (2014). A new approach for unsupervised word sense disambiguation in Hindi language using graph connectivity measures. *International Journal Artificial Intelligence Soft Computing* 4(4), 318–334.

# Hybrid Mobile Learning Architecture for Higher Education



**Asrat Mulatu, Addisu Anbessa, Sanjay Misra,  
Adewole Adewumi, Robertas Damaševičius and Ravin Ahuja**

## 1 Introduction

The great advancements in the area of information communication and information management have made a significant transformation in various aspects of the day-to-day activities of modern society. The transformation can obviously be seen in several domains like transactional business, health, educational, agricultural, and many other sectors. The major factor for all these changes is due to the wide utilization of computing devices such as smart phones, tablets, and laptops. These computing machines have got higher capacity in processing and storing of data besides getting cheaper every day.

---

A. Mulatu

Addis Ababa Science and Technology University, Addis Ababa, Ethiopia  
e-mail: ambnn2001@gmail.com

A. Anbessa

St. Mary's University, Addis Ababa, Ethiopia  
e-mail: ambassaa@gmail.com

S. Misra (✉) · A. Adewumi

Covenant University, Ota 1023, Nigeria  
e-mail: sanjay.misra@covenatuniversity.edu.ng

A. Adewumi

e-mail: Wole.adewumi@covenatuniversity.edu.ng

R. Damaševičius

Kaunas University of Technology, Kaunas, Lithuania  
e-mail: robertas.damasevicius@ktu.lt

R. Ahuja

University of Delhi, New Delhi, India  
e-mail: ravinahujadce@gmail.com

Recently, the architecture and interest of developing application software have been highly inclined to the Web-based scenario because of the usability and flexibility it offers. Web applications can be developed using a wide variety of programming languages, platforms, and technologies as they are intended to be accessed by different networked computing devices ranging from mobile phones to desktop computers. The mobile devices are mostly designed to use the wireless media to access networked systems. The infrastructure of wireless networks such as Wi-Fi and cellular is becoming part of the standard in most of public service providing sectors. This has situated the environment for computing while-on-the-move which is the most required flavor in the forthcoming computing technology. Furthermore, to enhance this approach of computing, nowadays, cloud computing (with concept of providing computing as a service) is playing a significant role. Therefore, in this paper, the core focus is on how to utilize these emerging features to enhance teaching and learning experiences. In doing so, in this work, tablets and smart phones are targeted due to their suitability for Android-based systems. Moreover, the mobile apps are intended to incorporate cloud services to enhance learner experiences and to improve teaching–learning services.

Therefore, in this work, the classroom-confined offerings are to be extended via anytime, anywhere, and for anything via a new learning architecture. Here, the new mobile learning application architecture, combining both the native and Web-based client–server, avoids the weaknesses of the two traditional models. The hybrid of the two models is used by integrating further with cloud services. The architectural design of the application has also considered all these issues against the constraints on mobile devices and opportunities available in computation technologies.

By using the hybrid architecture for mobile learning, it is possible to design and develop a notification and data message enhanced mobile learning application to realize the idea of how to timely deliver educational content and material independent of time and space. Similarly, it is possible to get support and to provide support on lessons on anytime anywhere basis. Furthermore, the approach provides an option to virtually extend the learning process to lifelong through informal learning. Generally, this work aims to enhance the learning experience by settling a new architecture for mobile learning application to facilitate educational delivering activities as a complementary option.

The agile software development process model and the iterative prototype testing technique are used besides the various methods and tools employed in this work. The latest Google cloud messaging service, named Firebase, and many supported APIs are used to launch contents from online learning content management system. The Android platform, via the Android studio, and the various APIs thereof are exploited in the process. Bootstrap for the responsive Web pages, PHP for server side scripting, and MySQL database system are used in the process. The developed system is also evaluated by experts and students alike in a networked environment in a laboratory setting.

The rest of the paper organized as follows. The next section presented a brief of review of literature and related works. Section 3 presents the proposed mobile learning architecture while Sect. 4 shows the prototype implementation followed by

Sect. 5 which shows the system validation and evaluation. The last section, Sect. 6, concludes the paper.

## 2 Related Works

In many literatures related to learning applications, mobile learning application is one part of the overall e-learning paradigm. Yet, the e-learning itself is included within the category of distance learning which was intended to make the traditional way of learning flexible [1]. The obvious problem related to the traditional learning method is being space and time dependent as its offerings is confined to a specific classroom and instructor led. So, the idea of mobile learning is rooted from the distance learning and e-learning. Besides, the continuous advancement in mobile computing and wireless technologies has given a sustainable ground for the mobile application, currently and for the upcoming future.

Furthermore, the future direction of ubiquitous computing in line with the current intensive utilization of mobile devices is attracting app developers toward developing many kinds of mobile applications of which mobile learning is one aspect. Either mobile learning or mobile e-learning or M-learning is all about designing and developing applications to provide educational content in anywhere-anytime-anyhow (any media format, formally/informally through mobile devices) mode on mobile devices through wireless network as presented in [1, 2]. Particularly, as stated in [1], the mobile learning definition was conceptualized as “any sort of learning that happens when the learner is not at a fixed, predetermined location, or learning that happens when the learner takes advantage of the learning opportunities offered by mobile technologies”.

Similarly, several authors [1–7] have coined the terms “wireless, ubiquitous, seamless, nomadic, or pervasive learning/education” to mean mobile learning.

From teaching and learning points of view, mobile learning is being facilitated in the twenty-first-century learning environments which includes the principles of student-centered pedagogies, ICT implementation and integration strategies, innovative teaching practices, learning objectives, and teacher’s competencies as shown in [1].

As far as the architecture of the mobile learning application is concerned, based on the literatures regarding mobile learning application, the most suggested architecture is the client–server one as described in [4, 11]. Another category of architecture proposed for mobile learning is the native model. To some extent, there is also a trial of mixing the features of the two architectures as presented in [4, 5]. Native architecture mobile apps are heavy weighted, installable, and hold static content. But these apps are good at utilizing the local device resources like camera, Global Positioning System (GPS) as discussed in [4]. Mobile apps with client server architecture are simple and optimized Web apps for mobile devices. The major drawback with this class of apps is their low utilization of local device resources as compared to the native ones [4].

Therefore, in this study, hybrid architecture is proposed. The hybrid mobile app is used to combine the strength of both client–server and native features. Since the app of this approach is browser independent and Internet enabled, it has got the advantage to be seamlessly integrated with the cloud service in order to make it on time through push notifications.

The proposed architecture of mobile learning application is also intended to utilize the opportunities provided by Google cloud services and the Android operating system. In Google cloud, there is a functionality to let application developers in sending message to a device running Android. This is through Google Cloud Messaging (GCM) which is a service developed by Google for device-to-device communication [8–11]. GCM works in client–server architecture to enable communication between client and server apps. This can be done through upstream message from client to server and downstream message in the reverse direction. Fortunately, Android has got the feature to support GCM and the latest version Firebase Cloud Messaging (FCM) [12–16]. The latter has cross-platform features unlike to GCM.

Android is an open-source platform initiated by Open Handset Alliance (OHA) in which Google is playing a leading membership role. On this platform, applications can be developed using Java to run on a client machine and can transfer message to other applications running on a server. Android also has several features as it is based on a monolithic Linux kernel which includes drivers for mobile device hardware like screen, keyboard, camera, Universal Serial Bus (USB), and Bluetooth, among others. There are also native libraries dependent on the underlying hardware architecture of mobile devices. All these have made Android a highly preferable operating system for the wider range of mobile devices. Taking these advantages of Android and Google cloud as an option, it is feasible to design and develop a notification and data message enhanced mobile learning application [17–24].

Generally, in this work, the proposed framework of architectural design of mobile learning application for higher education incorporates major components such as:

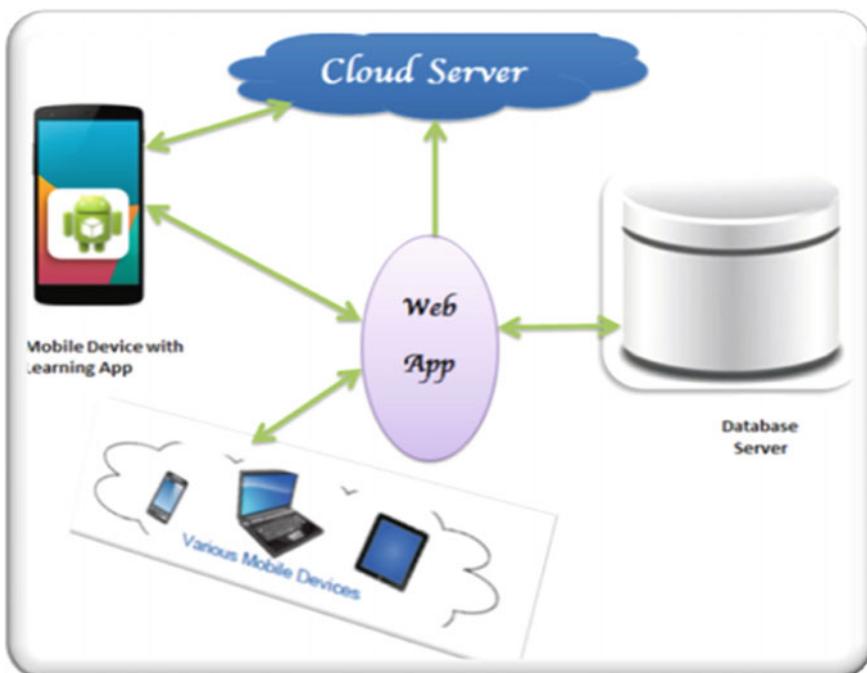
- Educational content management system that hosts the contents to be accessed by mobile devices through mobile app and any other Internet-connected devices using a browser.
- Mobile devices used as a terminal to deploy the mobile app and to launch the learning content from the content management system.
- Mobile app that is developed based on the proposed architecture.
- Part of cloud services offered by Google.
- Persistent data storage component.
- Both mobile and non-mobile devices that interact with the learning content management.
- And, the line of interactions between these component.

In addition to these, the way the architectural components communicate through both wireless and wire line communication technology is seamlessly considered. Finally, to test the proper functionality of the overall components, the implementation aspect was done in agile process model through prototyping.

### 3 The Proposed Architecture

The proposed architecture of mobile learning system for higher education has a couple of interacting components as depicted in Fig. 1. In order to achieve the proper functionality of the whole system, the role and method of interaction for every component were specified independently through the agile approach. As the core of the design process is on developing hybrid architecture for mobile learning application, modeling this component was the kick-off procedure. Then, the sequential integration of the remaining components followed. Furthermore, the internal structural design took the Model View Controller (MVC) pattern. This was applied to manage separation concern so that when modifying one aspect, the other should remain unchanged. Generally, the proposed system's physical architecture is as depicted in the figure below.

The architecture has five major components: the cloud server, the mobile learning app, the Web app at the center, mobile devices to access the learning app, and a database server to persistently maintain the learning application.



**Fig. 1** Proposed hybrid mobile learning architecture

## 4 Prototype Implementation

In this study, the intended mobile learning system architecture consists of external and internal components. The users/actors of the system are considered and modeled as outside environment/component of the system, whereas the actual mobile learning application's architectural framework is analyzed and designed separately as internal aspects of the system. Generally, the functional requirements are specified as per the specification of the proposed system architecture. Therefore, the detailed specification and implementation of each component are done here.

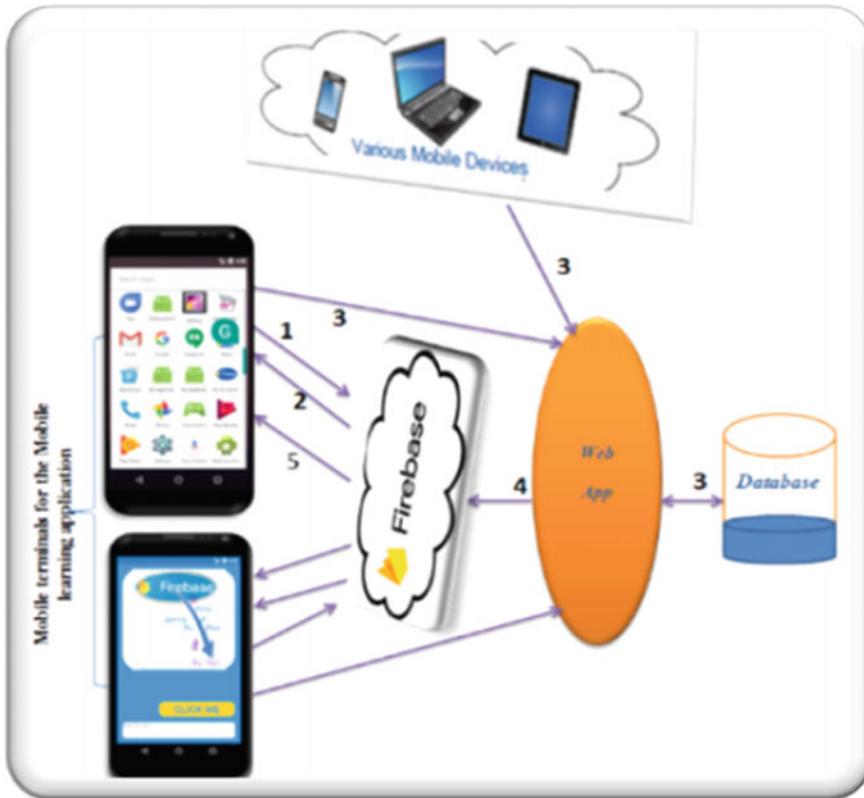
For the prototype implementation in this study, some major and basic functional requirements were identified by agile approach. Since the suggested mobile app is for Android platform, in the Android studio project app file has the following contents. These are:

- **Activity classes:** This represents the user interface components of the app on screen of a mobile terminal. It can hold several subcomponents as themes, views, etc.
- **Views:** It is a single element on a screen of activities and can be considered as a building block of the activities. Examples are Button, ImageView, TextView, and WebView.
- **Services:** Used to handle functionalities that run at the background.
- **Intents:** This is the part that handles the mechanisms of navigating between activities through intent messages relaying.
- **Firebase instance classes and services:** This part is intended for capturing the functionalities of integrating the mobile app with the Google cloud service FCM.

By applying the above on Android studio, all the necessary functionalities of the mobile app have been accomplished. The required functionalities are used for interfacing with the whole system components.

In this study, the core part is settling an architectural framework and the mechanism for sending push notifications and messages from the learning content management Web page. The process of managing tasks related to notification is facilitated by FCM. However, to enable a third-party Web application to utilize this cloud services, on the console of the firebase, there is an option of adding firebase to the Web app. Therefore, the learning content management Web application suggested has incorporated a Web page integrated with the firebase. Hence, for the mobile learning application in the study, this is the major component to broadcast a notification message to the Android mobile app.

According to the literature survey so far in mobile learning, all of them had used a browser to have access to the learning content on the Internet. Using hybridized mobile app for learning context in this research is one of a kind. Besides, seamlessly integrating the FCM service to the learning scenario is also another important contribution. The architectural framework of the whole system was framed by integrating those subcomponents. In Fig. 2, the detailed specification of how the system components are interacting is described.



**Fig. 2** Designed system component physical architecture and their interactions

The general interaction between the major structural components of the mobile learning application takes the following steps:

1. The mobile learning app on a mobile terminal will request for Reg\_ID from FCM server. This is performed only once when the app runs on a particular device for the first time. The step can be termed as **registration process** on FCM.
2. The FCM server will send back the unique Reg\_ID for the requesting terminal/mobile device. It is the **process of granting** of the FCM services up on **successful registration**.

The terminal then **stores its Reg\_ID** on a separate database through the Web app. This activity is performed to make the Reg\_ID available to the third-party application. The Web app of the mobile learning system is accessible by various devices. When someone wants to send a push notification message to a registered mobile terminal from the Web app, it can do so via Fir.

3. Reg\_ID of the specified terminal is retrieved from the database.

4. **Send the push notification to the FCM server.** Retrieving the Reg\_ID from the database, the Web application can attach a notification message to the ID and send to the FCM server.
5. Finally, the FCM server will **send the push notification to the specified terminal(s)**. The process of delivering the push notification to the targeted device(s) is done by the FCM server only. Therefore, after receiving the message from the Web application in step 4, the FCM pushes the message to the particular device.

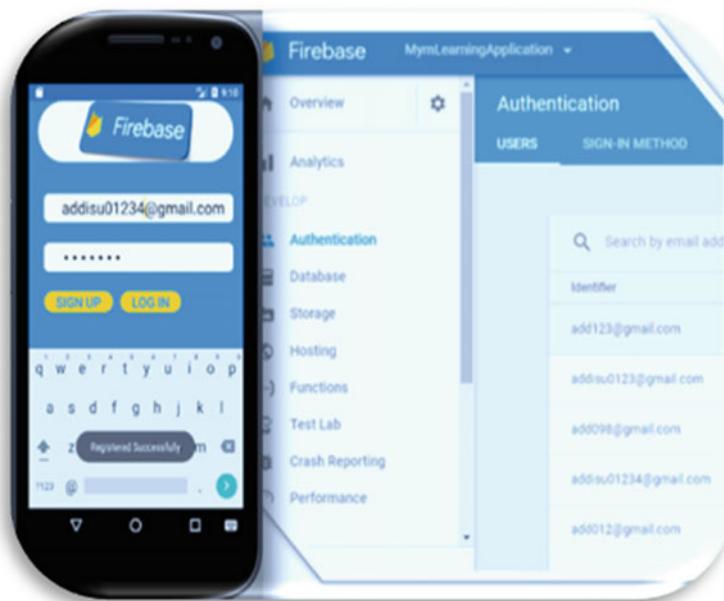
## 5 Validation and Evaluation

Here, the core functional components and usage scenarios are validated and evaluated from the complete hybrid architecture mobile learning system implemented. For that a couple of scenarios are presented below:

1. The Case of User Authentication with the Firebase System: One of the main objectives set was seamlessly integrating the Google cloud services with the mobile learning application. In this case, the particular service used was FCM. So, based on the implementation in the last section, this functionality was evaluated through authentication.
2. Actually, the FCM console can provide different authentications like in Gmail, Google++ and Facebook, for instance. But, for this case, only the Gmail case is tested and worked well as shown in Fig. 3.
3. The Case of Sending and Receiving Push Notifications: The another core functional element selected for the evaluation of the test case is the scenario of relaying notification message particularly from the server to online mobile devices running the app of mobile learning implemented in the previous section. The proper functionality of this case is tested by sending simple push notification from the firebase server using the console interfaces as illustrated in the Fig. 4. As it can be observed from the figure, the activity of sending the notification message from the server has worked fine.



**Fig. 3** User authentication using FCM console



**Fig. 4** A scenario of sending push notification

## 6 Conclusions and Future Works

In this work, to achieve the targeted objectives several procedural activities are performed to this end. As per the survey made and the gap identified, designing hybrid architecture of mobile app is taken as a suitable option for mobile learning applications. For enhanced functionality like receiving and sending push notifications, the mechanism of integrating the app with FCM cloud services is incorporated.

Agile process model was followed to deal with all the dynamicity of functionalities in the design and implementation of the proposed architecture. Basically, in implementation, the application has two major aspects as learning content management and Android application. The Android application is developed using Android studio and tested on virtual mobile devices or emulators. Then, it is verified for various usage and functionality scenarios to accessing overall mobile learning system. Generally, from the evaluation of the system functionality in the prototyped implementation, it is possible to say that the proposed physical architecture is implementable and usable.

The primary focus in designing and implementing the hybrid architecture mobile learning application in this study is for the case of Android platform only aided by FCM. However, recently Google has introduced FCM as a cross-platform approach for Android, iOS, and for Web apps. For cloud integration of mobile apps, this option

is very good. Yet, the remaining work will be settling architecture or a mechanism for designing a single system for all platforms.

To come up with FCM integrated and cross-platform architecture of mobile learning application, among the various options are:

- Designing a plugin architecture that can handle the process of integrating mobile applications with cloud services. The architecture should also deal with the task managing cross-platform-related issues.
- Designing a middleware architecture that can handle all the stuffs of making the mobile learning application cloud integrated and cross-platform.

## References

1. Rikala, J., & Kankaanranta, M. (2013). Mobile learning. A review of current research. *Reports of the Department of Mathematical Information Technology Series E. Educational Technology*, E, 1–65.
2. Bidin, S., & Ziden, A. A. (2013). Adoption and application of mobile learning in the education industry. *Procedia-Social and Behavioral Sciences*, 90, 720–729.
3. Martin, F., & Ertzberger, J. (2013). Here and now mobile learning: An experimental study on the use of mobile technology. *Computers and Education*, 68, 76–85.
4. Zbick, J. (2017) A web-based reference architecture for mobile learning: Its quality aspects and evaluation. In *IEEE International Conference on Software Architecture Workshops (ICSAW)*. IEEE.
5. Filho, D., Freitas, N., & Barbosa, E. F. (2015). A contribution to the establishment of reference architectures for mobile learning environments. *IEEE Revista Iberoamericana de Tecnologias del Aprendizaje*, 10(4), 234–241.
6. Duarte Filho, N. F., & Barbosa, E. F. (2014). A service-oriented reference architecture for mobile learning environments. In *Frontiers in Education Conference (FIE)*. IEEE.
7. Razaque, A., & Elleithy, K. (2011). Architecture based prototypes for mobile collaborative learning (MCL) to improve pedagogical activities. In *2011 14th International Conference on Interactive Collaborative Learning (ICL)*. IEEE.
8. Wang, N., et al. (2017). Design of a new mobile-optimized remote laboratory application architecture for m-learning. *IEEE Transactions on Industrial Electronics*, 64(3), 2382–2391.
9. Abarghooei, M. (2015). Designing a cross-platform mobile learning system. *Lecture Notes on Software Engineering*, 3(3), 195.
10. Malgaonkar, S., et al. (2014). Multipurpose android based mobile notifier. In *2014 International Conference on Advances in Electronics, Computers and Communications (ICAEC)*. IEEE.
11. Sharma, A., Eastham, P., & Nerieri, F. (2014). Designing an energy-efficient cloud messaging service for smartphones. *IEEE Pervasive Computing*, 13(1), 84–88.
12. Heckman, R. (2016). *Designing platform independent mobile apps and services*. Wiley.
13. Evers, S., Ernsting, J., & Majchrzak, T. A. (2016). Towards a reference architecture for model-driven business apps. In *2016 49th Hawaii International Conference on System Sciences (HICSS)*. IEEE.
14. KaiFang, F. (2009). Design and implementation of an instant messaging architecture for mobile collaborative learning. In *ISECS International Colloquium on Computing, Communication, Control, and Management, 2009, CCCM 2009*, Vol. 3. IEEE.
15. Nie, J. (2015). Research on mobile learning platform construction in higher vocational colleges based on cloud computing. In *2015 11th International Conference on Computational Intelligence and Security (CIS)*. IEEE.

16. Miao, G. (2013). The construction and development of university mobile learning system. In *2013 6th International Conference on Information Management, Innovation Management and Industrial Engineering (ICIII)*, Vol. 2. IEEE.
17. Cenka, B. A. N., & Hasibuan, Z. A. (2013). Enhancing educational services using cloud technology. In *2013 International Conference of Information and Communication Technology (ICoICT)*. IEEE.
18. Ramírez-Donoso, L., et al. (2017). Enhancing collaborative learning in higher education online courses through a mobile game app.
19. Fuad, M. M., & Deb, D. (2017). Cloud-enabled hybrid architecture for in-class interactive learning using mobile device.
20. Nabi, S. A., Gurram, D., & Ali, M. A. (2015). Mobile hybrid cloud computing for educational institutions: Mobihybrid educloud. In *2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*. IEEE.
21. Baccari, S., et al. (2016). A comparative study of the mobile learning approaches. In *International Conference on Mobile, Secure and Programmable Networking*. Springer International Publishing.
22. Klamma, R., Spaniol, M., & Cao, Y. (2006). Community aware content adaptation for mobile technology enhanced learning. In *EC-TEL*.
23. Lee, K., & Razaque, A. (2011). Suggested collaborative learning conceptual architecture and applications for mobile devices. In *Design, User Experience, and Usability. Theory, Methods, Tools and Practice* (pp. 611–620).
24. Baccari, S., et al. (2017). Comparative study of the mobile learning architectures. In *E-Learning, E-Education, and Online Training: Third International Conference, eLEOT 2016*. Dublin, Ireland: Springer International Publishing, 31 Aug–2 Sept 2016, Revised Selected Papers.

# Using Collaborative Robotics as a Way to Engage Students



**Lina Narbutaité, Robertas Damaševičius, Egidijus Kazanavičius  
and Sanjay Misra**

## 1 Introduction

Demand for science, technology, engineering, and mathematics (STEM) professionals is expected to grow by 8% each year until 2025 [1]. Germany, for example, is short of 210 000 workers in mathematics, computer science, natural sciences, and technology (MINT) disciplines [2]. However, there is a high-skill shortage in the STEM fields despite high unemployment rates in many countries, including European Union [3]. Many countries are suffering from low achievement and low interest among learners in STEM subjects compared to others.

STEM fields are core technological underpinnings of an advanced society and also related to the economic competitiveness of nations. Entrepreneurship, creativity, communication, and teamwork skills are needed to produce multidisciplinary scientific knowledge and innovation. However, traditionally schools and universities have struggled with the problem of embedding creativity into the STEM curriculum and attracting learners into STEM subjects. It is, therefore, important for education institutions to seek new ways to teach and grow soft skills in order to increase student interest in scientific education and technology-related careers.

---

L. Narbutaité · R. Damaševičius

Department of Software Engineering, Kaunas University of Technology, Kaunas, Lithuania  
e-mail: lina.narbutaitė@ktu.lt

R. Damaševičius

e-mail: robertas.damasevicius@ktu.lt

E. Kazanavičius

Centre of Real Time Computer Systems, Kaunas University of Technology, Kaunas, Lithuania  
e-mail: egidijus.kazanavicius@ktu.lt

S. Misra (✉)

Department of Electrical and Information Engineering, Covenant University, Ota, Nigeria  
e-mail: sanjay.misra@covenantuniversity.edu.ng

The inclusion of aspects of STEM in early education provides a strong motivation as argued by the Framework on Science Education in Europe [4]. By encouraging children and young people to express themselves through the use of engineering and technology through art and design, they can be engaged more effectively than by current emphasizing of the challenge of mathematics and science skills and future benefits involved.

The STEM education curricula focus on reasoned and clear solutions to the very specific set of problems, while the curriculum of art education typically demonstrates uncertainty, ambiguity, and vagueness—an essential foundation of educational experiences focused on the development of creativity and innovation. DIY robotics, robotic toys (such as developed using the Arduino platform (<https://www.arduino.cc/>) can effectively serve this purpose, offering broad opportunities to showcase a practical value in different areas such as social and humanistic sciences, mainly developing student's creativity, problem solving, communication, art, media, and teamwork skills, while almost directly familiarizing with basics of physics, electronics, programming and mechanics, thus opening a broader perspective and making STEM more attractive.

Even light usage of technology having a believable and likable outcome can strongly motivate future student's perception of science and engineering education [5]. A problem of "introducing the technology" can be solved via hands-on showcases and tangible familiarization. By introducing new and familiar topics to the future students, the teacher can challenge them to work on real practical problems, make a work of art/design and by that attract them to the STEM-based topics. The design thinking implies a work methodology based on trials and iterations, i.e., the kids get to create, build, test, and evaluate solutions in iterative cycle, as well as learn to present their work to others, pitch ideas, and give critique.

The educational STEM programs implemented through robotics need the model for teaching young boys and girls who will be able to gain new skill and competences to address problems facing society. The obstacles to implement robotics as a part of formal and informal learning curriculum appear to be of the time-consuming nature of robotic activities and the need to have skilled teachers familiar with the field of robotics. The problem increases when paired with perceptions that robotics, similarly to other MST subjects, is hard, gender-biased, and not inviting for most learners [6].

In this paper, we propose and describe the educational approach, which directly aims at boosting creativity and competitiveness of schoolchildren also encouraging teachers to play a more active role in adapting innovative educational methods. The approach addresses the main problem of educational robotics; that is, most of the experiments involving robotic activities are not integrated into regular classroom activities; as they are carried out in after-school programs, or summer camps. Currently, we are not aware of any pedagogical framework nor learning scenarios aimed to design or redesign the formal education curricular based on the application of robotics in education. Therefore, a more integrative approach is required.

Here, we argue that the robots can be used not only for teaching schoolchildren and university students to learn the STEM subjects of science, technology, engineering, and mathematics, but also in social and humanistic sciences to increase the engage-

ment of young people in technology and facilitate the acquisition of transdisciplinary knowledge. In order to be able to gain new skill, it is necessary to address real-world problems using educational robotics-based STEM courses (in a narrow sense) and STEM programmes (in a wider context) as the model. Next, we analyze the state-of-the-art approaches in educational robotics for STEM, end especially, STEAM, i.e., STEM and Arts (or all other) disciplines. The subjects of STEM and Arts, or more widely, arts, social, and humanities (AHSS), in themselves are very different by nature. Artistically oriented people operate with imagery, metaphors, and emotions, while scientists employ numbers, and formulas. Scientists are objective and artists are subjective. However, creativity allows to bridge both domains. Both social and technological skills will be very important in the twenty-first century. Robotics-enabled intercultural education (IcE) and computer science (CSE) education can facilitate multidisciplinary and multicultural projects using a low-cost, easily exported robot platform that allows students to expand their academic and personal experiences. The immediate feedback offered by robot behavior and the confidence that can help students overcome linguistic and cultural obstacles in acquiring twenty-first-century skills.

## 2 Pedagogical Backgrounds and Preconditions

The physical tangibility of robots raises the need for a shift to innovative and effective teaching methods for the engagement robots provide is considered conducive for learning. Schools have to be relocated at the center of the society to bring both boys and girls into the scientific world.

Engaging learners in multidisciplinary problem solving using educational robotics based on sound pedagogical framework requires academic restructuring of traditional educational models; therefore, transdisciplinary formal and informal educational STEM programs through robotics will be at the forefront of this transition.

Robot-aided learning (r-Learning) [7] has enough potential to be used as a tool of creativity in arts, humanities, and social sciences (AHSS) classes, thus attracting the attention of learners to cross-disciplinary subjects with elements of science, technology, engineering, and mathematics (STEM), where the learners can explore the combination of sculpture and robotics through the lens of art. Further on, the advantages of using robots in language instruction, known as robot-assisted language learning (RALL) [8], can be transferred to teaching other AHSS subjects. Actually spending time working with real robot-based examples gives the students many opportunities to see the topic from standpoints that are difficult or impossible to convey in a classical textbook-oriented lecture.

Discussions with industry leaders concerning characteristics to be cultivated in students suggest that they are looking for creative and innovative people (described as “thinking outside the box”), those who can work in teams with other people. Traditional STEM education based on constructionism focuses on the convergent skills, whereas social science and art focus on the divergent skills. Having the ability

to execute both at scale can better position the young people of Europe for global competitiveness [9]. Educational robotics can be employed to inspire curiosity and creativity in students [10].

Robotics enables recognizing the world by trying and doing rather than by observing or listening. The main appeal arises from the potential of educational robotics to enhance student's intellectual, social, emotional, physical, and artistic development and to foster creativity and a lifelong love of learning. The way robotics is currently introduced in educational settings usually focuses just on a narrow subset of topics mainly in the field of mathematics and physics. Further on, the interrelation between science and art is also reflected, and here the term "robotic art" emerges. "Robotic art" [11] is a type of art that makes use of robotics and automated technology, coupled with computer technology and sensors. Robotic art attracted attention with the rise of electronic media and technology in art.

Despite these nascent efforts, there is lack of pedagogical scenarios and methodological background in order to use educational robotics in non-STEM classes to attract students to STEM more systematically. There has been some effort in the context of Science, Mathematics, Art, Robot, and Technology (SMART) with little emphasis on the Art part of SMART [12]. Exploring a wider range of possible applications for robotics in the context of STEAM such as poetry, history, human anatomy [13], and biology [14] can engage young people (both girls and boys) in undertaking scientific careers. The schoolchildren interested in arts, humanities, and social sciences (AHSS) still could be attracted to interdisciplinary studies (e.g., design engineering) involving a significant part of technological subjects, if properly addressed and motivated. Instead of focusing on a single technological challenge such as design of an autonomous robotic carriage for line following or obstacle avoidance, robots could be deployed in a more creative environment such as development of robotic musical instruments for music-oriented students, development of wearable art with computing capabilities for art- and design-oriented students, and creation of robotic characters for humanities-oriented students. This is why the goal is to embrace the significant potential of educational robotics for boosting problem-solving and teamwork skills [10].

It is a known fact that investment in the pure STEM fields—science, technology, engineering, and mathematics—increases innovation and supplements to the economic development, which is very important to the developing countries. The decrease in unemployment through a positive notion of social impact of robots is expected. Educational robotics also can indirectly reduce the fear of robots as an alternative workforce and increase the familiarity of technical objects. A more direct economic impact will result from schoolchildren as future workforce which generates country gross domestic product (GDP) and makes it competitive in the world's economics. Consequently, the development invites growth in new jobs in a community.

Educational robotics can introduce the design, artistic, and creative processes to informal learning through the emphasis on engineering knowledge, improving student engagement, and reducing boundaries between different disciplines, establishing a synergistic relationship. By using robots as an art form and attracting artists who include science in their artworks, the interest of not only students, but educators

and researchers in STEAM disciplines can be increased. Introducing young people to hybrid works of art and technology and by offering a robot as an educational art tool, we will be able to help young people to understand more about the mix of STEM subjects with artistic/creative process, design thinking, and computational thinking [15, 16]. Robotics can alleviate the lack of interest toward STEM subjects in students and has the potential to change students' view on learning of STEM subjects [17]. The hands-on, imaginative approaches to science education, combining simple robotics with many of the methods used in the creative arts and design are aimed to attract and retain young people in the fields of STEM. Using educational robotics, Robotic Art can serve as a tool to facilitate STEAM learning [18]. Moreover, the value of educational robotics is in its applicability for all age groups, including kindergarten [19], schoolchildren from first grade to 12th grade [20], and university students [17], both in the formal education setting and in homeschooling environment [21].

However, the successful implementation of STEM education, however, requires the preparation of core activities [22], which in our case is the model-based design of educational robots. The methodology is described in more detail in Sect. 3.

### 3 Methodology

The methodology of using educational robotics in the classroom described in this paper is based on the duality of teamwork learning and collaborative robots. The approach is based on the duality between the problem domain, which in our case is the problem of team building and management of teamwork, and the application domain, where team building is implemented and explicitly visualized by collaborative interacting and communicating robots. The idea itself is not entirely new as is known in the domain of software engineering. In product line engineering (PLE), domain engineering focuses on the capture of knowledge, while application engineering reuses that knowledge for developing specific systems [23]. In agile software engineering, the dependencies between requirements and architecture have been described as the “twin peaks of requirements and architecture” [24]. The model focuses on build successful and cost-effective software systems by supporting the co-development of requirements and architecture.

Shulte [25] introduced the concept of educational lenses for the duality deconstruction of Informatics Systems for teaching ICT topics. The approach provides an example of word processor as command-based knowledge that leads to inefficient teaching model that focuses on introducing and practicing commands rather than giving the learner additional (strategic) knowledge. From the socio-technical perspective, all Information Technology (IT) artefacts are dual artefacts, which have technical structure focusing on data, algorithms, and operations, yet their function is social, which very much depends on its surrounding social context [26]. One of the outcomes of such duality is the crisp separation of developers (designers) and users of IT. While designers focus on creating new systems and products, the users are only capable of utilizing pre-given application. The results are the insider/outsider

[27] or the creator/consumer dichotomy, which in itself contributed to many misconceptions in computer science education, not the least being the notable gender gap in enrollment to computer science subjects [28].

Rethinking of the processes and concepts of IT reformulation in terms of duality [29] allow a deeper and more dialectical understanding of the interaction between technology and socium and has important implications for the development and use, especially for teaching of IT and computer science topics. The ability to understand and strengthen specific patterns of human behavior [30], which are recurring across different types of systems and domains can ease the steep learning curve of modern IT and computer science.

The development of complex modern IT systems such as multi-agent systems, artificial neural networks (ANNs), bio-inspired algorithms, or swarm robotics encounters nowadays a real challenge. Nevertheless, the similarity of such systems to real-world living systems or social networks provides an inspiration for better understanding of both technical systems and their real-world counterparts. Conceptualization of social processes and forces in terms of explicitly visible and physically tangible objects such as collaborating education robots allows to better understand their principles of function as well as to overcome the mechanistic view on technical systems as closed ones [31]. The approach also can contribute toward developing and enhancing non-technical skills of students in IT education [32].

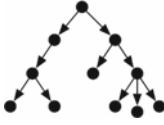
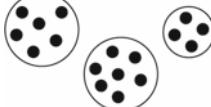
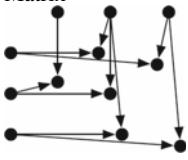
Systematic production of graduates who have critical skills of teamwork and team management and are employable in modern global workplace is a challenge to IT education. Practical knowledge and experience are essential to the formation of teams [32]. Effective deliverance of such knowledge and skills requires rethinking of traditional teaching and learning methods. The results of the study [33] show that the practice activities performed in cooperation had improved group performance and behaviors. The idea to apply project-based collaborative approach to educational robotics is not new and has been successfully applied elsewhere (see, e.g., [34]).

Following Hagen and Bouchard [32], the implementation of the simulated project with inclusive training of non-technical skills can be implemented in a four-step instructional process:

- Concept–Problem Identification: The students identify the technical problem and its dual social counterpart
- Cognitive reframing of problems to find viable solutions
- Implementation, which includes both traditional technical process of software programming and hardware assembly as well as social relationship building
- Evaluation and self-assessment both in terms of technical characteristics of created product and ore wider context of soft skills (communication, collaboration, team management, including conflict management) acquired.

Moreover, our methodology is based on the next-generation science standards (NGSS) framework practices [35] as follows: (1) asking questions and defining problems, (2) developing and deploying models, (3) planning and performing experiments, (4) analyzing and explaining data, (5) engaging in computational thinking,

**Table 1** Conceptual correspondence of between robotic and human teams

Type of Organization	Robotic teams	Human teams
Hierarchy 	A single leader has the authority to make decisions. Upper-level agents control lower-level agent	Leader (manager) assigns them sub-tasks and resources to complete the task
Coalition 	Loosely affiliated organization without a clear leader	Individual members take initiative to implement their ideas, while the burden on leaders to make every decision is reduced
Cooperative 	Agents cooperate to achieve common aim	An autonomous association of persons united voluntarily to meet their common needs
Matrix 	Agent capabilities and work commitments are shared between multiple leaders	A flat organization has few or no levels of middle management between staff and executives

(6) selecting best solutions, (7) discussing the results based on evidence, and (8) communicating the results and experience.

The domain of robotic teams is organized by applying the taxonomy of organizational paradigms in multi-agent systems presented in [36], which include hierarchy, coalition, cooperative, and matrix organizations (see Table 1). The conceptual similarity between robotic agents and humans helps to underscore the importance of teamwork in task execution. The students are introduced with the possible structure of their teams and can choose freely how to organize their team.

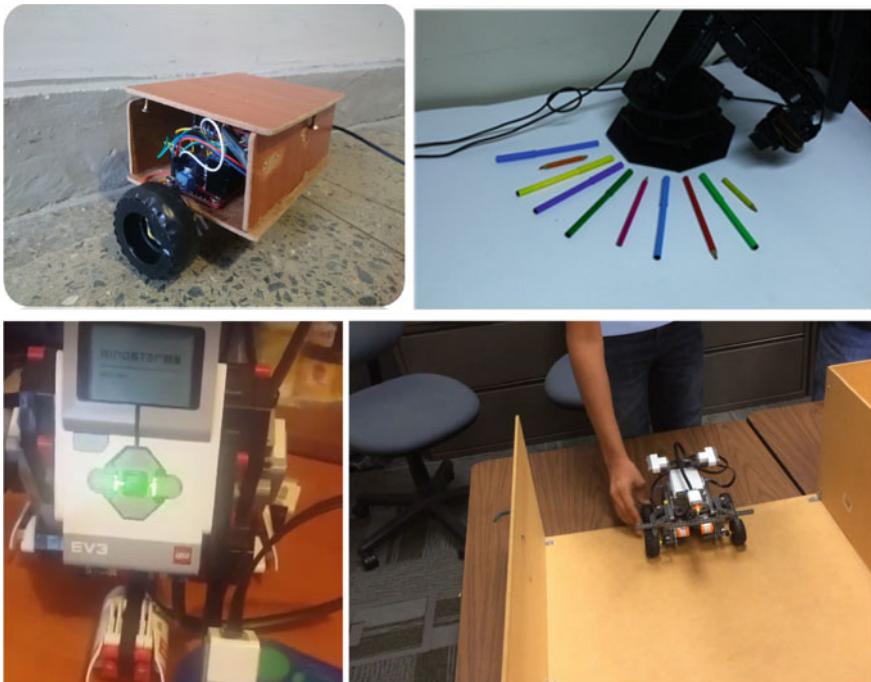
## 4 Case Study

The approach described in this paper has been adopted in Kaunas University of Technology, Faculty of Informatics. The students in the Robotics Programming Technologies Course (a part of Software Systems study programme), during 2012–2016, in total 293 students (2012–34, 2013–33, 2014–51, 2015–86, 2016–89). The course aims to teach students of the basic principles of robot programming and control using the collaborative teamwork approach [37, 38]. During 5 years, the students have implemented 89 team projects.

The robot hardware used during laboratory works included two LEGO NXT robots with NXT Intelligent Brick, Arduino 4WD Mobile Platform with ATmega328 microcontroller board, Lynxmotion 5LA Robotic Arm robotic arm with 6DOF, and a two-fingered gripper. The control was implemented using the SSC-32 protocol.

The course is based on project- and team-based approaches to teaching the principles of robotic programming. The approach involves giving students a robot to assemble, and providing increasingly complex challenges to solve, starting from simple line following to roaming in a crowded, dynamical changing environment. The use of entertaining ideas for project is encouraged as gamification plays an important role in student engagement and interest sustainment [39].

Students work on their semester assignments in groups of 3–4 students. The adopted learning scenario is as follows:



**Fig. 1** Examples of implemented robotics projects

- (1) A team of students are presented with a typical robotics problem (such as line following, obstacle avoidance) and materials required for solving it.
- (2) The students analyze study literature and select the most appropriate solutions under the guidance of the teacher. The design and modeling of a robotic system involve the use of visual programming environments [40] such as Microsoft Robotics Developer Studio, NXT-G, or Virtual Robot Experimentation Platform (V-REP).
- (3) The students construct, model, and implement a robot using the robot modeling and programming environment required to implement the task.
- (4) The students empirically validate the solution by performing several experimental runs of the robot.
- (5) The students present their implemented robot to other students and the teacher at the semester workshop. Presentation framed as a learning object (LO) [41–43] is encouraged; that is, the students formulate their educational aims, describe the implementation of the projects using multimedia materials (videos, photographs, diagrams), and present conclusions what they have learned. The Moodle learning platform is used as a common media to discuss projects and share learning experience.

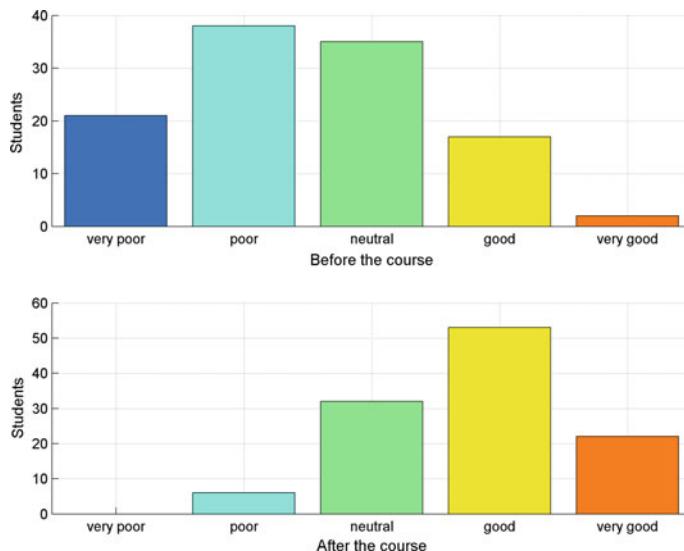
Some of the implemented robotics projects are illustrated in Fig. 1.

## 5 Evaluation

After the students completed the course, they were asked to complete a self-assessment survey and evaluate their abilities before and after the course: The students were asked to evaluate their abilities before and after the course as follows:

1. Ability to model and construct typical robots using a programming, modeling, and imitation environment Virtual Robot Experimentation Platform (V-REP) and robotic platforms (Arduino, Lego).
2. Able to explain robot control architectures and apply robot control algorithms.
3. Ability to design and implement the control of a typical robot.

The survey used a five-item Likert rating corresponding to as follows: 1. Very poor; 2. Poor; 3. Neutral; 4. Good; 5. Very good. The aggregate survey ( $N = 113$ ) results are presented in Fig. 2. The results show that students overall have improved their knowledge in the field of robot programming.



**Fig. 2** Aggregated survey results before and after course completion

## 6 Conclusion

Educational robotics focuses on the link between physical materials of the educational actions and the virtual ones like project Web site, student and teacher online support tools (e.g., Moodle) and to cultivate creativity and problem-solving skills via easy accessible robotic do-it-yourself (DIY) experiences, looking at it not as a robot hardware, but as a virtual storyteller (such as traveler in the labyrinth) in the integrated learning environment. The use of Moodle as the robotic learning (r-Learning) environment assures the interactivity of the educational content and the effective knowledge assimilation.

Educational robotics can be used to raise awareness of the career path of young girls and boys to successfully meet STEAM (STEM + Art) challenges in the educational process and to stimulate attractiveness of science education in line with the principles in gender equality for development of innovative, creative, and sustainable societies in Europe and elsewhere. The approach is especially relevant for the developing countries such as India or Nigeria. In addition, the integrated teaching approach fosters a stronger overlapping between formal, informal, and non-formal learning to make scientific and technological as well as transdisciplinary careers attractive to young students, increasing competitiveness of young people on the job market in the future, establishing a clear link between creativity and science, and providing with critical teamwork skills.

## References

1. European Commission. (2014). EU Skills Panorama. Skills challenges in Europe. Analytical Highlight.
2. Directorate general for internal policies. (2015). *Encouraging STEM studies: Labour market situation and comparison of practices targeted at young people in different member states*.
3. Dobson, I. (2013). STEM: country comparisons—Europe. In *A critical examination of existing solutions to the STEM skills shortage in comparable countries*. Melbourne: Australian Council of Learned Academies.
4. Forsthuber, B., Motiejunaite, A., & de Almeida Coutinho, A. S. (2011). *Science education in Europe: National policies, practices and research*. Education, Audiovisual and Culture Executive Agency, European Commission.
5. Flegg, J., Mallet, D., & Lupton, M. (2012). Students' perceptions of the relevance of mathematics in engineering. *International Journal of Mathematical Education in Science and Technology*, 43(6), 717–732.
6. Blikstein, P. (2013). Digital fabrication and ‘making’ in education: The democratization of invention. In J. Walter-Herrmann & C. Büching (Eds.), *FabLabs: Of machines, makers and inventors*. Transcript Publisher.
7. Han, J. (2010). Robot-aided learning and r-learning services. In D. Chugo (Ed.), *Human-robot interaction*, Vol. 288. Croatia: INTECH.
8. Aidinlou, N. A., Alemi, M., Farjami, F., & Makhdoumi, M. (2014). Applications of robot assisted language learning (RALL) in language learning and teaching. *International Journal of Language and Linguistics*, 2(3–1), 12–20.
9. Madden, M. E., Baxter, M., Beauchamp, H., Bouchard, K., Habermas, D., Huff, M., et al. (2013). Rethinking STEM education: An interdisciplinary STEAM curriculum. *Complex Adaptive Systems*, 2013, 541–546.
10. Zawieska, K., & Duffy, B. R. (2015). The social construction of creativity in educational robotics. *Progress in Automation, Robotics and Measuring Techniques*, 2, 329–338.
11. Kac, E. (1997). Foundation and development of robotic art. *Art Journal*, 56(3); *Digital reflections: The dialogue of art and technology* (pp. 60–67).
12. Hong, S. Y., & Hwang, Y. H. (2012). A study on smart curriculum utilizing intelligent robot simulation. In *Issues in Information Systems, IACIS* (Vol. 13, No. 2, pp. 131–137).
13. Hamner, E., & Cross, J. (2013). Arts & bots: techniques for distributing a STEAM robotics program through K-12 classrooms. In *Proceedings of the Third IEEE Integrated STEM Education Conference*, Princeton, NJ, USA.
14. Rahman, A., Saleh, A., & Abdelbaki, N. (2017). Innovative human-robot interaction for a robot tutor in biology game. In *18th International Conference on Advanced Robotics (ICAR)* (pp. 614–619).
15. Burbaitė, R., Drasutė, V., & Stuikys, V. (2018). Integration of computational thinking skills in STEM-driven computer science education. In *IEEE Global Engineering Education Conference, EDUCON* (pp. 1824–1832). <https://doi.org/10.1109/educon.2018.8363456>.
16. Štuikys, V., Burbaitė, R., Blažauskas, T., Barisė, D., & Binkis, M. (2017). Model for introducing STEM1 into high school computer science education. *International Journal of Engineering Education*, 33(5), 1684–1698.
17. Khanlari, A. (2013). Effects of educational robots on learning STEM and on students' attitude toward STEM. In *2013 IEEE 5th Conference on Engineering Education (ICEED)*, Kuala Lumpur (pp. 62–66). <https://doi.org/10.1109/iceed.2013.6908304>.
18. Daugherty, M. K. (2013). The Prospect of an ‘A’ in STEM education. *The Journal of STEM Education*, 10–15.
19. Ioannou, M., & Bratitsis, T. (2017). Teaching the notion of speed in kindergarten using the sphero SPRK robot. In *IEEE 17th International Conference on Advanced Learning Technologies, ICALT 2017* (pp. 311–312). <https://doi.org/10.1109/icalt.2017.70>.

20. Wang W. H. (2016). A mini experiment of offering STEM education to several age groups through the use of robots. In *2016 IEEE Integrated STEM Education Conference (ISEC)* (pp. 120–127), Princeton, NJ. <https://doi.org/10.1109/isecon.2016.7457516>.
21. Plaza, P., Sancristobal, E., Carro, G., & Castro, M. (2017). Home-made robotic education, a new way to explore. In *2017 IEEE Global Engineering Education Conference (EDUCON)* (pp. 132–136), Athens. <https://doi.org/10.1109/educon.2017.7942837>.
22. Yamagata, H., & Morita, T. (2017). Design of contest for educational underwater robot for STEM: Learning applying modeling based on control engineering. *Journal of Robotics and Mechatronics*, 29(6), 957–968. <https://doi.org/10.20965/jrm.2017.p0957>.
23. Frakes, W. B., & Kang, K. (2007). Software reuse research: Status and future. *IEEE Transactions on Software Engineering*, 31(7), 529–536.
24. Cleland-Huang, J., Hanmer, R. S., Supakkul, S., & Mirakhori, M. (2013). The twin peaks of requirements and architecture. *IEEE Software*, 30(2), 24–29.
25. Schulte, C. (2012). Uncovering structure behind function: The experiment as teaching method in computer science education. In *Proceedings of the 7th Workshop in Primary and Secondary Computing Education (WiPSCE '12)* (pp. 40–47). New York, NY, USA: ACM.
26. Damasevicius, R. (2009). On the human, organizational, and technical aspects of software development and analysis. In *Information systems development: Towards a service provision society* (pp. 11–19). [https://doi.org/10.1007/b137171\\_2](https://doi.org/10.1007/b137171_2).
27. Crutzen, C. K. M. (2000). *Interaction, a world of differences. A vision on informatics from the perspective of gender studies*. Open Universiteit Nederland.
28. Limanauskienė, V., Rutkauskienė, D., Kersiene, V., Bareisa, E., Damasevicius, R., Maskeliunas, & R., Targamadze, A. (2017). The study of gender equality in information sciences research institutions in Lithuania. In *Information and Software Technologies, ICIST 2017; Communications in computer and information science* (Vol. 756, pp. 499–511). [https://doi.org/10.1007/978-3-319-67642-5\\_42](https://doi.org/10.1007/978-3-319-67642-5_42)
29. Orlikowski, W. J. (1992). The duality of technology: Rethinking the concept of technology in organizations. *Organization Science*, 3(3), 398–427.
30. Di Maio, P. (2014). Towards a metamodel to support the joint optimization of socio technical systems. *Systems*, 2014(2), 273–296.
31. Katsioloudes, M. I. (1996). Socio-technical analysis: A Normative model for participatory planning. *Human Systems Management*, 15, 235–244.
32. Hagen, M., & Bouchard, D. (2016). Developing and improving student non-technical skills in IT education. *A Literature Review and Model*, 3(7) (2016).
33. Cavalier, J. C., Klein, J. D., & Cavalier, F. J. (1995). Effects of cooperative learning on performance, attitude, and group behaviors in a technical team environment. *Educational Technology Research Development*, 43, 61–72.
34. Karaman, S., Anders, A., Boulet, M., Connor, J., Gregson, K., Guerra, W., & Vivilecchia, J. (2017). Project-based, collaborative, algorithmic robotics for high school students: Programming self-driving race cars at MIT. In *7th IEEE Integrated STEM Education Conference ISEC 2017* (pp. 195–203). <https://doi.org/10.1109/isecon.2017.7910242>.
35. Ziaeefard, S., Miller, M. H., Rastgaar, M., & Mahmoudian, N. (2017). Co-robotics hands-on activities: A gateway to engineering design and STEM learning. *Robotics and Autonomous Systems*, 97, 40–50. <https://doi.org/10.1016/j.robot.2017.07.013>.
36. Horling, B., & Lesser, V. (2004). A survey of multi-agent organizational paradigms. *Knowledge Engineering Review*, 19(4), 281–316. <https://doi.org/10.1017/S026988905000317>.
37. Plauska, I., & Damasevicius, R. (2014). Educational robots for internet-of-things supported collaborative learning. In *International Conference on Information and Software Technologies, ICIST 2014*. Springer; *Communications in computer and information science* (Vol. 465, pp. 346–358). [https://doi.org/10.1007/978-3-319-11958-8\\_28](https://doi.org/10.1007/978-3-319-11958-8_28).
38. Damasevicius, R., Narbutaitė, L., Plauska, I., & Blazauskas, T. (2017). Advances in the use of educational robots in project-based teaching. *TEM Journal*, 6(2), 342–348. <https://doi.org/10.18421/TEM62-20>.

39. Aseriskis, D., & Damasevicius, R. (2014). Gamification patterns for gamification applications. In *6th International Conference on Intelligent Human Computer Interaction, IHCI 2014*, Évry, France, 8–10 Dec 2014; *Procedia Computer Science* 39, 83–90. <https://doi.org/10.1016/j.procs.2014.11.013>.
40. Plauska, I., Lukas, R., & Damasevicius, R. (2014). Reflections on using robots and visual programming environments for project-based teaching. *Elektronika ir Elektrotechnika*, 20(1), 71–74. <https://doi.org/10.5755/j01.eee.20.1.6169>.
41. Burbaitė, R., & Štuikys, V., Damasevicius, R. (2013). Educational robots as collaborative learning objects for teaching computer science. In *IEEE International Conference on System Science and Engineering, ICSSE 2013* (pp. 211–216). <https://doi.org/10.1109/icsse.2013.6614661>.
42. Štuikys, V., Burbaitė, R., & Damaševičius, R. (2013). Teaching of computer science topics using meta-programming-based GLOs and Lego robots. *Informatics in Education*, 12(1), 125–142.
43. Štuikys, V., Burbaitė, R., Drasute, V., & Bespalova, K. (2016). Robot-oriented generative learning objects: An agent-based vision. In *10th KES International Conference on Agent and Multi-Agent Systems: Technology and Applications, KES-AMSTA: Smart Innovation, Systems and Technologies* (Vol. 58, pp. 247–257). Springer. [https://doi.org/10.1007/978-3-319-39883-9\\_20](https://doi.org/10.1007/978-3-319-39883-9_20)

# Assessing Scratch Programmers’ Development of Computational Thinking with Transaction-Level Data



Milan J. Srinivas, Michelle M. Roy, Jyotsna N. Sagri  
and Viraj Kumar

## 1 Introduction and Related Work

The importance of teaching computational thinking [1] to school-age children is now globally recognized. Initiatives such as Computer Science for All [2] and CSpathshala ([cspatshala.org](http://cspatshala.org)) seek to influence educational policies by integrating computational thinking into the school curriculum. For introducing programming, resources for beginners such as Code.org’s Hour of Code typically use block-based programming languages. One such language is Scratch [3, 4], which is popular with millions of users in more than 150 countries, particularly in schools [5]. In this paper, we present a modification to the Scratch programming environment for capturing data that can help assess the development of student programmers’ computational thinking skills, and a tool for visualizing this data. Our code is open source and is available at: <http://goo.gl/GBpeq4>.

Researchers in the Learning Analytics community have demonstrated that data from educational contexts can improve teaching–learning quality [6, 7]. This data can be viewed at three levels [8]. System-level data (which is typically stored in institutional or governmental information systems) includes courses taken by learners and course grades, and reflects the overall performance of learners and institutions.

---

M. J. Srinivas · M. M. Roy · J. N. Sagri  
PES University, Bengaluru, India  
e-mail: milan.j.srinivas@gmail.com

M. M. Roy  
e-mail: michellemroy96@gmail.com

J. N. Sagri  
e-mail: sagrijn@gmail.com

V. Kumar (✉)  
Indian Institute of Science, Bengaluru, India  
e-mail: viraj.kumar.cs@gmail.com

Individual-level data (which is often stored in local or cloud-based learning management systems) includes course-specific details for each learner such as test scores and scores on individual test items. Transaction-level data is specific to computer-based systems (e.g., programming environments or intelligent tutoring systems) and captures low-level details of how each learner interacts with the system over time. Analysis of this low-level data can identify knowledge component gaps and learner misunderstandings that cannot be found in higher-level data [9]. Such data also has the potential to help instructors efficiently identify learners who require assistance but remain silent [10, 11]. Transaction-level data is therefore important for the accurate assessment of learner abilities, but it is generally difficult to capture [6].

Although systems for capturing transaction-level data have been created for certain programming environments including AgentSheets [12], Alice [13], and specific research projects [14, 15], we are unaware of any logging system for the popular Scratch programming language that captures data at the fine-grained level we describe in this paper. For instance, Google Analytics has been used to gain some insights into overall patterns of behavior of programmers for a variant of Scratch called ScratchJr [16], but this data is not collected at the level of individual learners. Velasquez et al. [17] have analyzed data at the individual level, but this consists of open-ended textual comments in a networking forum.

Tools such as Hairball [18] and Dr. Scratch [19] assess the computational thinking abilities of Scratch programmers, but this analysis is performed on completed code artifacts. In the framework for computational thinking proposed by Brennan and Resnick [20], this analysis is therefore restricted to computational thinking concepts and does not reflect learner design practices. The latter can be inferred from transaction-level data [14], and hence our paper addresses an important requirement for applying these techniques to Scratch programming.

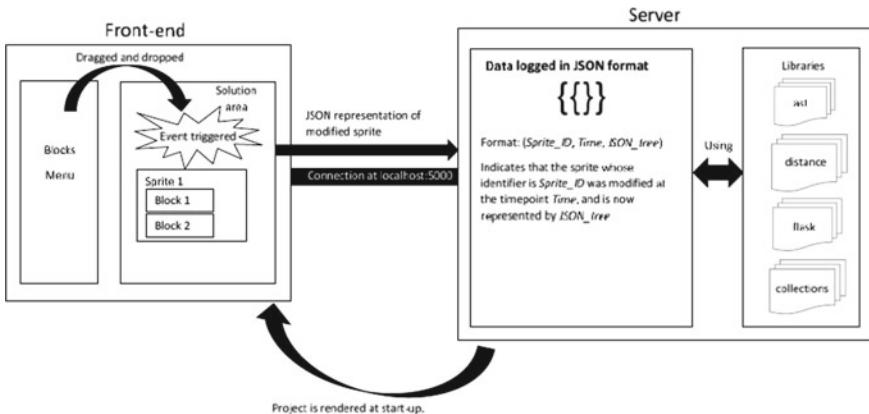
The rest of this paper is organized as follows. We describe the architecture of our transaction-level data stream logging system in Sect. 2. In Sect. 3, we illustrate the value of this type of data by building a data visualization tool that helps instructors understand computational thinking development of learners. Lastly, we present our conclusions in Sect. 4.

## 2 Logging Transaction-Level Data Streams in Scratch

A Scratch project is essentially a zipped archive including the project information file (text in the JavaScript Object Notation or JSON format) together with project media files (including sound clips and images). The project information file contains the project metadata (author information, project history, etc.), objects (known as sprites) that perform actions in the project, and code that manipulates these objects (known as scripts). Each sprite has a unique identifier. The programmer manipulates sprites by adding, modifying, or deleting blocks. Each block also has a unique identifier, as well as the identifier of its parent (which is either the preceding block within the



**Fig. 1** A Scratch sprite (left) with several individual blocks (labeled “go to”, “forever”, etc.) and its corresponding textual representation in JSON (right)



**Fig. 2** A system diagram of the Scratch environment. Our primary modification replaces the original Node.js server with a Python–Flask server that logs time-stamped user events in JSON format

same sprite or, in the case of the first block, the sprite itself). Thus, each sprite is encoded (in JSON format) as a string, as shown in Fig. 1.

The Scratch environment consists of a front end (with which the user interacts) and a server, as shown in Fig. 2. At any point in time, the user interacts with a specific sprite in the solution area (e.g., by dragging a new block from the Blocks Menu and dropping it into the solution space of this sprite). Each block is an SVG element, and we modify the file index.html (part of the Scratch virtual machine repository, responsible for rendering the programming environment) to add a “mousedown” event listener for each such block. We also add a “mouseup” event listener for the whole document to recognize when blocks are dropped into sprites. In the original Scratch code, the front end makes a jQuery AJAX call to the Node.js Server (running on port 8083) every 2 s, while the user is active, and after 6 s of inactivity. The data embedded within this AJAX call is the JSON representation of the user’s Scratch program. In our implementation, we modify the playground.js file (included by index.html) to

**Table 1** A time-stamped sequence of user events across 23 s, with changes to a fragment of the Scratch program (shown in JSON representation)

Time stamp	Program fragment (JSON representation)
1520435153	...': '18', 'shadow': ...
1520435157	...': '20', 'shadow': ...
1520435162	...': '20', 'shadow': ...
1520435174	... 'SUBSTACK', 'bloc...
1520435176	... 'SUBSTACK': { 'name...
...	...

make this AJAX call to a Python–Flask [21] server (running on port 5000) 200 ms after the “mouseup” event. The content of the AJAX call consists of the same data as before. The server time-stamps and logs this stream of user events, as shown in Table 1.

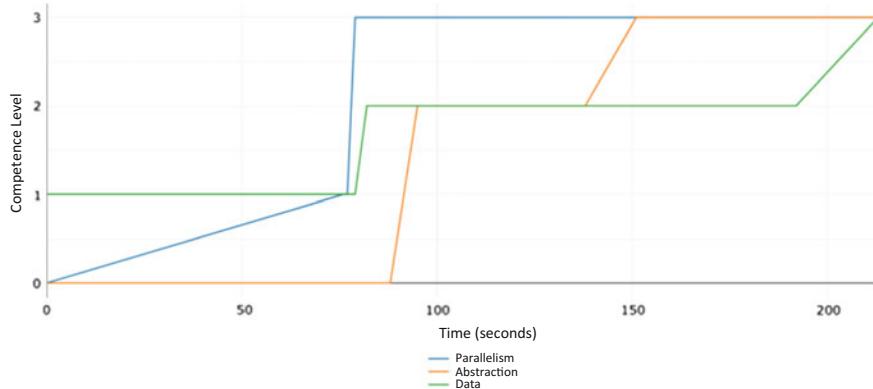
Analogous to the architecture of the REACT tool [12], it is possible to record this data stream online in a central server from where instructors can access the live data. In the country where this research was conducted, however, the lack of infrastructure in schools makes this approach infeasible at scale [22]. Hence, our tool records each learner’s data stream on the learner’s own device in a simple CSV format. These files must be transferred at a later stage to the instructor’s device for analysis.

### 3 Visualizing Learners’ Computational Thinking Development

There are three dimensions to Brennan and Resnick’s framework for computational thinking: concepts, practices, and perspectives [20]. This paper restricts attention to concepts (e.g., iteration and parallelism) and practices (the processes of thinking and learning, such as developing code iteratively, testing and debugging).

Several computational thinking concepts have been proposed [14, 19]. Methods to quantify learner competence in these concepts statically analyze the final artifact (the submitted program) created by the learner. Since our tool logs each step in the creation process, it is easy to apply these measures of competence at every step. As an example, Fig. 3 shows three measures of competence defined by Dr. Scratch [19] for one learner. Since this learner demonstrates the highest level of competence (3) for all concepts after less than 4 min of programming, the instructor can presume that this learner does not require immediate assistance.

As stated earlier, this information can be presented to instructors in real time provided the necessary infrastructure (a reliable central server to which learner and instructor devices can simultaneously connect) is available. However, even when this information is available only after a session, instructors may still find it valuable to note those learners who appear to be struggling as they prepare for the next session.



**Fig. 3** A plot of the competence of one student for three computational thinking concepts (parallelism, abstraction, and data) versus time. Competence (as defined in Dr. Scratch [19]) can be 0 (null), 1 (basic), 2 (developing), or 3 (proficiency)

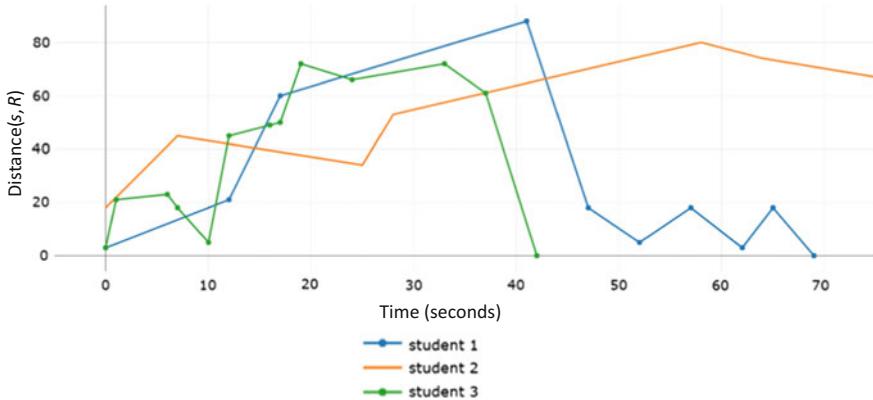
Brennan and Resnick [20] point out that development of computational thinking cannot be defined solely in terms of mastering concepts (the content of learning)—it also includes practices (the process of learning as learners engage with concepts) and learner perspectives. It is unlikely that transaction-level data can shed light on learner perspectives, but it can be used to assess all four types of programming practices identified in [20]: incrementally and iteratively building code, reusing existing code (a practice actively encouraged by the Scratch community), abstracting/modularizing code, and testing and debugging code.

The first three practices impact code structure in predictable ways and can therefore be detected by statically analyzing code as it evolves. This evolution can be visualized for instructors in a manner similar to the development of concepts (Fig. 3). In general, testing and debugging practices can impact code structure in ways that are far less predictable. Eguiluz et al. [23] avoid this concern by heavily restricting the set of programs that learners can create. Under these conditions, they can track the degree of discrepancy between each learner's solution and the optimal solution, and can reliably assess how effective each learner is in testing and debugging. The approach we propose does not restrict learners in any way, but it is less reliable.

Consider a classroom where every learner is assigned a common programming task  $T$ . In general, there may be infinitely many solutions for task  $T$ , but suppose that the instructor can specify a finite set  $R$  of “realistic” solutions to task  $T$ . For any given learner solution  $s$  to task  $T$ , we define how far  $s$  is from being correct as:

$$Distance(s, R) = \min_{r \in R} d(s, r) \quad (1)$$

Here, the function  $d(p, q)$  defines a suitable notion of the “distance” between two programs  $p$  and  $q$ . In the context of Scratch where programs can be represented as JSON strings, we can define the function  $d(p, q)$  as the Levenshtein distance between



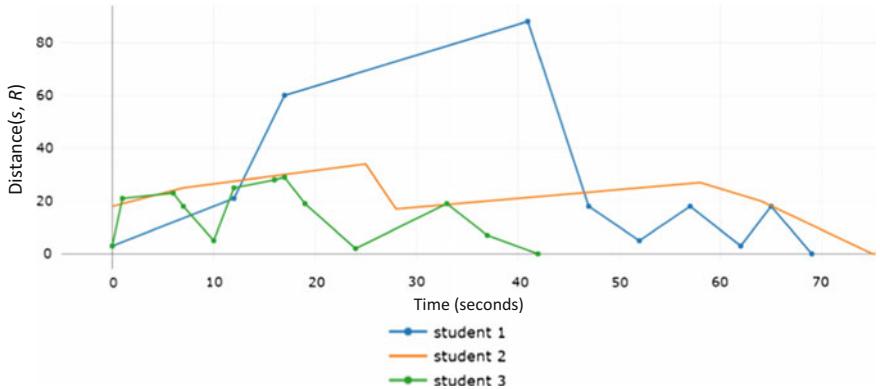
**Fig. 4** A plot of time (represented in seconds on the  $x$ -axis) versus  $Distance(s, R)$  (represented on the  $y$ -axis) for three students solving the same problem

the strings representing  $p$  and  $q$ , which is the fewest number of deletions, insertions, or substitutions required to convert one string to the other (also called the edit distance). When  $Distance(s, R)$  is high, it is likely that the learner is far from any correct solution. Note that if  $s$  matches some solution in  $R$ , then  $Distance(s, R)$  is 0. The instructor can visualize the data streams of learners by plotting  $Distance(s, R)$  against time, as shown in Fig. 4. Such a plot can indicate which learners are converging (slowly or rapidly) toward a realistic solution (i.e., programming effectively) and which learners appear to be struggling.

We conducted a small-scale study for a single task, where the set  $R$  consisted of a single solution (shown in Fig. 1). All learners were given a common (partial) solution to begin with, and Fig. 4 visualizes the data streams for three learners. Note that the times indicated on the  $x$ -axis are relative to the time stamp of the first event. Thus, both student 1 (blue trace) and student 3 (green trace) have nearly correct solutions at time zero (their solution distances, recorded on the  $y$ -axis, are nearly zero), whereas student 2 (orange trace) has a partial solution that is at a greater distance from the one solution in  $R$ .

Note that student 1 takes nearly 70 additional seconds to complete the task and appears to spend the last 20 s of this time vacillating significantly before finally reaching the solution. In contrast, student 3 completes the task in just over 40 s and in fact gets very close to the solution twice in the initial 10 s. The graph in Fig. 4 suggests that student 2 is far from achieving the desired solution. Instructors with access to such information in real-time could consider stepping into assist learners such as student 2 based on such traces.

We now consider the primary limitation of this approach. We have assumed that the instructor specifies the set  $R$  of all realistic solutions ahead of time. In practice, learners may come up with novel solutions that lie outside the set  $R$ . In this case, the value of  $Distance(s, R)$  may increase, even though the learner is converging toward a correct solution. In such a case, an instructor may incorrectly believe that the learner



**Fig. 5** The same data presented in Fig. 4, but with the set  $R$  containing both the instructor's solution and student 2's

is struggling and may consider offering assistance. We believe that in most situations of this kind, the instructor will quickly recognize that the learner's approach is in fact correct. Such an instance arose in the example presented above—the instructor found that student 2's solution was correct, but somewhat different in approach.

In such a situation, our visualizer can re-plot the data by adding novel solutions to the set  $R$ . A revised plot of the same data presented earlier is shown in Fig. 5.

Note that student 2's trace in Fig. 5 correctly shows the final distance as zero, indicating that student 2 achieved a correct solution (in just under 80 s). We can also observe that the sharp *increase* in distance between student 2's solution and the instructor's solution near the 27 s mark in Fig. 4 corresponds to a sharp *decrease* in distance in Fig. 5, as this learner shifts toward a different approach.

It is also instructive to compare the traces of students 1 and 3 in Fig. 4 and Fig. 5—note that student 1's trace is unchanged, whereas student 3's trace is very different after the initial 10 s.

To understand this difference, we introduce the following notation. Let  $s_2$  and  $s_3$  denote the solutions of student 2 and student 3, respectively, and let  $R$  and  $R'$  denote the sets of “realistic” solutions for the plots in Fig. 4 and Fig. 5, respectively. Note that  $R' = R \cup \{s_2\}$  (i.e.,  $R'$  is a superset of  $R$ ). Hence, from Eq. (1) it follows that for any solution  $s$ :

$$\text{Distance}(s, R') \leq \text{Distance}(s, R) \quad (2)$$

In other words, growing the set of realistic solutions cannot increase distances, but distances may decrease since the minimum is being computed over a superset. Since student 1's trace is unchanged, we conclude that at every intermediate point, student 1's partial solution was never closer to solution  $s_2$  than to the instructor's solution. In contrast, Fig. 5 clearly shows that student 3's solution was very similar to solution  $s_2$ .

after 23 s (a fact that cannot be ascertained from Fig. 4). Instead of completing this solution, student 3 rapidly modified solution  $s_3$  to match the instructor's solution.

## 4 Conclusions and Extensions

The key contributions of this paper are a mechanism to log transaction-level data generated by Scratch programmers and an instructor tool for visualizing this data to assess the development of individual learners' computational thinking abilities. Currently, we are evaluating the robustness of our logging mechanism by testing it in large classrooms.

There are multiple ways in which this work can be extended. To begin with, although Levenshtein distance can be calculated efficiently, the programs we expect to deal with are small. Since Scratch programs can be represented as labeled trees, we are investigating whether edit distances on trees [24], which are computationally more expensive, are nevertheless more meaningful in this context.

We also note that plotting distances over time may give instructors an added ability to recognize instances of unfair means (e.g., plagiarism). For instance, reconsider student 3's trace in Fig. 4. If the sudden drop in distance (at approximately 38 s into the trace) was observed without any instructor intervention, it would be consistent with the possibility that this learner obtained the correct answer from another source. In situations where such behavior is unacceptable, an automated tool could flag such instances for the instructor to carefully investigate manually.

Finally, we believe that automated tools can be developed to identify specific learner misconceptions and conceptual gaps in their understanding from low-level transaction-level data streams. To perform this task effectively, it may be necessary to combine low-level events in the data stream into a single event that corresponds to a high-level user action, as has been done for Alice programming [13].

## References

1. Wing, J. M. (2006). Computational Thinking. *CACM Viewpoint*, 33–35.
2. Ladner, R. E., & Israel, M. (2016). For all in computer science for all. *Communications of the ACM*, 59(9), 26–28.
3. Resnick, M., Maloney, J. H., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., et al. (2009). Scratch: Programming for all. *Communications of the ACM*, 52(11), 60–67.
4. Maloney, J. H., Peppler, K., Kafai, Y., Resnick, M., & Rusk, N. (2008). Programming by choice: Urban youth learning programming with Scratch. In *Proceedings of the 39th SIGCSE Technical Symposium on Computer Science Education* (pp. 367–371).
5. Scratch Statistics. (2017). <https://scratch.mit.edu/statistics>. Accessed 24 Dec 2017.
6. Siemens, G. (2012). Learning analytics: envisioning a research discipline and a domain of practice. In *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge*.

7. Siemens, G., & Baker, R. S. (2012). Learning analytics and educational data mining: towards communication and collaboration. In *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge*.
8. Chung, G. (2014). Toward the relational management of educational measurement data. *Teachers College Record*, 116(11).
9. Davies, R., Nyland, R., Chapman, J., & Allen, G. (2015). Using transaction-level data to diagnose knowledge gaps and misconceptions. In *Proceedings of the Fifth International Conference on Learning Analytics And Knowledge (LAK '15)* (pp. 113–117).
10. Tsui, A. (1996). Reticence and anxiety in second language learning. In *Voices from the language classroom* (pp. 145–167).
11. Karabenick, S. A., & Richard, N. S. (2013). *Help seeking in academic settings: Goals, groups, and contexts*. Routledge.
12. Basawapatna, A. R., Repenning, A., & Koh, K. H. (2015). Closing the cyberlearning loop: Enabling teachers to formatively assess student programming projects. In *Proceedings of the 46th ACM Technical Symposium on Computer Science Education (SIGCSE '15)* (pp. 12–17).
13. Werner, L., McDowell, C., & Denner, J. (2013). Middle school students using Alice: what can we learn from logging data? In *Proceeding of the 44th ACM Technical Symposium on Computer Science Education* (pp. 507–512).
14. Chao, P.-Y. (2016). Exploring students' computational practice, design and performance of problem-solving through a visual programming environment. *Computers & Education*, 95, 202–215.
15. Gal, L., Hershkovitz, A., Morán, A. E., Guenaga, M., & Garaizar, P. (2017). Suggesting a log-based creativity measurement for online programming learning environment. In *Proceedings of the 4th ACM Conference on Learning @ Scale (L@S '17)* (pp. 273–277).
16. Leidl, K., Bers, M. U., & Mihm, C. (2017). Programming with ScratchJr: A review of the first year of user analytics. In *The proceedings of the International Conference on Computational Thinking Education*.
17. Velasquez, N. F., Fields, D. A., Olsen, D., Martin, T., Shepherd, M. C., Strommer, A., & Kafai, Y. B. (2014). Novice programmers talking about projects: What automated text analysis reveals about online scratch users' comments. In *Proceedings of the 47th Hawaii International Conference on System Sciences*.
18. Boe, B., Hill, C., Len, M., Dreschler, G., Conrad, P., & Franklin, D. (2013). Hairball: Lint-inspired static analysis of scratch projects. In *Proceeding of the 44th ACM Technical Symposium on Computer Science Education (SIGCSE '13)* (pp. 215–220).
19. Moreno-León, J., Robles, G., & Román-González, M. (2015). Dr. Scratch: Automatic analysis of scratch projects to assess and foster computational thinking. *RED-Revista de Educación a Distancia*.
20. Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 Annual Meeting of the American Educational Research Association* (pp. 1–25).
21. Flask, A. (2017). Python Microframework. <http://flask.pocoo.org>. Accessed 24 Dec 2017.
22. School Education in India. (2017). Flash Statistics 2015–16. <http://udise.in/Downloads/Publications/Documents/U-DISE-SchoolEducationInIndia-2015-16.pdf>. Accessed 24 Dec 2017.
23. Eguiluz, A., Guenaga, M., Garaizar, P., & Olivares-Rodriguez, C. (2017). Exploring the progression of early programmers in a set of computational thinking challenges via clickstream analysis. *IEEE Transactions on Emerging Topics in Computing*, PP(99).
24. Pawlik, M., & Augsten, N. (2015). Efficient computation of the tree edit distance. *ACM Transactions on Database Systems*, 40(1).