

Evolutionary Computing applied to solve some Analytical and Operational problems in Banking, Financial Services and Insurance

A thesis submitted during 2020 to the University of Hyderabad in partial
fulfillment of the award of a Ph.D. degree in School of Computer and
Information Sciences

by

GUTHA JAYA KRISHNA



School of Computer and Information Sciences

University of Hyderabad
(P.O.) Central University, Gachibowli
Hyderabad - 500 046
Telangana
India



CERTIFICATE

This is to certify that the thesis entitled “**Evolutionary Computing applied to solve some Analytical and Operational problems in Banking, Financial Services and Insurance**” submitted by **GUTHA JAYA KRISHNA** bearing **Reg. No. 14MCPC14** in partial fulfillment of the requirements for award of Doctor of Philosophy in the School of Computer & Information Sciences is a bonafide work carried out by him under my supervision and guidance at IDRBT, Hyderabad.

This thesis is free from plagiarism and has not been submitted previously in part or in full to this or any other University or Institution for award of any degree or diploma.

Parts of this thesis have been published online in following publications (Please refer Appendix D for further details):

1. **Gutha Jaya Krishna** and Vadlamani Ravi. Evolutionary computing applied to customer relationship management: A survey. Engineering Applications of Artificial Intelligence, vol. 56, pages 30–59, 2016. ISSN:0952-1976 (Work reported in this journal article appears in Chapter 2)
2. **Gutha Jaya Krishna**, Vadlamani Ravi, and S. Nagesh Bhattu. Key Generation for Plain Text in Stream Cipher via Bi-Objective Evolutionary Computing. Applied Soft Computing, vol. 70, pages 301–317, 2018. ISSN: 1568-4946 (Work reported in this journal article appears in Chapter 10)
3. **Gutha Jaya Krishna** and Vadlamani Ravi. Evolutionary Computing Applied to Solve Some Operational Issues in Banks. In Optimization in Industry. Management and Industrial Engineering. pages 31–53. Springer, Cham, 2019. (Work reported in this book chapter appears in Chapter 2).
4. **Gutha Jaya Krishna** and Vadlamani Ravi. Modified harmony search applied to reliability optimization of complex systems. In Proceedings of the 2nd International Conference on Harmony Search Algorithm, vol. 382, pages 169–180, Seoul, South Korea, 2015. Springer Verlag. (Work presented in this international conference appears in Chapter 3).
5. **Gutha Jaya Krishna** and Vadlamani Ravi. Hybrid Evolutionary Algorithm for Optimizing Reliability of Complex Systems. In Proceedings of the International Conference on Intelligent Systems Design and Applications, vol. 941, pages 108–119, Vellore, India, 2018. Springer Verlag. (Work presented in this international conference appears in Chapter 3).

6. **Gutha Jaya Krishna** and Vadlamani Ravi. Feature Subset Selection using Adaptive Differential Evolution: An Application to Banking. In Proceedings of the ACM India Joint International Conference on Data Science and Management of Data, pages 157–163, Kolkata, India, 2019. ACM Press. (Work presented in this international conference appears in Chapter 5).
7. **Gutha Jaya Krishna** and Vadlamani Ravi. Outlier Detection using Evolutionary Computing. In Proceedings of the International Conference on Informatics and Analytics - ICIA-16, pages 17:1–17:6, Pudicherry, India, 2016. ACM Press. (Work presented in this international conference appears in Chapter 8),
8. **Gutha Jaya Krishna** and Vadlamani Ravi. Keystroke based User Authentication using Modified Differential Evolution. In Proceedings of the 2019 IEEE Region 10 Conference (TENCON), pages 739–744, Kochi, Kerala, India, 2019. IEEE. (Work presented in this international conference appears in Chapter 8).
9. **Gutha Jaya Krishna**, Meesala Smruthi, Vadlamani Ravi, and Bhamidipati Shan-dilya. Differential Evolution Trained Fuzzy Cognitive Map: An Application to Modeling Efficiency in Banking. In Proceedings of the International Conference on Intelligent Systems Design and Applications, vol. 941, pages 1–11, Vellore, India, 2018. Springer Verlag. (Work presented in this international conference appears in Chapter 9).

Further, the student has passed the following courses towards fulfillment of coursework requirement for Ph.D:

	Course Code	Name	Credits	Pass/Fail
1	BT701	Data Structures and Algorithms	4	Pass
2	BT702	Operating System and Programming	4	Pass
3	BT713	Global Optimization and Evolutionary Computing	4	Pass
4	BT717	Advanced Data Mining	4	Pass

Supervisor
Prof. Vadlamani Ravi
Center of Excellence in
Analytics, IDRBT
Hyderabad-500 057, India

Director
Dr. A.S. Ramasastri
IDRBT
Hyderabad-500 057, India

Dean
Prof. Chakravarthy Bhagvati
School of Computer and
Information Sciences, UOH
Hyderabad-500 046, India

DECLARATION

I, **GUTHA JAYA KRISHNA**, hereby declare that this thesis entitled “**Evolutionary Computing applied to solve some Analytical and Operational problems in Banking, Financial Services and Insurance**” submitted by me under the guidance and supervision of **Prof. Vadlamani Ravi**, is a bonafide research work and is free from plagiarism. I also declare that it has not been submitted previously in part or in full to this University or any other University or Institution for the award of any degree or diploma. I hereby agree that my thesis can be deposited in Shodganga/INFLIBNET.

A report on plagiarism statistics from the University Librarian is enclosed.

Date:

Signature of the Student

(GUTHA JAYA KRISHNA)

Reg. No.: 14MCPC14

//Countersigned//

Signature of the Supervisor:

*Dedicated to The God,
My Family & Teachers*

Acknowledgements

First and above of all, I praise **God**. I am very much grateful to him for his incredible love towards me, the gift of everlasting life and the grace of amazing energy to work hard.

Later, I would like to thank my parents **Sri. Gutha Krishnaiah Naidu** and **Smt. Gutha Satyavathi** for their best support and encouragement through all times in pursuing my dreams. I would also like to thank my elder brother (Dr. Gutha Naveen Kumar), and brother's daughter (Lekhana) for their uncountable love.

I specially acknowledge **University of Hyderabad (UoH)** for its academic support. I also extend my acknowledgments to **Institute for Development and Research in Banking Technology (IDRBT)**, a R&D center established by RBI, for its kind financial support and beautiful environment for conducting the research. I express my sincere thanks to **Center of Excellence in Analytics (CoEA)**, IDRBT for making me part of experimenting with datasets of various banks and presenting the results to the bankers in my Ph.D.

This thesis would not have been possible without the help and support of the kind people around me. Among them, first of all, I owe my greatest gratitude to my supervisor **Prof. Vadlamani Ravi**, Professor, IDRBT, Hyderabad for his precious foundational & advanced lectures of Data Mining and related topics, strict supervision, regular encouragement and timely support from the introductory level to the concluding level that enabled me to understand and implement the concepts learned. He guided me in each stage of my research including problem definition, idea creation, problem solution and in paper writing. He gave me a free hand and allowed me to think freely throughout my doctoral program.

It is my privilege to thank **Prof. Arun Agrawal**, Pro Vice- Chancellor of UoH, Hyderabad and **Prof. K. Narayana Murthy**, Former Dean of School of Computer and Information Sciences (SCIS), UoH, Hyderabad for their academic support. Its also my privilege to thank the **Prof. Chakravarthy Bhagvati**, Dean, School of Computer and Information Sciences (SCIS), UoH, Hyderabad for his academic support. It is an honor for me to thank **Dr. A. S. Ramasastry**, Director, IDRBT for the research facilities and support provided to me throughout the span of my Ph.D. for conducting excellent research. I also thank **Mr. B. Sambamurthy**, Former Director, IDRBT for extending his cooperation at the preliminary stage of my research. I would like to extend my sincere thanks to **Prof. B. M. Mehtre**, Professor, IDRBT, **Dr. M. V. N. K. Prasad**, Associate Professor, IDRBT, and **Dr. G. R. Gangadharan**, Associate Professor, IDRBT for their regular reviews of the progress of research and kind inputs as the members of my Doctoral Research Committee. I also take a minute to thank **Prof. B. L. Deekshatulu**, **Prof. B. Yegnanarayana**, **Prof. D. K. Subramanyam**, **Prof. P. Ananth Raj** and **Prof. Venu Govindaraju** for their periodical reviews of my work and their kind inputs.

I would also like to express my sincere thanks and gratitude to the faculty at IDRBT, including Prof. V. N. Sastry, Dr. V. Radha, Dr. S. Nagesh Bhattu, Dr. Rajarshi Pal and Dr. N. V. Narendra Kumar for their timely suggestions in the the areas of their expertise. I am also thankful to Mr. G. Raghuraj for his support. I am always grateful to the supporting staff of IDRBT including Mr. S. Rashmi Dev, Mrs. Varsha Srivatsava, Mr. Vijay Belgurikar and Mr. P. Balasubramanyam and Ms. T. Preethi for their support in attending conferences, Late Mr. P. Ratna Kumar, Mr. Prakash Dhavale and Mr. Srinivas for their support in utilizing library resources to conduct research, Mr. K. V. R. Murthy for his technical support, Mr. K. Srinivas, Mrs. Pavani, Mr. N. Namdev, Mrs. N. Sony, Mrs. A. Madhuri and Ms. Tejaswi Swarna Sagarika, Mr. D. Chaitanya for administrative support and finally, the staff of Prism Catering Services for their continuous support of serving refreshments throughout my doctoral study.

I would also like to thank all Doctoral students, Dr. D. Pradeep Kumar, Dr. B. Shravan Kumar, Dr. Kumar Ravi, Mr. Sk. Kamaruddin and Mr. Amit Soni Arya; M.Tech students, Research Associates (RA) and Project Interns at CoEA, IDRBT during my Ph.D. research work. I also would like to thank Project Interns, Mr. Raja Kumar Dubey and Miss Smruthi Meesala, for their association with me during my Ph.D. research work. I would also like to thank the RAs: Mr. P. Sai Ravi Teja, Mr. Harshal Jaiswal, Mr. B. Vikram Reddy, Mr. Md. Zaheeruddin, for their collaboration with me. I would also like to thank Mr. Bhamidipati Shandilya Sir, senior domain expert in the field of banking, for his collaboration with me.

Gutha Jaya Krishna

Abstract

Performance, profitability, and risk-reduction would be the main goals that the banking, financial services and insurance (BFSI) sectors attempt to achieve. In this data-driven world, performance, profitability, and risk-reduction are influenced by the analytical and operational technologies which could give better insights along with return on the investment for the banking, financial and insurance sectors. The focus of these technologies is to provide efficient models, enhance customer service, and to protect their systems against cyber threats.

While evolutionary computing (EC) techniques made steady inroads into the analytical and operational areas, and it has not done as much as it has in other fields of application. Our work demonstrates that EC techniques applied to solve the analytical and operational problems which still offers a fertile ground, where EC techniques, either in isolation or in combination, have a tremendous role to play.

Traditionally, data mining (DM) techniques have already been applied to resolve various analytical problems within BFSI sector. In turn, optimization techniques have traditionally been employed for training several of the DM techniques. However, in the past several years, evolutionary techniques have grown to be so powerful and versatile that they can be deployed as an alternative for many DM techniques. This trend caught the interest associated with the researchers involved in the analytical area because they too started solving the analytical problems using evolutionary techniques alone. Numerous DM techniques will need optimization at various stages, such as for instance feature selection, training a classifier/regression model or determining the suitable user-defined parameter combination that will result bring about accurate outcomes for the technique. Having said that,

evolutionary techniques can be employed in standalone mode to deal with various data mining tasks such as classification, regression, clustering, association rule mining, and outlier detection, alongside subtasks such as feature selection. Given this established versatility of EC techniques, it's possible to safely imagine that EC techniques may be employed to resolve a few of the analytical problems of BFSI sector.

Analytical problems of BFSI sectors are listed below, where EC techniques have an application are: 1) Customer Segmentation, 2) Customer Lifetime Value, 3) Market Basket Analysis, 4) Credit Scoring, 5) Fraud Detection, 6) Churn Detection, 7) Default detection, etc. The above listed analytical problems in BFSI sector were earlier solved by employing traditional data mining techniques. We in our work are using EC techniques alone to solve them. BFSI sector faces many problems in the operational front while servicing the customers, where evolutionary computing may become handy. Whenever we consider operational problems, conventional techniques usually do not yield comparatively better results compared to the evolutionary-based techniques. A few of the operational problems associated with the BFSI sector such as, for instance Security and Efficiency Prediction, the banking, financial and insurance industries are looking beyond conventional ways toward more automated and much more robust methods. EC based techniques play an important role in solving the above-mentioned operational problems simply because they yield global or near-global optimal results.

Depending on the necessity we employed a few of the evolutionary algorithms within the literature, the variants of these evolutionary algorithms, a novel memetic algorithm, also an adaptive type of evolutionary algorithm and a multi-objective evolutionary algorithm to solve the problems we considered within the thesis.

Following analytical problems of BFSI sector are solved using EC techniques alone. We addressed the data mining subtask of feature selection using evolutionary computing where important features for credit scoring and detecting financial statement fraud are found by EC-based feature subset

selection techniques. Rule based classification is also performed, addressing the data mining task of classification using evolutionary computing on all the features of credit scoring and financial statement fraud data, as well as, on the feature subsets from above wrapper-based feature subset selection. Customer segmentation based on monetary value is performed by High Utility Itemset Mining and High Utility Association Rule Mining addressing the data mining task of Market Basket Analysis using evolutionary computing. Lastly, anomaly detection is performed addressing the data mining tasks of clustering and outlier detection using evolutionary computing for keystroke detection, churn, default and insurance fraud detection by evolutionary computing-based subspace outlier detection.

We addressed the data mining task of regression along with the operational problem of predicting the efficiency of banks using evolutionary computing where efficiency prediction of banks is performed with a Differential Evolution trained Fuzzy Cognitive Map. Lastly, we addressed the operational problem of security using evolutionary computing by generating keys for plain text in stream cipher via bi-objective evolutionary computing

Contents

Abstract	i
List of Figures	xiv
List of Tables	xviii
List of Algorithms	xxiv
Acronyms	xxv
1 Introduction	1
1.1 Introduction	1
1.2 Motivation	2
1.3 Problem Statement	5
1.3.1 Tested Evolutionary Computing Techniques	5
1.3.2 Evolutionary Computing techniques applied to solve some Analytical problems	6
1.3.3 Evolutionary Computing techniques applied to solve some Operational problems	6
1.4 Contributions in the Thesis	7
1.5 Organisation of the thesis	9
2 Systematic Review of the Related Work	16
2.1 Introduction	16
2.1.1 Optimization Techniques and Data Mining (DM)	16
2.1.2 Analytical and Operational Problems in BFSI Sector	17
2.1.2.1 Customer Segmentation	17

2.1.2.2	Direct Marketing, Customer Campaigning and Service Marketing	18
2.1.2.3	Customer Lifetime Value (CLTV)	18
2.1.2.4	Credit Scoring	18
2.1.2.5	Market Basket Analysis (MBA)	18
2.1.2.6	Fraud Detection	19
2.1.2.7	Default Detection	19
2.1.2.8	Customer Churn Detection	19
2.1.2.9	Key Generation for Security	20
2.1.2.10	Bank Efficiency Prediction	20
2.2	Theme of the proposed work	21
2.2.1	Evolutionary Computation applied for solving Analytical Problems of BFSI sector	21
2.2.2	Evolutionary Computation applied for solving Operational Problems of BFSI sector	23
2.3	Review Methodology	23
2.3.1	Techniques relevant to the review	23
2.3.2	Flowchart of Review Process	25
2.4	Distribution of Reviewed Articles	25
2.4.1	Count of Research Articles Reviewed by various Journals / Books / Proceedings	25
2.4.2	Distribution of Research Articles by Analytical and Operational Problems	25
2.4.3	Distribution of Research Articles by EC Technique	25
2.4.4	Datasets used in reviewed articles	37
2.4.5	Trend	38
2.5	Review of Works in the Current Study	38
2.5.1	Credit Scoring	40
2.5.2	Customer Segmentation	44
2.5.3	Fraud Detection	44
2.5.4	Market Basket Analysis	46
2.5.5	Customer Life Time Value	48
2.5.6	Customer Churn Prediction	48

2.5.7	Default Prediction	49
2.5.8	Key Generation	49
2.6	Discussion of the Review and Future Directions	69
2.7	Conclusions	72
3	A Hybrid and the variants of Differential Evolution and Harmony Search	73
3.1	Introduction	73
3.2	Motivation	74
3.3	Related Work	74
3.4	Improved Modified Harmony Search + Modified Differential Evolution .	75
3.4.1	Improved Modified Harmony Search	75
3.4.2	Modified Differential Evolution	77
3.4.3	Improved Modified Harmony Search + Modified Differential Evolution	78
3.5	Constraint Handling	79
3.5.1	Static Extinctive Penalty	79
3.6	Problem Definitions	80
3.7	Results and Discussion	81
3.8	Conclusions and Future Directions	83
4	Large Scale Global Optimization via Improved Modified Harmony Search and Modified Differential Evolution Hybrid	88
4.1	Introduction	88
4.2	Related Work	89
4.3	CEC 2013 LSGO Benchmarks	91
4.4	Proposed Hybrid Optimization Algorithm	94
4.4.1	Nature of Optimization Problem	94
4.4.2	Proposed Improved Modified Harmony Search + Modified Differential Evolution	94
4.4.3	Time complexity of the proposed hybrid	103
4.4.4	Parameter tuning for the proposed hybrid	103
4.4.4.1	Parameter List and Tuning Methodology	104
4.5	Results Analysis	107

4.6	Conclusions and Future Directions	113
5	Feature Subset Selection using Adaptive Differential Evolution: An application to banking	114
5.1	Introduction	114
5.2	Motivation	115
5.3	Related Work	116
5.3.1	EC-based FSS using classification techniques	116
5.3.2	Credit Scoring and Financial Statement Fraud	117
5.4	Classification Algorithms and Evaluation Measures	117
5.4.1	Classifiers Employed for the proposed FSS	117
5.4.1.1	Probabilistic Neural Network	117
5.4.1.2	Logistic Regression	117
5.4.1.3	Naive Bayes	118
5.4.1.4	Support Vector Machine	118
5.4.2	Evaluation Measures used in the Proposed FSS	119
5.4.2.1	Sensitivity	119
5.4.2.2	Specificity	119
5.4.2.3	Accuracy	119
5.4.2.4	Area Under Curve	119
5.4.2.5	Matthews Correlation Coefficient	119
5.5	Evolutionary Computing	120
5.5.1	Differential Evolution	120
5.5.2	Adaptive Differential Evolution	120
5.6	Proposed ADE-based Feature Subset Selection	121
5.7	Dataset Description	123
5.8	Results and Discussion	124
5.9	Conclusions and Future Directions	128
6	Rule-based Classification via Differential Evolution for the banking applications	129
6.1	Introduction	129
6.2	Motivation	131
6.3	Contribution	131

6.4	Related Work	132
6.5	Differential Evolution (DE) and Adaptive Differential Evolution (ADE)	135
6.5.1	Differential Evolution	135
6.5.2	Adaptive Differential Evolution	136
6.6	Proposed Methodology	137
6.6.1	Rule Encoding	137
6.6.2	DE and ADE based Rule-based Classification	137
6.6.3	Initialisation	140
6.6.4	Mutation	140
6.6.4.1	Mutation for DE	140
6.6.4.2	Mutation for ADE	141
6.6.5	Recombination	141
6.6.5.1	Recombination in DE	141
6.6.5.2	Recombination in ADE	141
6.6.6	Discretisation and Swapping	142
6.6.7	Fitness Evaluation	142
6.6.8	Selection	143
6.7	Dataset Description and Experimental Setup	143
6.8	Results and Discussion	145
6.9	Conclusions and Future Directions	161
7	High Utility Itemset and Association Rule Mining using Binary Differential Evolution: An application to Customer Segmentation	162
7.1	Introduction	162
7.2	Motivation	164
7.3	Contribution	165
7.4	Related Work	165
7.4.1	Evolutionary Computing applied to Data Mining	165
7.4.2	High Utility based Itemset Mining	166
7.4.3	Evolutionary Computing based High Utility Itemset Mining and High Utility Association Rule Mining	166
7.5	High Utility Itemset Mining (HUIM) and High Utility Association Rule Mining (HUARM)	167

7.5.1	Definitions of HUIM and HUARM	168
7.5.2	HUIs Mined for the Example	170
7.5.3	HUARMs mined for the Example	170
7.6	Binary Differential Evolution (BDE) and Adaptive Binary Differential Evolution (ABDE)	172
7.6.1	Differential Evolution	172
7.6.2	Adaptive Differential Evolution	172
7.6.3	Binary Differential Evolution and Adaptive Binary Differential Evolution	173
7.7	Proposed High Utility Itemset Miners	174
7.7.1	BDE and ABDE based HUIM	174
7.7.1.1	Initialization	174
7.7.1.2	Mutation	175
7.7.1.3	Recombination	176
7.7.1.4	Discretization	176
7.7.1.5	Fitness Evaluation	177
7.7.1.6	Selection	177
7.8	Proposed High Utility Association Rule Miners	180
7.8.1	BDE and ABDE based HUARM	181
7.8.1.1	Initialization	181
7.8.1.2	Mutation	181
7.8.1.3	Recombination	182
7.8.1.4	Discretization	183
7.8.1.5	Fitness Evaluation	183
7.8.1.6	Selection	183
7.9	Results and Discussion	187
7.9.1	Results of HUIM	188
7.9.1.1	Number of Itemsets Mined at various Thresholds	188
7.9.1.2	Convergence of proposed HUIMs with extant methods .	189
7.9.1.3	HUIM-based Customer Segmentation based on Value .	197
7.9.1.4	Optima Obtained and Statistics by proposed HUIM al- gorithms	200

7.9.1.5	Memory and Time Consumed by the Proposed HUIM Algorithms	203
7.9.2	Results of HUARM	207
7.9.2.1	Number of Association Rules Mined at various Thresholds	207
7.9.2.2	Convergence of proposed HUARM algorithms	212
7.9.2.3	HUARM-based Customer Segmentation based on Value	215
7.9.2.4	Optima Obtained and Statistics of proposed HUARM algorithms	217
7.9.2.5	Memory and Time Consumed by the Proposed HUARM Algorithms	221
7.9.3	Control Parameters of the BDE and ABDE	221
7.10	Conclusions and Future Directions	222
8	Subspace based Anomaly Detection via Harmony Search and Differential Evolution: An application to Analytical CRM and Security	223
8.1	Introduction	223
8.2	Related Work	224
8.2.1	Keystroke-based User Authentication	225
8.2.2	Clustering-based Anomaly detection for CRM	227
8.3	Optimization Problem Formulation	227
8.4	Proposed approach for anomaly detection in keystroke data	229
8.4.1	Euclidean-based Method	229
8.4.2	Manhattan-based Method	229
8.4.3	Mahalanobis-based Method	230
8.4.4	Harmony Search	231
8.4.5	Improved Modified Harmony Search	232
8.4.6	Differential Evolution	233
8.4.7	Modified Differential Evolution	235
8.4.8	Improved Modified Harmony Search + Modified Differential Evolution	237
8.5	Proposed approach for anomaly detection in banking and insurance data	237
8.5.1	Modified Differential Evolution with selection constraint	237
8.6	Description of Datasets	240

8.6.1	Keystroke data	240
8.6.2	Churn Dataset	240
8.6.3	Insurance Fraud Dataset	240
8.6.4	Default Dataset	241
8.7	Results and Discussion	241
8.7.1	Results of keystroke-based user authentication	241
8.7.1.1	AUC and EER	245
8.7.2	Results of banking and insurance data	246
8.8	Conclusions and Future Directions	248
9	Differential Evolution trained Fuzzy Cognitive Map: An application to modelling efficiency in banking	249
9.1	Introduction	249
9.2	Motivation	250
9.3	Contribution	251
9.4	Related work	251
9.5	Proposed Methodology	252
9.5.1	Activation Function	252
9.5.2	Mean Absolute Percentage Error	253
9.5.3	Root Mean Squared Error	253
9.5.4	DE trained FCM	253
9.5.5	The FCM with Hebbian Training Mode	253
9.5.6	Sequential and Batch modes of Training	253
9.6	Description of the Dataset	256
9.7	Experimental Setup	256
9.8	Results and Discussion	257
9.9	Conclusions and Future Directions	260
10	NSGA-II application in Security: A case of key generation for plain text in stream cipher	261
10.1	Introduction	261
10.1.1	Symmetric-key Cryptography	261
10.1.2	Random Key Generation	262
10.1.3	Codes and Ciphers	264

10.2	Related Work in Cryptography	264
10.3	Problem Definition	265
10.4	Motivation	265
10.5	Contribution	265
10.6	Proposed Evolutionary Computing based Key Generation	267
10.6.1	Objective Functions	267
10.6.2	Proof of Maximizing Entropy ensures Uniformly Distributed Keys	268
10.6.3	Threshold for the Fitness Function	269
10.6.4	Combinatorial Explosion of the Problem	270
10.6.5	Key generation using EC techniques	270
10.6.5.1	IMHS - Key Generation Algorithm	270
10.6.5.2	DE - Key Generation Algorithm	271
10.6.5.3	IMHS+DE - Key Generation Algorithm	273
10.6.5.4	NSGA-II - Key Generation Algorithm	274
10.7	Mutated Huffman Code Table	275
10.7.1	Huffman Tree Coding	275
10.7.2	Mutated Huffman Tree (MHT) Coding	275
10.8	Key Streams Generation	275
10.9	Procedure and Case Studies	276
10.9.1	Example Procedure	276
10.9.2	Case Study	276
10.10	Experimental Results	280
10.10.1	Encryption Times Compared	286
10.10.2	Compression Ratio	290
10.10.3	Empirical Attainment Function of the Bi-Objective Optimization	291
10.10.4	Parameter Settings	292
10.10.5	Time Complexity of the Key Generation Algorithms, MHT cod- ing, and Encryption	293
10.11	Security Analysis and Comparison	294
10.11.1	Key Space	294
10.11.2	Classical Attacks	295
10.11.3	Test for Cryptographically Secure PRNG	296
10.11.4	Analysis of the proposed KGAs with standard Stream Ciphers .	297

10.12	Conclusions and Future Directions	298
11	Conclusions and directions for future research	300
11.1	Summary and conclusions	300
11.2	Future Directions	303
	References	305
	List of Journal Publications	337
	List of Book Chapter Publications	337
	List of Conference Publications	338
	Under Review	339
A	Overview of Evolutionary Computing Techniques	340
B	State-of-the-Art Evolutionary Computing Techniques Employed	347
B.1	Harmony Search	347
B.2	Differential Evolution	348
B.3	Non-dominated Sorting Genetic Algorithm-II (NSGA-II)	350
C	Additional Supporting Results of IMHS+MDE on CEC 2013 LSGO Benchmark (Chapter 4)	354
D	Annexure (Fact Sheet of the Publications Online)	360

List of Figures

1.1	Overarching Diagram for the Ph.D. Thesis	8
2.1	Relationship between Analytical problems, Data Mining, and Optimization (EC) Techniques	22
2.2	Operational Problems Considered	23
2.3	Structure of Considered Techniques	24
2.4	Flowchart of Review Methodology	26
2.5	Analytical and operational problems using EC	28
2.6	Bar Chart based on Count of Analytical and Operational problems . . .	29
2.7	Bar Chart based on Count of all EC Techniques	35
2.8	Use of all EC Techniques	36
2.9	Trend of articles published per year	39
3.1	The Process of coupling between IMHS memory and MDE population vectors	80
3.2	Complex System of Life Support System in a space capsule	81
3.3	Reliability Complex Bridge System	82
4.1	The Process of coupling between IMHS memory and MDE population vectors	95
4.2	IMHS Process Flow	96
4.3	MDE Process Flow	97
4.4	Box Plot of the non-adaptive parameters	105
5.1	Adaptive Differential Evolution based FSS	123

6.1	Rule Encoding for the Antecedent	137
6.2	Antecedent part of the rule expanded	137
6.3	Flow Diagram of DE and ADE based Rule Miners	140
7.1	Solution Encoding Scheme for HUIM	174
7.2	BDE and ABDE-based HUIM Algorithms Process Flow	175
7.3	Before and after binary discretization of the high utility encoded association rule	180
7.4	BDE and ABDE-based HUARM Algorithms Process Flow	181
7.5	Convergence of <i>Chess</i> Dataset (HUIM)	191
7.6	Convergence of <i>Mushroom</i> Dataset (HUIM)	192
7.7	Convergence of <i>Connect</i> Dataset (HUIM)	193
7.8	Convergence of <i>Accidents</i> Dataset (HUIM)	194
7.9	Convergence of <i>Foodmart</i> Dataset (HUIM)	195
7.10	Convergence of <i>Retail</i> Dataset (HUIM)	196
7.11	No. of Itemsets Mined (<i>OnlineRetail</i> Dataset)	199
7.12	Convergence of <i>OnlineRetail</i> Dataset (HUIM)	200
7.13	Threshold vs Time (in Secs) graph of <i>Chess</i> Dataset	204
7.14	Threshold vs Time (in Secs) graph of <i>Mushroom</i> Dataset	204
7.15	Threshold vs Time (in Secs) graph of <i>Connect</i> Dataset	205
7.16	Threshold vs Time (in Secs) graph of <i>Accidents</i> Dataset	205
7.17	Threshold vs Time (in Secs) graph of <i>Foodmart</i> Dataset	206
7.18	Threshold vs Time (in Secs) graph of <i>Retail</i> Dataset	206
7.19	Threshold vs Time (in Secs) graph of <i>OnlineRetail</i> Dataset	207
7.20	No. of Association Rules Mined <i>Chess</i> Dataset	209
7.21	No. of Association Rules Mined <i>Mushroom</i> Dataset	209
7.22	No. of Association Rules Mined <i>Connect</i> Dataset	210
7.23	No. of Association Rules Mined <i>Accidents</i> Dataset	210
7.24	No. of Association Rules Mined <i>Foodmart</i> Dataset	211
7.25	No. of Association Rules Mined <i>Retail</i> Dataset	211
7.26	Convergence of <i>Chess</i> Dataset (HUARM)	212
7.27	Convergence of <i>Mushroom</i> Dataset (HUARM)	213
7.28	Convergence of <i>Connect</i> Dataset (HUARM)	213

7.29	Convergence of <i>Accidents</i> Dataset (HUARM)	214
7.30	Convergence of <i>Foodmart</i> Dataset (HUARM)	214
7.31	Convergence of <i>Retail</i> Dataset (HUARM)	215
7.32	No. of Association Rules Mined (<i>OnlineRetail</i> Dataset)	218
7.33	Convergence of <i>OnlineRetail</i> Dataset (HUARM)	218
8.1	Proposed approach for detecting anomalies in keystroke data	230
8.2	Flowchart for HS-based Subspace Anomaly Detection	231
8.3	Flowchart for IMHS-based Subspace Anomaly Detection	232
8.4	Flowchart for DE-based Subspace Anomaly Detection	234
8.5	Flowchart for MDE-based Subspace Anomaly Detection	236
8.6	Proposed approach for detecting anomalies in banking and insurance data	238
8.7	Flowchart for MDE-based Subspace Anomaly Detection with selection constraint	239
9.1	The Fuzzy Cognitive Model for the bank's efficiency data	259
10.1	Block Diagram of Symmetric-key Cryptography	262
10.2	Block Diagram of Stream Cipher	263
10.3	Block Diagram of the Process	266
10.4	Block Diagram for the Threshold	269
10.5	Flowchart for IMHS-Key Generation Algorithm	271
10.6	Flowchart for DE-Key Generation Algorithm	272
10.7	Flowchart for NSGA-II-Key Generation Algorithm	274
10.8	Huffman Tree for the "goodperson1199" and Mutated Huffman Tree mutated at nodes 6 and 7	277
10.9	Sample Text Snippet	278
10.10	Plain Text and Unique ASCII Characters	278
10.11	Entropy values generated by OBJ_{BO} vs computed from the OBJ_{SO} (1.80 GHz)	283
10.12	Entropy values generated by OBJ_{BO} vs computed from the OBJ_{SO} (860 MHz)	283
10.13	EAF of the Bi-objective Optimization using NSGA-II	292
B.1	Non-dominated Sorted Genetic Algorithm-II (NSGA-II)	351

B.2	Steps for Non-dominated Sorting	352
B.3	Steps for Crowding Distance	352
B.4	Steps for Non-dominated Sorted Gentic Algorithm-II (NSGA-II)	353
D.1	Screenshot of Engineering Applications of Artificial Intelligence Publi- cation [168]	363
D.2	Screenshot of Applied Soft Computing Publication [174]	364
D.3	Screenshot of Optimization in Industry Book Chapter Publication [171]	364
D.4	Screenshot of ICHSA-2015 Publication [167]	365
D.5	Screenshot of ICIA-2016 Publication [169]	365
D.6	Screenshot of ISDA-2018 Publication [170]	366
D.7	Screenshot of ISDA-2018 Publication [175]	366
D.8	Screenshot of CoDS COMAD-2019 publication [172]	367
D.9	Screenshot of IEEE TENCON 2019 publication [173]	367

List of Tables

2.1	Count of Surveyed Papers	27
2.2	Count based on analytical and operational problems	29
2.3	Research Articles based on analytical and operational problems	30
2.4	Count based on Single Objective EC Techniques	31
2.5	Count based on Multi-Objective EC Techniques	32
2.6	Count based on Hybrid EC Techniques	33
2.7	Use of EC Techniques in Single Objective framework	33
2.8	Use of Multi-Objective EC Techniques	34
2.9	Use of Hybrid EC Techniques	34
2.10	Datasets analyzed in articles	37
2.10	Datasets analyzed in articles (Continued)	38
2.11	Results of Classification Problems	51
2.11	Results of Classification Problems (Continued)	52
2.11	Results of Classification Problems (Continued)	53
2.11	Results of Classification Problems (Continued)	54
2.11	Results of Classification Problems (Continued)	55
2.11	Results of Classification Problems (Continued)	56
2.11	Results of Classification Problems (Continued)	57
2.11	Results of Classification Problems (Continued)	58
2.12	Results of Association Rule Mining (M.B.A.)	60
2.12	Results of Association Rule Mining (M.B.A.) (Continued)	61
2.13	Objective Functions, Properties of objective function and Parameters set for the EC techniques	62

2.13	Objective Functions, Properties of objective function and Parameters set for the EC techniques (Continued)	63
2.13	Objective Functions, Properties of objective function and Parameters set for the EC techniques (Continued)	64
2.13	Objective Functions, Properties of objective function and Parameters set for the EC techniques (Continued)	65
2.13	Objective Functions, Properties of objective function and Parameters set for the EC techniques (Continued)	66
2.13	Objective Functions, Properties of objective function and Parameters set for the EC techniques (Continued)	67
2.13	Objective Functions, Properties of objective function and Parameters set for the EC techniques (Continued)	68
3.1	Summary of the Results: Problem-1 Case (i)	84
3.2	Summary of the Results: Problem-1 Case (ii)	85
3.3	Summary of the Results: Problem-2	86
3.4	Statistics of the employed optimizations algorithms	87
4.1	Summary of functions used in CEC 2013 LSGO competition	93
4.2	Parameters for the IMHS+MDE	106
4.3	Specialist Method of Parameter tuning for PAR, CR, and F	106
4.4	Number of best values and ranking based on statistical tests	109
4.5	Ranking based on Medians (As performed in CEC 2015 LSGO) for benchmarks F1 to F8	110
4.6	Ranking based on Medians (As performed in CEC 2015 LSGO) for benchmarks F9 to F15	110
4.7	F1 racing points table for score calculation	111
4.8	F1 Score based on Median Ranking (As given in CEC 2015 LSGO) . . .	112
5.1	Results of German Credit Scoring Dataset	125
5.2	Results of Australian Credit Scoring Dataset	125
5.3	Results of Financial Statement Fraud Dataset	126
5.4	t-Test Results of German Credit Scoring Dataset	126
5.5	t-Test Results of Australian Credit Scoring Dataset	126

5.6	t-Test Results of Financial Statement Fraud Dataset	127
6.1	Results of DE and ADE Rule Miners on 20 % Validation Set of Australian Credit Scoring Dataset	147
6.2	Results of DE and ADE Rule Miners on 20 % Validation Set of German Credit Scoring Dataset	147
6.3	Results of DE and ADE Rule Miners on 20 % Validation Set of Financial Statement Fraud Dataset	148
6.4	Results of DE and ADE Rule Miners on 20 % Validation Set of Australian Credit Scoring Dataset with Feature Subsets from [172]	149
6.5	Results of DE and ADE Rule Miners on 20 % Validation Set of German Credit Scoring Dataset with Feature Subsets from [172]	149
6.6	Results of DE and ADE Rule Miners on 20 % Validation Set of Financial Statement Fraud Dataset with Feature Subsets from [172]	150
6.7	t-Test results of the DE vs ADE	151
6.8	t-Test results of the DE vs ADE with feature subsets from [172]	151
6.9	Rules obtained by DE, ADE Rule Miners and CHAID on Australian Credit Scoring Dataset	152
6.9	Rules obtained by DE, ADE Rule Miners and CHAID on Australian Credit Scoring Dataset (Continued)	153
6.10	Rules obtained by DE, ADE Rule Miners and CHAID on German Credit Scoring Dataset	153
6.10	Rules obtained by DE, ADE Rule Miners and CHAID on German Credit Scoring Dataset (Continued)	154
6.10	Rules obtained by DE, ADE Rule Miners and CHAID on German Credit Scoring Dataset (Continued)	155
6.10	Rules obtained by DE, ADE Rule Miners and CHAID on German Credit Scoring Dataset (Continued)	156
6.10	Rules obtained by DE, ADE Rule Miners and CHAID on German Credit Scoring Dataset (Continued)	157
6.10	Rules obtained by DE, ADE Rule Miners and CHAID on German Credit Scoring Dataset (Continued)	158

6.11	Rules obtained by DE, ADE Rule Miners and CHAID on Financial State- ment Fraud Dataset	158
6.11	Rules obtained by DE, ADE Rule Miners and CHAID on Financial State- ment Fraud Dataset (Continued)	159
6.11	Rules obtained by DE, ADE Rule Miners and CHAID on Financial State- ment Fraud Dataset (Continued)	160
6.11	Rules obtained by DE, ADE Rule Miners and CHAID on Financial State- ment Fraud Dataset (Continued)	161
7.1	Transactional Database with Quantities	168
7.2	Itemsets with a threshold of 25 on Utility	171
7.3	Datasets	188
7.4	Thresholds on Utility	188
7.5	Thresholds on Utility Confidence (UConf)	189
7.6	HUIs mined by the proposed algorithms and others	190
7.7	Statistics of the optimum (utility) values obtained over 25 Runs	202
7.8	Memory Consumed (in Mega Bytes) by BDE and ABDE	203
7.9	Time Consumed (in Seconds) by BDE and ABDE	203
7.10	Number of association rules mined at various thresholds	208
7.11	Statistics of the optimum (Utility * Uconf) obtained for 25 Runs	220
7.12	Memory Consumed (in Mega Bytes) by BDE-HUARM and ABDE-HUARM	221
7.13	Time Consumed (in Seconds) by BDE-HUARM and ABDE-HUARM	221
8.1	Results of Proposed Algorithms in AUC	243
8.2	Wilcoxon Signed Rank Test	244
8.3	Comparison with the State-of-the-art using EER	245
8.4	Results of Churn Dataset	247
8.5	Results of Insurance Fraud Dataset	247
8.6	Results of Default Dataset	248
8.7	Parameters of MDE-based Subspace Anomaly Detection	248
9.1	The Six Concepts	256
9.2	Sequential mode of training results for FCM	257
9.3	Batch mode of training results for FCM	258

10.1	Encoding and Encryption Example	277
10.2	Results of Case Study (Case-1)	279
10.3	Results of Case Study (Case-2)	279
10.4	Average optimal objective function values of OBJ_{BO} and OBJ_{SO}	282
10.5	Comparison of Entropy values generated by OBJ_{BO} vs computed from the OBJ_{SO}	284
10.6	Average of optimal objective values of OBJ_{WS}	285
10.7	Results of Encryption Time in seconds	287
10.8	Results of Encryption Time in seconds-(a)	288
10.9	Results of Encryption Time in seconds-(b)	289
10.10	Results of Compression Percentages	291
10.11	Parameter Setting	293
10.12	Comparison of proposed KGAs with Standard DRBGs (HMAC, Hash) .	297
C.1	Results of CEC 2010, CEC 2012, CEC 2013 and SOCO 2011 LSGO winners and proposed hybrid algorithms applied on CEC 2013 LSGO Benchmarks (F1, F2, F3, F4, F5, F6, F7, F8)	355
C.1	Results of CEC 2010, CEC 2012, CEC 2013 and SOCO 2011 LSGO winners and proposed hybrid algorithms applied on CEC 2013 LSGO Benchmarks (F1, F2, F3, F4, F5, F6, F7, F8) (Continued)	356
C.1	Results of CEC 2010, CEC 2012, CEC 2013 and SOCO 2011 LSGO winners and proposed hybrid algorithms applied on CEC 2013 LSGO Benchmarks (F1, F2, F3, F4, F5, F6, F7, F8) (Continued)	357
C.2	Results of CEC 2010, CEC 2012, CEC 2013 and SOCO 2011 LSGO winners and proposed hybrid algorithms applied on CEC 2013 LSGO Benchmarks (F9, F10, F11, F12, F13, F14, F15)	357
C.2	Results of CEC 2010, CEC 2012, CEC 2013 and SOCO 2011 LSGO winners and proposed hybrid algorithms applied on CEC 2013 LSGO Benchmarks (F9, F10, F11, F12, F13, F14, F15) (Continued)	358
C.2	Results of CEC 2010, CEC 2012, CEC 2013 and SOCO 2011 LSGO winners and proposed hybrid algorithms applied on CEC 2013 LSGO Benchmarks (F9, F10, F11, F12, F13, F14, F15) (Continued)	359

D.1	Fact sheet of Engineering Applications of Artificial Intelligence journal publication [168]	360
D.2	Fact sheet of Applied Soft Computing journal publication [174]	360
D.3	Fact sheet of Optimization in Industry book chapter publication [171] .	361
D.4	Fact sheet of ICHSA-2015 publication [167]	361
D.5	Fact sheet of ICIA-2016 publication [169]	361
D.6	Fact sheet of ISDA-2018 publication [170]	362
D.7	Fact sheet of ISDA-2018 publication [175]	362
D.8	Fact sheet of CoDS COMAD-2019 publication [172]	362
D.9	Fact sheet of IEEE TENCON-2019 publication [173]	363

List of Algorithms

1	Improved and Modified Harmony Search (IMHS)	99
2	Modified Differential Evolution (MDE)	100
3	IMHS+MDE	101
4	Adaptive Differential Evolution (ADE)–FSS	122
5	Differential Evolution (DE) based Rule Miner	138
6	Adaptive Differential Evolution (ADE) based Rule Miner	139
7	Binary Differential Evolution (BDE) based HUIM	178
8	Adaptive Binary Differential Evolution (ABDE) based HUIM .	179
9	Binary Differential Evolution (BDE) based HUARM	185
10	Adaptive Binary Differential Evolution (ABDE) based HUARM	186
11	Sequential DE trained FCM	254
12	Batch DE trained FCM	255

Acronyms

ABC	Artificial Bee Colony
ABDE	Adaptive Binary Differential Evolution
ABOD	Angle Based Outlier Detection
AC	ASCII Characters
ACO	Ant Colony Optimization
ACR	Associative Classification Rules
ADACOM	Advanced Ant Colony Optimization Miner
ADE	Adaptive Differential Evolution
ADE-FSS	Adaptive Differential Evolution - Feature Subset Selection
AI	Artificial Intelligence
AIRS	artificial immune-based recognition system
AIS	Artificial Immune System
AKGA	Ant Colony-based Key Generation Algorithm
AM	Arithmetic Mean
ANN	Artificial Neural Network
AUC	Area Under ROC Curve
BDE	Binary Differential Evolution

BFFO-TA	Binary Firefly Optimization- Threshold Accepting
BFSI	Banking, Financial Services and Insurance
BPSO	Binary Particle Swarm Optimization
BPSO-TA	Binary Particle Swarm Optimization- Threshold Accepting
BW	Bandwidth
CART	Classification and Regression Tree
CBM	Constraint-Based Mining
CC-CMA-ES	Cooperative Coevolution with Co-variance Matrix Adaptation Evolutionary Strategy
CCEA	Cooperative Coevolution Evolutionary Algorithm
CCGS	Cooperative Coevolution Evolutionary Algorithm with Global Search
CEC	Congress on Evolutionary Computation
CHAID	Chi-square Automatic Interaction Detector
CLTV	Customer Lifetime Value
CLV	Customer Lifetime Value
CMAR	Classification Based on Multiple-class Association Rules
CMU	Carnegie Mellon University
CPAR	Classification Based on Predictive Association Rule
CRM	Customer Relationship Management
CS	Cuckoo Search
DASA	Differential Ant-Stigmergy Algorithm
DE	Differential Evolution

DEA-FMADM	Data Envelopment Analysis - Fuzzy Multi-Attribute Decision Making
DECC-DML	Differential Evolution with Cooperative Co-evolution - Delta Method
DE-KGA	Differential Evolution - Key Generation Algorithm
DE-FCM	Differential Evolution - Fuzzy Cognitive Map
DE-FSS	Differential Evolution - Feature Subset Selection
DE+TA	Differential Evolution + Threshold Accepting
DETA	Differential Evolution and Threshold Accepting
DE+TS	Differential Evolution + Tabu Search
DHL	Differential Hebbian Learning
DM	Data Mining
DMEL	Data Mining by Evolutionary Learning
DMS-PSO	Dynamic Multi-Swarm Particle Swarm Optimizer
DRBG	Deterministic Random Bit Generator
EA	Evolutionary Algorithm
EAF	Empirical Attainment Function
EAFS	Evolutionary Algorithm for Feature Selection
EA-PRNG	Evolutionary Algorithm-Pseudo Random Number Generator
EC	Evolutionary Computing
ECM	Expected Cost of Misclassification
EER	Equal Error Rate
ER	Equity Ratio

EMO	Evolutionary Multi-objective Optimization
EOEA	Ensemble Optimization Evolutionary Algorithm
ESO	Evolutionary and Swarm Optimization
EUCS	Estimated Utility Co-occurrence Structure
FCM	Fuzzy Cognitive Map
FE	Function Evaluations
FIM	Frequent Itemset Mining
FF-TA	Firefly Optimization- Threshold Accepting
FFO	Fire Fly Optimization
FMO-BFFO-TA	Fuzzy Multi-Objective - Binary Firefly Optimization - Threshold Accepting
FMO-BPSO	Fuzzy Multi-Objective - Binary Particle Swarm Optimization
FMO-BPSO-TA	Fuzzy Multi-Objective - Binary Particle Swarm Optimization - Threshold Accepting
FOIL	First Order Inductive Learner
FOREX	Foreign Exchange
FP-Growth	Frequent Pattern-Growth
FSS	Feature Subset Selection
G3P	Grammar Guided Genetic Programming
GA	Genetic Algorithm
GABL	GA Batch Concept Learner
GaDE	Generalized adaptive Differential Evolution
GASS	Genetic Algorithm hybridized with Scatter Search

GK	Gustafson-Kessel
GM	Geometric Mean
GNP	Genetic Network Programming
GP	Genetic Programming
GP-ACO	Genetic Programming- Ant Colony Optimization
HM	Harmony Memory
HMCR	Harmony Memory Considering Rate
HMS	Harmony Memory Size
HPSACO	Heuristic Particle Swarm Ant Colony Optimization
HS	Harmony Search
HTWUIs	High Transaction Weighted Utility Itemsets
HUARM	High Utility Association Rule Mining
HUIM	High Utility Itemset Mining
HUPE-BPSO	High Utility Pattern Extracting - Binary Particle Swarm Optimization
HUPE-BPSO+	High Utility Pattern Extracting - Binary Particle Swarm Optimization Tree
HUPE-GARM	High Utility Pattern Extracting - Genetic Algorithm with Ranked Mutation
HUPE-GARM+	High Utility Pattern Extracting - Genetic Algorithm with Ranked Mutation Tree
IMHS	Improved Modified Harmony Search
IMHS+DE-KGA	Improved Modified Harmony Search + Differential Evolution - Key Generation Algorithm

IMHS-KGA	Improved Modified Harmony Search - Key Generation Algorithm
IMHS+MDE	Improved Modified Harmony Search + Modified Differential Evolution
INESA	Improved Non-Equilibrium Simulated Annealing
jDElscop	Self-adaptive Differential Evolution algorithm for large scale optimization
jDEsps	Self-adaptive Differential Evolution algorithm with small population size
JOA	Joint Operations Algorithm
KGA	Key Generation Algorithm
KNN	K-Nearest Neighbour
LCV	Lifetime Customer Value
LDA	Linear Discriminant Analysis
LMDEa	Differential Evolution with Landscape Modality Detection and a Diversity Archive
LOCI	Local Outlier Correlation Integral
LOF	Local Outlier Factor
LR	Logistic Regression
LSGO	Large Scale Global Optimization
LTV	Lifetime Value
MA	Memetic Algorithm
MA-ALSD	Memetic Algorithm with Adaptive Local Search Depth
MA-SW-Chains	Memetic Algorithm and Solis-Wets' with local Chains

MABC	Memetic Artificial Bee Colony Optimization
MAPE	Mean Absolute Percentage Error
MBA	Market Basket Analysis
MBO	Migrating Birds Optimization
MBPSO	Modified Binary Particle Swarm Optimization
MCC	Matthews Correlation Coefficient
MDA	Multiple Discriminant Analysis
MDE	Modified Differential Evolution
MGDA	Modified Great Deluge Algorithm
MGP	Modified Genetic Programming
MHS	Modified Harmony Search
MHT	Mutated Huffman Tree
MHS+MGDA	Modified Harmony Search + Modified Great Deluge Algorithm
MHSTA	Modified Harmony Search and Threshold Accepting
MLP	MultiLayer Perceptron
MO-BPSO	Multi-Objective Binary Particle Swarm Optimization
MO-BPSO-TA	Multi-Objective - Binary Particle Swarm Optimization - Threshold Accepting
MO-BFFO-TA	Multi-Objective - Binary Firefly Optimization - Threshold Accepting
MO-PSO	Multi-Objective Particle Swarm Optimization
MOEA	Multi-Objective Evolutionary Algorithm
MOGA	Multi-Objective Genetic Algorithm

MOS	Multiple Offspring Sampling
MURAME	MUlticriteria RAnking MEthod
NB	Naive Bayes
NGP	Normalized Inputs Genetic Programming
NHPSO	Novel Hybrid Particle Swarm Optimization
NMA	Nelder Mead Algorithm
NN	Neural Network
NSGA-II	Non-dominated Sorting Genetic Algorithm-II
NSGA-II-KGA	Non-dominated Sorting Genetic Algorithm-II-Key Generation Algorithm
PA	Probit Analysis
PAR	Pitch Adjusting Rate
PAM	Partitioning Around Medoids
PKGA	Particle Swarm-based Key Generation Algorithm
PM	Profit Margin
PM	Profit Margin
PNN	Probalistic Neural Network
PRNG	Psuedo Random Number Generator
PSO	Particle Swarm Optimization
PSOTA	Particle Swarm Optimization and Threshold Accepting
PV	Predetermined Value
QDA	Quadratic Discriminant Analysis
RBC	Rule-based Classification

RCS	Reliability of Complex Systems
RFM	Recency, Frequency and Monetary
RL	Random Localization
RMSE	Root Mean Square Error
RNG	Random Number Generator
ROA	Return on Assets
ROC	Receiver Operating Characteristic
ROE	Return on Equity
ROI	Return on Investment
RS	Random Search
RWDE	Random Walk with Direction Exploitation
RWS	Roulette Wheel Selection
SA	Simulated Annealing
SabDE	Self-adaptive binary Differential Evolution
SCS	Simple Classifier System
SDE	Sequential Differential Evolution
SDENS	Sequential Differential Evolution with Neighborhood Search
SHS	Sub-regional Harmony Search
SI	Swarm Intelligence
SOM	Self Organizing Map
SPEA-2	Strength Pareto Evolutionary Algorithm - 2
SVM	Support Vector Machine

TA	Threshold Accepting
TFE	Total Function Evaluations
TS	Tabu Search
TWDC	Transaction Weighted Downward Closure
TWU	Transaction Weighted Utility
UA	Utilization of Assets
UAC	Unique ASCII Characters
UCI	University of California, Irvine
WCA	Water Cycle Algorithm
WOE	Weight of Evidence
WUR	Weight Update Rule

Chapter 1

Introduction

1.1 Introduction

Evolutionary Computing (EC) is a name for a collection of optimization techniques propelled by biological evolution, such as, selection and genetic redesigning. Swarm computing motivated of the foraging and flocking behaviour of insects, flying creatures, and animals. These methods are finding a host of uses in differing disciplines. They form a subset of a greater set called optimization procedures, which incorporate traditional and differentiation-based methods as well as point-based meta-heuristics, such as for instance, simulating-annealing [156], threshold accepting [85], tabu search [107, 108], and so forth.

Evolutionary and Swarm Optimization (ESO) algorithms are population-based random search procedures in which the population of solutions are refreshed utilizing heuristics until the point of convergence by an optimization problem is accomplished, regardless of whether it is a continuous or discrete/combinatorial. These systems vary in how the solutions are encoded in addition to heuristics they normally use while refreshing the solutions. Genetic algorithm (GA) [110], differential evolution (DE) [293], particle swarm optimisation (DE) [150], ant colony optimisation (ACO) [67], firefly-based optimisation (FFO) [320], cuckoo search (CS) [320], artificial bee colony (ABC) [146], and the harmony search (HS) [104] are part of the family.

Performance, profitability, and risk-reduction would be the main goals that the banking, financial and insurance sectors attempt to achieve. In this data-driven world, performance, profitability, and risk-reduction are influenced by the analytical and oper-

ational technologies which could give better insights along with return on the investment for the banking, financial and insurance sectors. The focus of these technologies is to provide efficient models, enhance customer service, and to protect their systems against cyber threats.

Traditionally, data mining (DM) techniques have already been applied to resolve various analytical problems within the Banking, Financial and Insurance (BFSI) sector. In turn, optimization techniques have traditionally been employed for training several of the DM techniques. However, in the past several years, evolutionary techniques have grown to be so powerful and versatile that they can be deployed as an alternative for many DM techniques. This trend caught the interest associated with the researchers involved in the analytical area because they too started solving the analytical problems using evolutionary techniques alone. Analytical problems of BFSI sector are listed below where EC techniques have an application: 1) Customer Segmentation, 2) Customer Lifetime Value, 3) Market Basket Analysis, 4) Credit Scoring, 5) Fraud Detection, 6) Churn Detection, 7) Direct Marketing, Customer Campaigning and Service Marketing, etc.

Solving operational problems such as for instance Security, Efficiency Prediction, Portfolio optimization, Bankruptcy prediction, FOREX rate prediction, etc., banking, financial and insurance industries are looking beyond conventional ways toward more automated and much more robust methods. ESO-based techniques play an important role in solving the above-mentioned operational problems simply because they yield global or near-global optimal results.

With those stated earlier analytical and operational aspects of BFSI sector, it clear that the effective use of evolutionary computing techniques can help automate and solve the business problems that may, in turn, help BFSI sector in gaining a return on the investment and an advantage over competitors. With this, evolutionary computing techniques assist the BFSI industry in alleviating their analytical and operational problems in conducting business.

1.2 Motivation

While EC techniques made steady inroads into the analytical and operational areas, and it has not done as much as it has in other fields of application. The work demonstrates

that EC techniques applied to solve the analytical and operational problems which still offers a fertile ground, where EC techniques, either in isolation or in combination, have a tremendous role to play.

The most striking observation from the review is that EC techniques turn out to be viable alternatives for classical DM techniques like Neural Networks, Support Vector Machines, Decision Trees, K-means, etc. as sometimes they offer better results in less time. EC techniques solve DM tasks by generating ‘if-then’ rules from the underlying data. These rules can be considered as early warning expert systems in tasks such as fraud detection, churn detection, default detection, credit scoring, etc. In this sense, EC techniques, are termed as ‘transparent systems’ as opposed to black boxes like Logistic Regression, Multilayer Perceptron, Support Vector Machines, etc. Considering the above discussion, algorithms are developed which can perform the data mining tasks of classification, clustering, association rule mining, anomaly detection and regression using EC techniques alone without using standard DM techniques.

EC techniques like DE, proved to be better in dealing with multi-modal, non-linear, and non-differential objective functions and non-convex search spaces. Further, hybrid EC techniques have immense potential in solving these tasks in less time with higher accuracies. However, the challenge is the correct choice of constituent EC techniques and the suitable design of the hybrid. Many analytical problems can be formulated as multi-objective optimization problems by suitably identifying the multiple objectives. Then, Evolutionary Multi-objective Optimization techniques can be employed, leading to innovation in decision making. It is well known that user-defined parameters must be tuned carefully for the EC techniques. It consumes time and effort to fine tune and identify the optimal combination of parameters for the EC techniques in a given application. Like all other non-parametric, intelligent techniques, this is one limitation of these techniques. But, this limitation can be overcome by moving towards adaptive versions of EC. For instance, DE, which outperformed GA in many problems in other domains, can be employed to investigate its potential for solving analytical and operational problems. Therefore, in the current study, single objective optimization algorithm i.e. DE as well as an adaptive version of DE are utilized for all problems and also a multi-objective optimization algorithm i.e. NSGA-II is utilized in one problem.

The motivation for the feature subset selection, which is an important subtask in all the data mining tasks, is to propose a novel, adaptive DE-based wrapper FSS

algorithm, where there would be no need to tune the control parameters of DE manually. Another motivation for the work is to utilize a sound evaluation measure than AUC, namely Matthews Correlation Coefficient (MCC), for the ADE-based wrapper. The repetitive features from this study are passed to problem of rule-based classification to judge the efficacy of this study.

EC techniques solve DM tasks by generating ‘if-then’ rules from the underlying data. These rules can be considered as early warning expert systems in problems such as fraud detection, credit scoring, etc. Therefore, with the motivation to generate transparent models, a rule-based classifier is developed for the above problems of fraud detection and credit scoring using differential evolution and adaptive differential evolution.

It is found that high utility itemset mining, generates high amount of return on investment as frequent itemsets are not always interesting and sometimes rare itemsets yield high profit. In this connection, it is observed that evolutionary algorithms find an interesting application in this area as well, with utility as the objective function. Therefore, a novel high utility itemset and association rule mining algorithms are developed using binary differential evolution.

As the dimensionality increases every point becomes a potential anomaly. Therefore, to address this problem of anomaly detection, a subspace-based anomaly detection algorithm is developed using the sparsity coefficient as an objective function. But, dividing the feature space into subspaces entails the combinatorial exploration of the subspaces. To search all the subspaces is complex and it can’t be achieved within a stipulated time as dimensionality increases to a larger value. To overcome this, differential evolution and harmony search separately are employed to search for the optimal feature subspaces.

The motivation for Differential Evolution (DE) trained Fuzzy Cognitive Map (FCM) is to automate the decision-making process in the banking domain with multiple solutions from Differential Evolution (DE) where the bank efficiency is never predicted as a function of the important financial ratios using FCM.

The motivation for the final work, i.e. providing security to the information by efficient key generation technique using existing evolutionary computing algorithms, came from three points listed below. Firstly, the role of evolutionary computing techniques is providing security to the mother PRNG by hiding mother PRNG’s functional-

ity. Thus, evolutionary computing provides additional security around mother PRNG. Consequently, it will be hard for the attacker to compromise the mother PRNG. Secondly, evolutionary technique-based key generation followed by compression along with encryption was never reported before. In order to achieve compression, Mutated Huffman Coding technique is invoked, which also adds to the security of the plain text. Adopting entropy for key generation to generate uniformly distributed key streams is the third motivation. Therefore, it becomes hard for the attacker to crack the key. Thus, key generation problem was formulated as a bi-objective optimization problem that now includes entropy too.

1.3 Problem Statement

1.3.1 Tested Evolutionary Computing Techniques

Depending on the necessity, a few of the state-of-the-art evolutionary algorithms, the variants of these evolutionary algorithms, a novel memetic algorithm, also an adaptive type of evolutionary algorithm and a multi-objective evolutionary algorithm are employed to solve the problems considered within the thesis.

Initially, to start out with, single objective variants, in other words Improved Modified Harmony Search (IMHS), Modified Differential Evolution (MDE) are developed and also few of the state-of-the-art evolutionary computing algorithms are utilized. In addition to the above variants, a novel memetic algorithm is also developed, in other words IMHS+MDE.

- The evolutionary algorithms are tested (i.e. Harmony Search (HS) and Differential Evolution (DE)), the variants (i.e. IMHS and MDE) and also the novel memetic algorithm (i.e. IMHS+MDE) on optimization test functions as well as constrained continuous reliability optimization problems.
- Also tested the novel memetic algorithm (i.e. IMHS+MDE) on large-scale global optimization (LSGO) test suite developed by Congress of Evolutionary Computing (CEC), i.e. CEC LSGO 2013.

1.3.2 Evolutionary Computing techniques applied to solve some Analytical problems

Numerous DM techniques will need optimization at various stages, such as for instance feature selection, training a classifier/regression model or determining the suitable user-defined parameter combination that will result bring about accurate outcomes for the technique. Having said that, evolutionary techniques may be employed in standalone mode to deal with various data mining tasks such as classification, regression, clustering, association rule mining, and anomaly detection, alongside subtasks such as feature selection. Given this established versatility of EC techniques, it's possible to safely imagine that EC techniques may be employed to resolve a few of the analytical problems of BFSI sector. A few of the analytical problems of BFSI sector which are solved by using evolutionary computing are listed below:

- Selecting important features for credit scoring and detecting financial statement fraud by wrapper-based feature subset selection techniques (addressed the data mining subtask of feature selection using evolutionary computing);
- Rule based classification on all the features of credit scoring and financial statement fraud data as well as the feature subsets from above wrapper-based feature subset selection (addressed the data mining task of classification using evolutionary computing);
- Customer segmentation based on monetary value by High Utility Itemset Mining (HUIM) and High Utility Association Rule Mining (HUARM) (addressed the data mining task of Market Basket Analysis using evolutionary computing);
- Anomaly detection in CMU's keystroke detection, churn, default and insurance fraud detection by evolutionary computing-based subspace anomaly detection (addressed the data mining tasks of clustering and anomaly detection using evolutionary computing).

1.3.3 Evolutionary Computing techniques applied to solve some Operational problems

BFSI sector faces many problems in the operational front while servicing the customers, where evolutionary computing may become handy. Whenever operational problems are

considered, conventional techniques usually do not yield comparatively better results compared to the evolutionary-based techniques. A few of the operational problems associated with the BFSI sector which are solved by evolutionary computing are listed below:

- Efficiency prediction of banks with a Differential Evolution trained Fuzzy Cognitive Map (addressed the data mining task of regression along with the operational problem of predicting the efficiency of banks using evolutionary computing);
- Generating keys for plain text in stream cipher via bi-objective evolutionary computing (addressed the operational problem of security using evolutionary computing).

1.4 Contributions in the Thesis

The contributions in the thesis are listed as follows:

- Developed and tested the evolutionary algorithms (i.e. Harmony Search (HS) and Differential Evolution (DE)), the variants (i.e. IMHS and MDE) and also the novel memetic algorithm (i.e. IMHS+MDE) on optimization test functions as well as constrained continuous reliability optimization problems.
- Also developed and tested the novel memetic algorithm (i.e. IMHS+MDE) on large-scale global optimization (LSGO) test suite developed by Congress on Evolutionary Computing (CEC), i.e. CEC LSGO 2013.
- Proposed a novel, adaptive wrapper-based FSS algorithm, where there would be no need to tune parameters manually. In this study, a sound evaluation measure namely Matthews Correlation Coefficient (MCC) is utilized for the ADE-based wrapper.
- Developed a rule-based classification algorithms using EC techniques on all the features of credit scoring and financial statement fraud data as well as on the feature subsets from above wrapper-based feature subset selection.

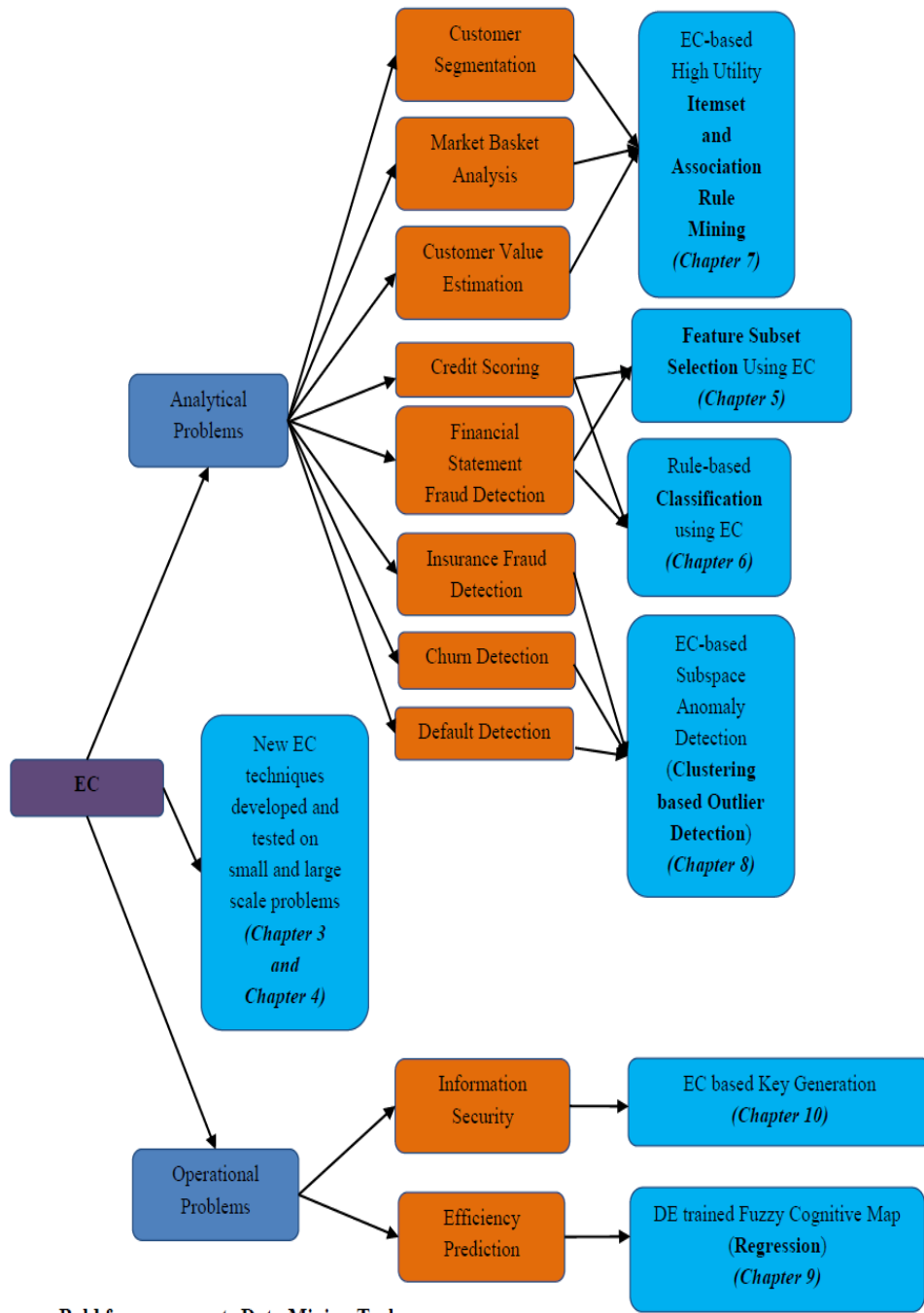


Figure 1.1: Overarching Diagram for the Ph.D. Thesis

-
- Developed powerful EC based HUIM and HUARM algorithms. Explored a possible application of the HUIM and HUARM for the business needs, i.e. customer segmentation based on monetary value of the customers. Novel objective function for HUARM task is developed.
 - Developed a Harmony Search and a Differential Evolution based anomaly detection algorithms with sparsity-based subspace clustering for numerical datasets and applied them to CMU's keystroke dataset, churn dataset, default dataset and insurance fraud dataset.
 - Developed a novel Differential Evolution (DE) trained Fuzzy Cognitive Map (FCM) for predicting bank efficiency. Two modes of training namely (i) sequential and (ii) batch modes are employed. The DE trained FCM models are compared with conventional Hebbian training in both modes.
 - Developed novel key generation algorithms using Non-Dominated Sorting Genetic Algorithm-II (NSGA-II) in the bi-objective optimization framework and Improved Modified Harmony Search + Differential Evolution (IMHS+DE), Differential Evolution (DE) and Improved Modified Harmony Search (IMHS), in the single objective optimization framework.

1.5 Organisation of the thesis

This thesis consists of 11 chapters including an introductory chapter and a concluding chapter. The content of each of these chapters is summarised below:

Chapter 1: Introduction

This chapter is an introductory chapter. This chapter provides the motivation and necessary background for the work reported in this thesis. The chapter begins with the application of evolutionary computing to solve some analytical and some operational problems in the BFSI sector. Later, it presents the motivation of work, defines the problem statement, describes the problems found in the BFSI sector and lists the contributions of thesis related to the problem statement.

Chapter 2: Systematic Review of the Related Work

-
- In this chapter, the research papers are surveyed, in which the application of evolutionary computing (EC) techniques to analytical and operational problems within the BFSI sector will be the focus. The chapter includes papers involving evolutionary computing techniques applied to solve the analytical and operational problems within the BFSI sector under single- along with multi-objective optimization framework. The goal of this chapter would be to allow the reader to realize the versatility and power of EC techniques in solving analytical and operational problems within the BFSI industry.
 - This chapter provides an in-depth review centered on analytical problems of BFSI sector like MBA, fraud detection, churn detection, direct marketing, customer segmentation, customer lifetime value (CLTV), default detection, service marketing and customer campaigning. The chapter also details how EC techniques may be used to enhance the relationship with the customer, thereby enhancing the return on the investment into the company.
 - Another part of the chapter would be to present and explore the research articles in which the EC techniques are utilized to resolve a few of the operational problems associated with the BFSI sector. This chapter's central theme would be to discuss how EC techniques may be utilized to solve the problems better than conventional techniques. Conventional techniques are now and again helped by EC techniques to solve the operational problems, and quite often EC techniques alone may be used to solve the operational problems. Conventional techniques include techniques such as for instance machine learning-based techniques, expert-based techniques, statistical-based techniques, age-old practices. Only some of the operational problems of banks considered here and are not exhaustive.

Chapter 3: A Hybrid and the variants of Differential Evolution and Harmony Search

- In this chapter, a hybrid optimization algorithm of Harmony Search and Differential Evolution is proposed and applied to three reliability complex systems with static, extinctive constraint handling.
- The proposed hybrid is contrasted with Harmony Search, Improved Modified Harmony Search, Differential Evolution, Modified Differential Evolution as well

as other algorithms previously employed for optimizing Reliability of Complex Systems.

- It is experimentally discovered that the proposed hybrid, Improved Modified Harmony Search + Modified Differential Evolution needs fewer function evaluations than the contrasted algorithms.

Chapter 4: Large Scale Global Optimization via Improved Modified Harmony Search and Modified Differential Evolution Hybrid

- In management, business, economics, science, engineering, and research domains, Large Scale Global Optimization (LSGO) plays a predominant and vital role. Though LSGO is applied in lots of associated application domains, it is a very troublesome and a perverse task.
- The Congress on Evolutionary Computation (CEC) began an LSGO competition to come up with algorithms for the standard benchmarks devised for unconstrained LSGO functions.
- Therefore, in this chapter, a hybrid meta-heuristic algorithm is proposed, which combines Improved Modified Harmony Search (IMHS), along with a Modified Differential Evolution (MDE) with an alternative selection strategy.
- Harmony Search (HS) does the task of exploration and exploitation, and Differential Evolution does the task of giving a perturbation into the exploration of IMHS, as harmony search suffers from being stuck during search at the basin of local optimal.
- To judge the performance of the suggested algorithm, the proposed algorithm is compared with ten excellent meta-heuristic algorithms on fifteen LSGO benchmark functions, which are all of 1000 continuous decision variables, as per the CEC 2013 LSGO special session.
- The experimental results consistently show that the proposed hybrid meta-heuristic performs statistically on par with a few algorithms in a few problems, although it ended up being the best in a couple of problems.

Chapter 5: Feature Subset Selection using Adaptive Differential Evolution: An application to banking

- In this chapter, a feature subset selection method is developed by employing Adaptive Differential Evolution as a wrapper.
- The proposed wrapper utilizes four independent classifiers namely Logistic Regression, Probabilistic Neural Network, Naive Bayes and Support Vector Machine.
- The Matthews Correlation Coefficient (MCC) is employed as the fitness function or the evaluation measure.
- To demonstrate the efficacy of the proposed method, it is tested on three datasets, of which two are linked to credit scoring, and another relates to fraud in the statement of finance.
- The proposed method yielded better results than the standard methods earlier employed and the Differential Evolution-based wrapper.
- A statistical significance test is also performed, i.e. t-test at a 1% significance, which infers that some of the proposed wrappers are statistically the same while others are not.

Chapter 6: Rule-based Classification via Differential Evolution for the banking applications

- Developed rule-based classification algorithms using DE and ADE.
- The Sensitivity $\times$ Specificity is utilized as the objective function.
- The features from ADE-SVM based feature subset selection with repeatability are also utilized.
- The rule-based classification algorithms are applied on credit scoring and financial statement fraud datasets.
- DE driven rule-based classification algorithm and ADE driven rule-based classification algorithm were found to be statistically equal.

Chapter 7: High Utility Itemset and Association Rule Mining using Binary Differential Evolution: An application to Customer Segmentation

- In this thesis, a Binary Differential Evolution (BDE) and an Adaptive Binary Differential Evolution (ABDE)-based high utility itemset and high utility association rule mining are proposed separately for the high utility tasks, which are itemset and association rule mining.
- High Utility Itemset Mining (HUIM) with the BDE and ABDE are compared with the extant HUIMs earlier employed which are (i) Binary Particle Swarm Optimization, (ii) Genetic Algorithm and (iii) a two-phase HUIM.
- Later, in a first-of-its-kind-study, High Utility Association Rule Mining (HUARM) making use of the BDE as well as the ABDE are proposed and compared.
- These HUI and HUARM algorithms are applied separately on six datasets and a real-life Online Retail dataset. The real-life Online Retail dataset is used as an application of HUIM and HUARM for customer segmentation driven by value, in other words. utility.
- The results show that BDE based HUIM and HUARM outperformed ABDE based HUIM and HUARM on all seven datasets. The BDE and ABDE based HUIM outperformed the extant HUIM algorithms on three datasets with regards to the number of itemsets generated, while for the fourth dataset, mixed results were obtained.

Chapter 8: Subspace based Anomaly Detection via Harmony Search and Differential Evolution: An application to Analytical CRM and Security

- Harmony Search and Differential Evolution based anomaly detection algorithms are developed and employed on CMU's keystroke dataset, churn dataset, default dataset and insurance fraud dataset.
- The sparsity coefficient is taken as objective function for finding anomalies within the churn, insurance fraud and default data as well as the CMU's keystroke dataset.

-
- A statistical significance test is conducted from the optimal sparsity coefficient.

Chapter 9: Differential Evolution trained Fuzzy Cognitive Map: An application to modelling efficiency in banking

- In this chapter, a Differential Evolution (DE) trained Fuzzy Cognitive Map (FCM) is developed for predicting bank efficiency.
- Two modes of training were employed namely (i) sequential and (ii) batch modes. The DE trained FCM models are compared with conventional Hebbian training in both modes.
- Mean Absolute Percentage Error (MAPE) is employed as an error measure while predicting the efficiency from five financial ratios.
- 5x2-fold cross-validation framework is employed.
- In the 1st case, which is sequential mode of training, the DE trained FCM statistically outperformed the Hebbian trained FCM. But, in the next case, which is batch mode of training, DE trained FCM is statistically same as the Hebbian trained FCM.
- Therefore, training times in the batch mode are compared to break the tie, where DE trained FCM turned to be 19% faster than the Hebbian trained FCM.
- The proposed model can be employed on solving similar banking and insurance problems.

Chapter 10: NSGA-II application in Security: A case of key generation for plain text in stream cipher

- Evolutionary algorithms are widely used to solve a wide variety of continuous, discrete and combinatorial optimization problems. Evolutionary multi-objective optimization problems seek Pareto front so that you can negotiate the trade-off amongst various objective functions contained in the problem.
- Much of the works on cryptography centers around making the inference problem harder, for securing the information.

-
- In this thesis, key generation algorithms are developed using Non-Dominated Sorting Genetic Algorithm-II (NSGA-II) in the bi-objective optimization framework and Improved Modified Harmony Search + Differential Evolution (IMHS+DE), Differential Evolution (DE) and Improved Modified Harmony Search (IMHS), in the single objective optimization framework.
 - For encoding the keystream thus generated as well as the plain text, the Mutated Huffman Tree Coding algorithm is utilized.
 - In the next phase, the encoded keystream as well as the encoded plain text is encrypted to generate the cipher text.
 - The cipher text is then decrypted by making use of the encoded key stream accompanied by the decoding the deciphered text with the code tables.
 - Following the earlier works, random texts of varying lengths were generated and code table sizes in order to demonstrate the effectiveness of the proposed method.
 - The proposed algorithms are compared against the extant methods.
 - When it comes to bi-objective optimization, the empirical attainment function (EAF) surface is also plotted to summarize the effectiveness of the NSGA-II based key generation algorithm.
 - Of significance is the highest entropy value yielded by the NSGA-II based algorithm, which in turn indicates the strength of the keys generated by the NSGA-II.

Chapter 11: Conclusion and future directions

This chapter summarises the contributions of the thesis and outlines future directions.

Chapter 2

Systematic Review of the Related Work

2.1 Introduction

Evolutionary computing is the aggregate name for a set of problem-solving strategies inspired by biological evolution, for instance, natural selection and genetic re-engineering. Swarm intelligence that is inspired by the flocking and foraging behavior of insects, birds, and other creatures also forms part of this grand set. These techniques have found a myriad of applications in diverse disciplines. They form a subset of a bigger set called optimization techniques, which include classical and derivative based techniques, point based meta-heuristics such as simulated annealing, tabu search, etc.

Evolutionary algorithms (EA) are, without exception, population-based random search techniques where a population of solutions gets updated iteratively using algorithm-specific heuristics until convergence is achieved. These techniques differ in the way in which the solutions are encoded and the heuristics they adopt when updating the solutions [24]. Genetic algorithms, differential evolution, particle swarm optimization, ant colony optimization, firefly algorithm, cuckoo search, artificial bee colony, and the harmony search algorithm are some of the members of this family.

2.1.1 Optimization Techniques and Data Mining (DM)

Numerous DM techniques require optimization at various stages, such as feature selection, training a classifier/regression model or determining the optimal user-defined

parameter combination that would result in accurate results for the technique. Fitting of logistic regression, wrapper-based feature selection techniques, parameter and structure optimization of the neural network, finding the optimal parameter combination for a support vector machine, fine tuning the parameters of a fuzzy inference system, and optimizing the rule base size in a fuzzy rule-based classifier are some of the examples in this category [231]. On the other hand, evolutionary techniques can be employed in standalone mode to solve various data mining tasks such as classification, regression, clustering, association rule mining, and outlier detection, along with subtasks such as feature selection. Given this established versatility of EC techniques, one can safely imagine that EC techniques alone can be employed to solve various analytical and operational problems. This survey centers around this unique strength of EC techniques and also includes the review of papers involving both standalone and hybrid EC techniques in a single- as well as a multi-objective optimization framework. The works dealing with soft computing hybrids involving EC on one hand and machine learning techniques on the other are outside the scope of this survey.

2.1.2 Analytical and Operational Problems in BFSI Sector

BFSI sector is moving away from conventional ways toward more automated and more robust methods. Evolutionary Computing (EC) based techniques play a vital role in solving the above-mentioned operational issues because they yield global or near-global optimal results. The following are few of the analytical and operational problems considered:

2.1.2.1 Customer Segmentation

Segments are homogeneous gatherings of comparative buyers with comparative needs and desires [112]. Customer segmentation is the division of customer base into a small number of homogeneous groups that have comparable qualities or attributes. Customer segmentation is an effective tool for recognizing unfulfilled customer needs. Organizations can identify underserved segments and then beat the competition by serving them suitably and growing remarkably in the whole process. Customer segmentation can be termed successful when an organization caters to those segments that are the most profitable. It helps organization practice target marketing and eventually leading to

personalized marketing. Amazon, Google, and other such companies already practice one-to-one or personalized marketing. This prioritization can help organizations evaluating techniques to both high- and low-benefit clients [112].

2.1.2.2 Direct Marketing, Customer Campaigning and Service Marketing

Direct Marketing is performed by purchasing the database of customer profiles and immediately showcasing the product to a specific group of customers, whereas, customer campaigning is performed by presenting a product to the customers by way of advertisements, rebates and regular mails [163]. Service marketing incorporates showcasing of services to the customer, for example, information transfer services, financial services, rental services, insurance services, expert services, etc., [163].

2.1.2.3 Customer Lifetime Value (CLTV)

CLV or CLTV is also called lifetime customer value (LCV), or lifetime value (LTV). It is one of the important analytical problems. It is an expectation of the net revenue, through tangible or intangible benefit obtained by future association with a customer. It is the present value of money streams associated with the customer by his/her whole relationship with the organization —[112]. This can be estimated by the recency, frequency and monetary (RFM) aspects of his/her association with the company.

2.1.2.4 Credit Scoring

Credit scoring assessment results in a numerical expression taking into account the individual's credit documents that speak financial status of the individual. The rating is arrived at basically from the credit report data normally sourced from credit agencies. It can also be utilized to profile customers [112]. Credit scoring assesses the credit worthiness of a customer.

2.1.2.5 Market Basket Analysis (MBA)

Market basket analysis is based on the hypothesis that many customers tend to purchase a certain group of things together [118]. Therefore, after applying association rule mining algorithms on the customers' transactional data, we obtain association rules which together represent the knowledge. Using this knowledge, the customers who

purchased a subset of those products, can also be cross-sold other products, which are not yet purchased by them. Cross-sell and up-sell constitute the MBA. While cross-sell involves recommending those products not already held by a customer, up-sell typically involves recommending the upgraded products in the same category.

2.1.2.6 Fraud Detection

Fraud is deliberately acting to deny another from claiming his/her worth for one's own specific benefit. Fraud manifests itself in diverse forms. On the other hand, of late, the technological advancements while providing a lot of comfort and convenience to the customer also leave ample scope for perpetrating technology driven fraud. Detecting and identifying fraud becomes very critical to the profitability and reputation of a firm [233].

2.1.2.7 Default Detection

Credit scoring and Default detection are two sides of the same coin. While credit scoring happens before a loan is granted, default detection happens after a loan is granted. The default is the inability to meet the legitimate commitments. Default happens when the borrower has not made a planned installment of interest or principal, typically in specified period of time. Technically, default occurs when an agreed upon or a negative pledge is violated. Defaults can be predicted based on the customers' transactional and credit related data thereby saving huge losses to the financial firm [112].

2.1.2.8 Customer Churn Detection

Customer retention is much easier and cheaper proposition than customer acquisition because we have complete data about the existing customers. If we can analyze that data meticulously and perform cross-sell and up-sell, then they will become more profitable than before. However, we hardly have any information about the prospective or new customers. Despite this fact, service industries do lose profitable and loyal customers due to competition and various other reasons. In order to address this tendency, using predictive analytics, the firms must predict well in advance the set of customers who are going to attrite so that the marketing teams swing into action immediately to

have a dialogue with them and retain them. In that sense, customer churn detection and customer retention are two sides of the same coin [308].

2.1.2.9 Key Generation for Security

According to Basel II Committee, Operational Risk is defined as follows: “The risk of a change in value caused by the fact that actual losses, incurred for inadequate or failed internal processes, people and systems, or from external events”. Based on the above statement, operational risk is classified into risk emanating from people, process and technology. IT risk is subsumed under technology and security risk is subsumed under IT risk. Therefore, Security is the important operational issue under the broader category of IT risk. Therefore, key generation which is important part of security plays a vital role. Key generation is the process of generating keys for cryptography. The key is used to encrypt and decrypt data whatever the data is being encrypted or decrypted. Random numbers serve as keys in cryptography. Keys or random numbers are generated through a Random Number Generator (RNG) or a Pseudo Random Number Generator (PRNG) [28]. If the PRNG is compromised, then the cryptographic algorithm is compromised. Therefore, it is essential to use a PRNG which is hard to crack by any attacker. The randomness can be guaranteed by the evolutionary computing algorithms can be used in generating random numbers that are hard to break. Two of the cryptographic approaches used in symmetric-key are stream cipher [229] and block cipher [267].

2.1.2.10 Bank Efficiency Prediction

According to [246]: “Performance and efficiency of commercial banks are key elements of a country’s economy. As the banking industry is an important financial sector of the Indian economy, it is very important for senior managers, regulators and investors to identify the major drivers of a bank’s efficiency. Efficiency and profitability are the major performance indicators besides many measures (financial ratios) on which we can depend on in order to analyze the efficiency of banks”. Therefore, bank efficiency prediction is an important operational problem for banks and other organisations alike. The efficiency in turn depends on the several key financial indicators or ratios (please see Chapter 9 further details).

2.2 Theme of the proposed work

The main theme is to discuss how EC techniques can be utilized to solve the analytical and operational problems better than conventional techniques. Conventional techniques sometimes helped by EC techniques to solve the analytical and operational problems, and sometimes EC techniques alone can be used to solve the analytical and operational problems. Conventional techniques include machine learning based techniques, expert based techniques, statistical based techniques, age-old practices, etc. EC techniques include point based local search metaheuristics and population-based global search metaheuristics techniques. The main theme is divided into two sub-themes as discussed in the following subsections 2.2.1 and 2.2.2. These two themes core focus is to address the problems pertaining to BFSI sector.

2.2.1 Evolutionary Computation applied for solving Analytical Problems of BFSI sector

Analytical problems can be formulated as data mining (DM) tasks. Data mining and optimization go hand-in-hand when solving analytical problems. We can apply optimization alone for solving analytical problems, or we can apply optimization before or after applying DM techniques [231]. Considering the power and versatility of EC techniques, ***EC techniques can alone be applied to solve various analytical problems*** which fall into various buckets of data mining tasks, is one of the two parts of the study. This is shown in Figure 2.1, which depicts various analytical problems and the corresponding DM tasks used traditionally to solve these analytical problems. Having observed that EC techniques can be used as a viable alternative to traditional analytical and operational solutions, one has to remember that the performance of EC techniques is sometimes superior to that of traditional analytical and operational solutions, given the nature of all of these techniques encapsulated in the no-free-lunch theorem [315].



Figure 2.1: Relationship between Analytical problems, Data Mining, and Optimization (EC) Techniques

2.2.2 Evolutionary Computation applied for solving Operational Problems of BFSI sector

Only a two of the operational problems of the BFSI sector are considered here, and they are not exhaustive. Solving analytical problems is well documented (see subsection 2.2.1). So, we are going to focus on two of the important operational problems namely key generation for security problems and bank efficiency prediction separately as the other part of the theme. Figure 2.2 depicts these two operational problems considered for the other part of the study. These two operational problems are chosen based on their huge implication to the banking sector and also based on the fertile ground it offers, as little or no work is reported in these areas with EC techniques.

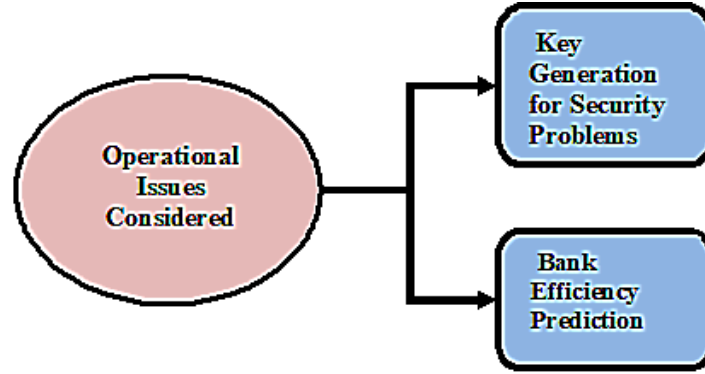


Figure 2.2: Operational Problems Considered

2.3 Review Methodology

2.3.1 Techniques relevant to the review

In an earlier survey, [220] reviewed pure and hybrid data mining (DM) techniques and DM techniques hybridized with optimizations techniques applied to analytical problems. Therefore, for our review paper to be unique, we considered stand-alone and hybrid, EC techniques applied in single- and multi-objective framework and reviewed those papers involving them. Figure 2.3 summarizes the above.

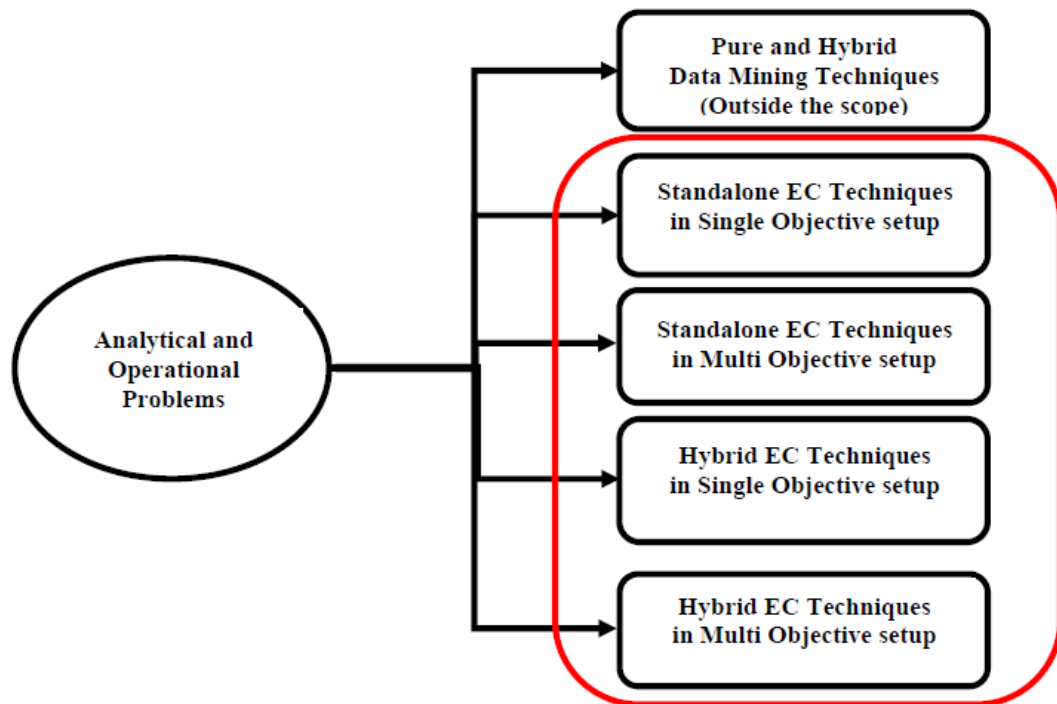


Figure 2.3: Structure of Considered Techniques

2.3.2 Flowchart of Review Process

Standard research databases are searched for the relevant articles to be included in the review. The order of search is journals, book chapters, and conference proceedings. Research articles which employ evolutionary algorithms for solving analytical and operational problems are searched for in the research databases. Then, the data set relevance to the banking, finance and insurance sectors is checked in the research articles. The flowchart presented in Fig. 2.4 depicts the methodology followed in collecting the articles for the review.

2.4 Distribution of Reviewed Articles

2.4.1 Count of Research Articles Reviewed by various Journals / Books / Proceedings

We surveyed around 21 papers that appeared in international journals with relevance to the topic of interest as shown in the flow chart. 23 book chapters and proceedings of conferences were also surveyed. So, a total of 44 research articles were surveyed. Expert Systems with Applications has the highest number of surveyed papers of 5 in journals, and IEEE has the highest number of conference proceedings with 13. The summary is presented in Table 2.1.

2.4.2 Distribution of Research Articles by Analytical and Operational Problems

This subsection summarizes the count of research articles arranged according to the analytical and operational problems in the Tables 2 and 3. From Tables 2 and 3, we can infer that customer lifetime value (LTV), default prediction, service marketing, and customer campaigning have the possible scope of further research for applying EC techniques. Tables 2.2 and 2.3, Figures 2.5 and 2.6 depict the distribution.

2.4.3 Distribution of Research Articles by EC Technique

This subsection summarizes the count of research articles arranged by EC techniques in Tables 4-9. From the Tables 2.4 to 2.9 and Figures 2.7 and 2.8, we can infer that GA was used extensively. We can also infer that there are also other EC techniques

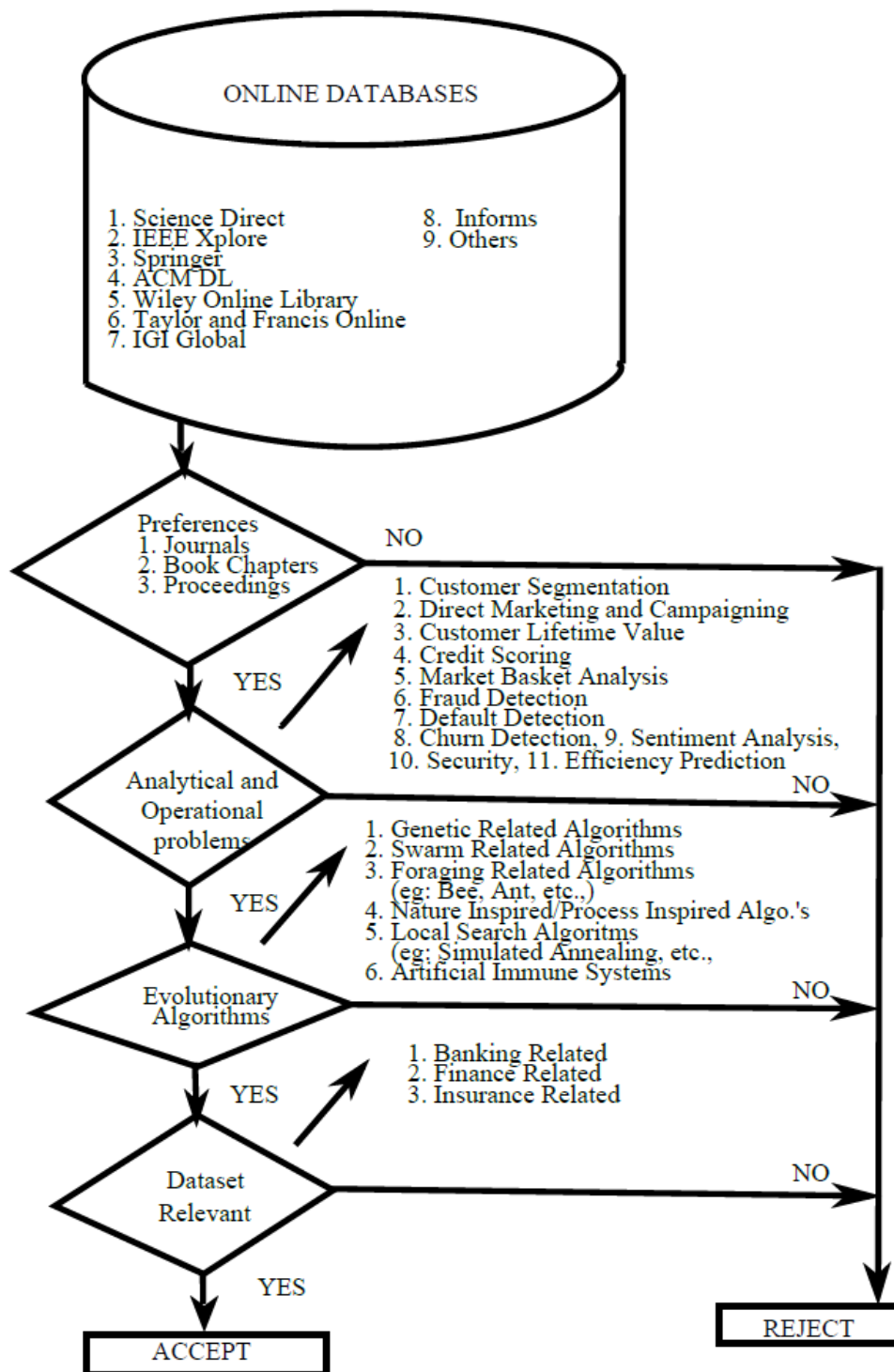
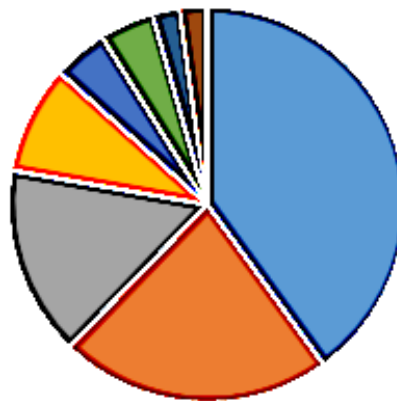


Figure 2.4: Flowchart of Review Methodology

Table 2.1: Count of Surveyed Papers

<i>Articles in Journals / Books / Proceedings</i>	<i>Number</i>
Expert Systems with Applications	5
Applied Soft Computing	2
IEEE Transactions on Evolutionary Computation	1
Soft Computing	1
Knowledge Based Systems	1
Data & Knowledge Engineering	1
Applied Mathematics and Computation	1
An International Journal: Computers & Mathematics with applications	1
Intelligent Systems in Accounting, Finance and Management	1
Engineering Applications of Artificial Intelligence	1
Information Systems Journal	1
Neural Computing and Applications	1
Security and Communication Networks	1
Journal of Information Assurance and Security	1
Journal of King Saud University - Computer and Information Sciences	1
IEEE Conference Proceedings	13
Springer Conference Proceedings	6
Book Chapters (Springer, Taylor & Francis, IGI Global)	4 (1,1,2)
Others	1
<i>Total</i>	<i>44</i>

**Analytical and Operational Problems
(Using EC Techniques)**



- Credit Scoring (40%)
- Fraud Detection (22%)
- Market Basket Analysis (16%)
- Key Generation (9%)
- Customer Churn detection (5%)
- Default detection (5%)
- Customer Segmentation (2%)
- Customer LTV (2%)
- Service Marketing (0%)
- Direct Marketing (0%)
- Customer Campaigning (0%)
- Bank Efficiency Prediction (0%)

Figure 2.5: Analytical and operational problems using EC

Table 2.2: Count based on analytical and operational problems

Solving Analytical and Operational problems (Using EC techniques)	No. of Journals	No of Conferences / Book Chapters	Total
Customer Segmentation	-	1	1
Fraud Detection	3	7	10
Market Basket Analysis	5	2	7
Credit Scoring	9	9	18
Customer LTV	-	1	1
Customer Churn detection	1	1	2
Default detection	1	1	2
Service Marketing	-	-	-
Direct Marketing	-	-	-
Customer Campaigning	-	-	-
Key Generation	3	1	4
Bank Efficiency Prediction	-	-	-

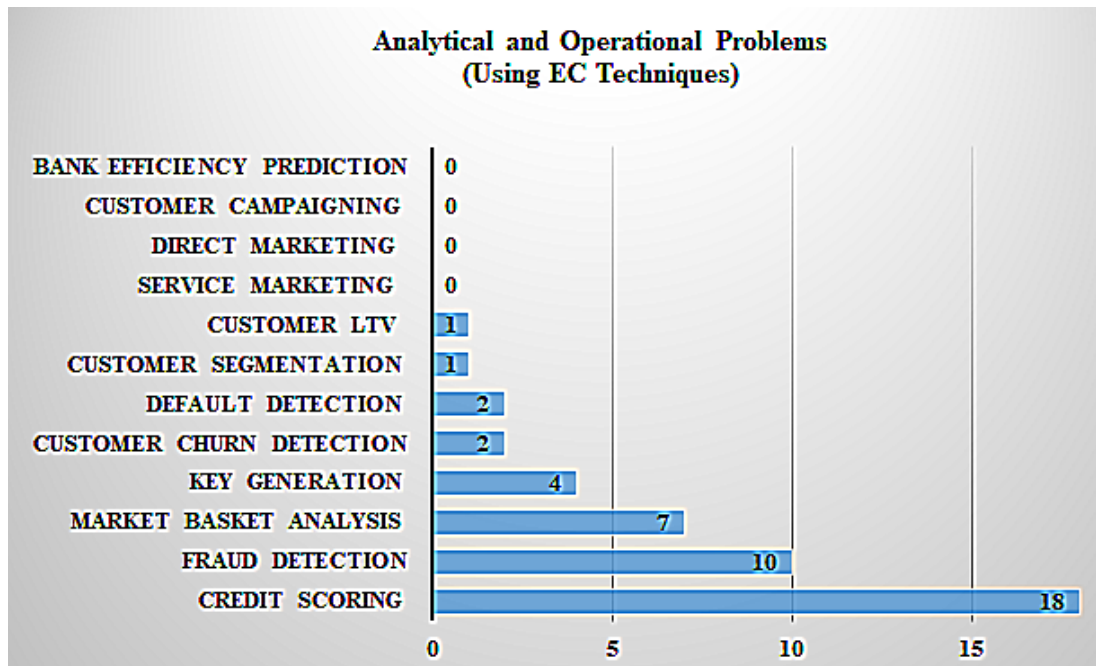


Figure 2.6: Bar Chart based on Count of Analytical and Operational problems

Table 2.3: Research Articles based on analytical and operational problems

Solving Analytical and Operational problems (Using EC techniques)	Journal articles	Conferences / Book Chapters
Customer Segmentation	-	[217]
Fraud Detection	[117, 127, 316]	[21, 36, 86, 102, 228, 245, 277]
Market Basket Analysis	[62, 177, 198, 263, 319]	[103, 244]
Credit Scoring	[11, 97, 115, 131, 165, 225, 262, 307, 312]	[18, 45, 59, 68, 83, 114, 124, 194, 310]
Customer LTV	-	[217]
Customer Churn detection	[22]	[88]
Default detection	[111]	[35]
Service Marketing	-	-
Direct Marketing	-	-
Customer Campaigning	-	-
Key Generation	[285, 286, 287]	[284]
Bank Efficiency Prediction	-	-

Table 2.4: Count based on Single Objective EC Techniques

Evolutionary Algorithms (Used for solving analytical and operational problems)	No. of papers in Jour- nals	No of papers in Con- ferences / Book Chapters	Total
Genetic Algorithms (GA)	5	4	9
Genetic Programming (GP)	3	6	9
Genetic Network Programming (GNP)	1	-	1
Grammar Guided Genetic Programming (G3P)	1	-	1
Particle Swarm Optimization (PSO)	3	4	7
Ant Colony Optimization (ACO)	3	1	4
Migrating Birds Optimization (MBO)	-	1	1
Simulated Annealing (SA)	-	1	1
Threshold Accepting (TA)	-	1	1
Firefly Optimization (FFO)	-	1	1
DMEL*	1	-	1
Artificial Immune System (AIS)	2	3	5

*DMEL=Data Mining by Evolutionary Learning

applied to analytical and operational problems, which is a direction for further research. Also, multi-objective optimization framework can also be considered for further research apart from single objective optimization framework. Hybrid EC frameworks developed by combining one or more EC frameworks can also be utilized for the analytical and operational problems as they feed on merits of one and another.

Table 2.5: Count based on Multi-Objective EC Techniques

Multi-Objective Evolutionary Algorithms (Used for solving analytical and operational problems)	No. of Journals	No of Conferences / Book Chapters	Total
Multi-Objective Genetic Algorithm	-	1	1
NSGA-II*	2	-	2
SPEA-2*	1	-	1
MO-BPSO*	-	1	1
MO-PSO*	1	1	2
MO-BPSO-TA*	-	1	1
MO-BFFO-TA*	-	1	1
FMO-BPSO*	-	1	1
FMO-BPSO-TA*	-	1	1
FMO-BFFO-TA*	-	1	1

*NSGA-II=Non-dominated Sort Genetic Algorithm-II

*SPEA-2= Strength Pareto Evolutionary Algorithm-2

*MO-PSO = Multi-Objective Particle Swarm Optimization

*MO-BPSO = Multi-Objective Binary Particle Swarm Optimization

*MO-BPSO-TA = Multi-Objective - Binary Particle Swarm Optimization - Threshold Accepting

*MO-BFFO-TA = Multi-Objective - Binary Firefly Optimization - Threshold Accepting

*FMO-BPSO = Fuzzy Multi-Objective - Binary Particle Swarm Optimization

*FMO-BPSO-TA = Fuzzy Multi-Objective - Binary Particle Swarm Optimization - Threshold Accepting

*FMO-BFFO-TA = Fuzzy Multi-Objective - Binary Firefly Optimization - Threshold Accepting

Table 2.6: Count based on Hybrid EC Techniques

Hybrid Evolutionary Algorithms (Used for solving analytical and operational problems)	No. of Journals	No of Conferences / Book Chapters	Total
GP-ACO*	-	1	1
BPSO-TA*	-	1	1
BFFO-TA*	-	1	1
FF-TA*	-	1	1

*GP-ACO=Genetic Programming- Ant Colony Optimization

*BPSO-TA=Binary Particle Swarm Optimization- Threshold Accepting

*BFFO-TA=Binary Firefly Optimization- Threshold Accepting

*FF-TA=Firefly Optimization- Threshold Accepting

Table 2.7: Use of EC Techniques in Single Objective framework

Evolutionary Algorithms (Used for solving analytical and operational problems)	No. of Journals	No. of Conferences / Book Chapters
Genetic Algorithm	[62, 97, 111, 127, 165]	[35, 45, 228, 310]
Genetic Programming	[11, 131, 225]	[18, 21, 59, 88, 124, 194]
Genetic Network Programming	[319]	-
Grammar Guided Genetic Programming	[198]	-
Particle Swarm Optimization	[262, 263, 285, 286]	[68, 103, 284]
Ant Colony Optimization	[177, 287, 307]	[18]
Migrating Birds Optimization	-	[86]
Simulated Annealing	-	[83]
Threshold Accepting	-	[245]
Firefly Optimization	-	[245]
DMEL	[22]	-
Artificial Immune System	[117, 316]	[36, 102, 277]

Table 2.8: Use of Multi-Objective EC Techniques

Multi Objective Evolutionary Algorithms (Used for solving analytical and operational problems)	Journal articles	Conferences / Book Chapters
Multi-Objective Genetic Algorithm	-	[217]
NSGA-II	[198, 312]	-
SPEA-2	[198]	-
MO-BPSO (Association rule mining, MBA)	-	[103]
MO-PSO	[115]	[114]
MO-BPSO-TA (Association rule mining, MBA)	-	[103]
MO-BFFO-TA (Association rule mining, MBA)	-	[103]
FMO-BPSO	-	[244]
FMO-BPSO-TA	-	[244]
FMO-BFFO-TA	-	[244]

Table 2.9: Use of Hybrid EC Techniques

Name of the Hybrid EC Techniques (Used for solving analytical and operational problems)	Journal Articles	Conferences / Book Chapters
GP-ACO	-	[18]
BPSO-TA (Association rule mining, MBA)	-	[103]
BFFO-TA (Association rule mining, MBA)	-	[103]
FF-TA*	-	[245]

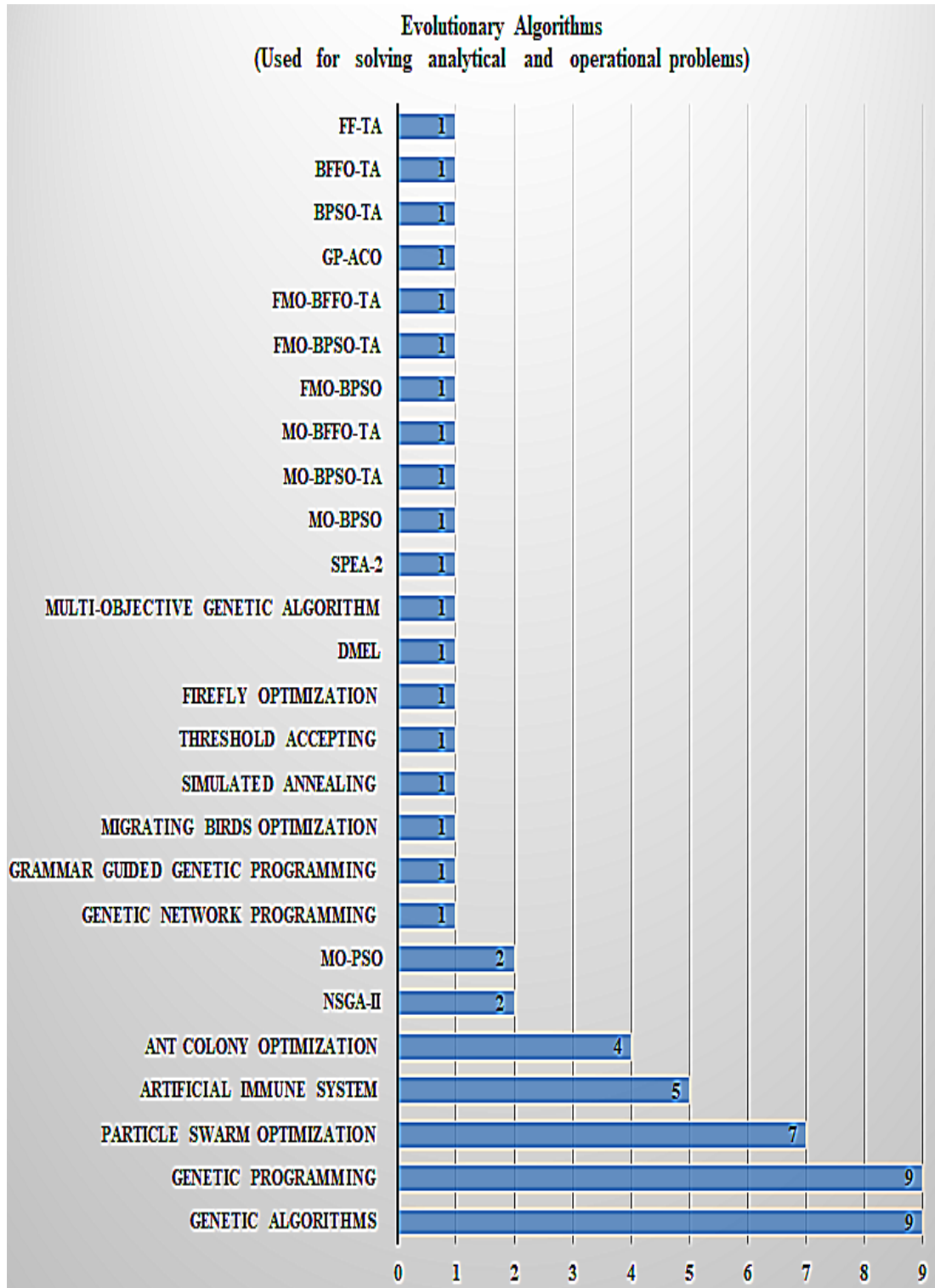


Figure 2.7: Bar Chart based on Count of all EC Techniques

Evolutionary Algorithms (Used for solving analytical and operational problems)

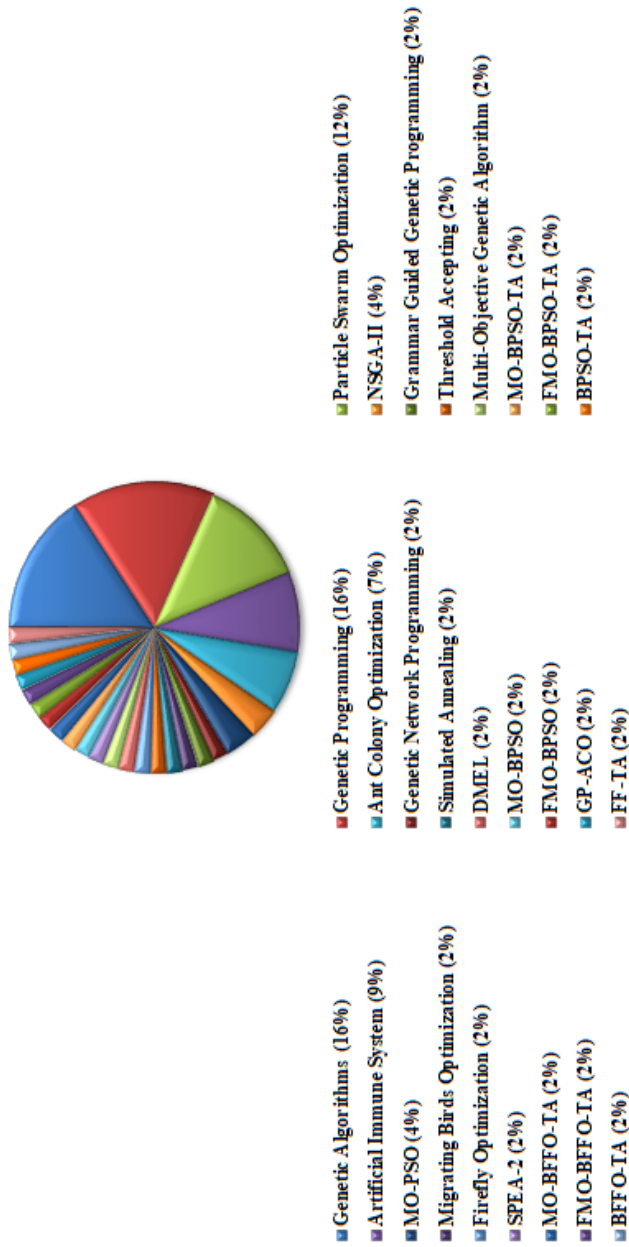


Figure 2.8: Use of all EC Techniques

2.4.4 Datasets used in reviewed articles

In the reviewed articles the followings are some statistics: Bank datasets were analyzed in ten 9 papers. German Credit Scoring dataset was analyzed in nine 12 papers. Australian Credit Scoring dataset was analyzed in five 7 papers. Credit Card datasets were analyzed in five 5 papers. All the details are summarised in Table 2.10.

Table 2.10: Datasets analyzed in articles

<i>Articles</i>	<i>Dataset Description</i>
[225]	German and Australian Credit Datasets
[131]	German and Australian Credit Datasets
[165]	Venezuelan Bank Data
[97]	Data from Equifax, Experion, and Trans Union Credit Bureaus Data
[11]	Egyptian Public Sector Banks Data
[111]	Data of Defaulting 6200 firms of Italy by the end of 2009
[127]	SEC's (Securities Exchange Commission) Data from May 1998 to March 2004
[177]	Data from National Health Insurance Research Database provided by National Health Insurance, Department of Health and National Health Research Institutes
[263]	XYZ Bank Dataset
[312]	German and Australian Credit Datasets
[22]	Credit Card Database / Social Database / PBX Database
[117]	Brazilian Bank Data
[198]	German Credit Dataset
[319]	German Credit Dataset
[68]	Major Bank in North Eastern Italy, the Bana Popolare di Vicenza
[194]	47,000 samples and 76 original Characteristics of Finance Enterprise
[45]	German Credit Dataset
[83]	German Credit Dataset
[18]	German and Australian Credit Datasets
[310]	German and Australian Credit Dataset/ Tunisian Bank Data/ HMEQ Datasets
[88]	Invertors Dataset
[35]	200 Company Data
[228]	Real Life Data
[21]	Brazilian Electronic Payment Company Data

Table 2.10: Datasets analyzed in articles (Continued)

[124]	Dow Jones Industrial Average (DJIA)- 2010 to end of 2013
[59]	Credit Card Data provided by Chinese Commercial Bank
[277]	POS Registered Transactional Records
[217]	Data from Credit Card Transactions of Retail Bank in North Africa
[86]	Original Credit Card Fraud Data(22 Million records with 978 as Fraudulent)
[316]	Major Australian Bank Data
[36]	Data Provided by WebBiz
[102]	Brazilian Card Issuer Data
[103]	XYZ Bank Dataset
[244]	XYZ Bank Dataset
[245]	Financial Statement Fraud Dataset
[62]	Data from Yahoo! Finance (10 companies of S&P 500 (2008))
[262]	German, Australian and Taiwan Credit Scoring Datasets
[114]	German and UK-1,2 Credit Scoring Datasets
[115]	German, Taiwan and UK-1,2 Credit Scoring Datasets
[307]	Australian Credit Scoring Dataset, AnalCat and Qualitative Bankruptcy Datasets
[287]	— No Dataset —
[286]	— No Dataset —
[285]	— No Dataset —
[284]	— No Dataset —

2.4.5 Trend

Figure 2.9 depicts the trend of the number of publications by year in journals, book chapters and conferences. The trend line marked by dotted line in the Figure 2.9 indicates that the trend is increasing starting from 1998 to 2020.

2.5 Review of Works in the Current Study

A brief overview of the research conducted in each of the analytical and operational problems with the help of EC techniques during the period of review is presented in the following subsections. There are no articles where EC techniques alone were employed

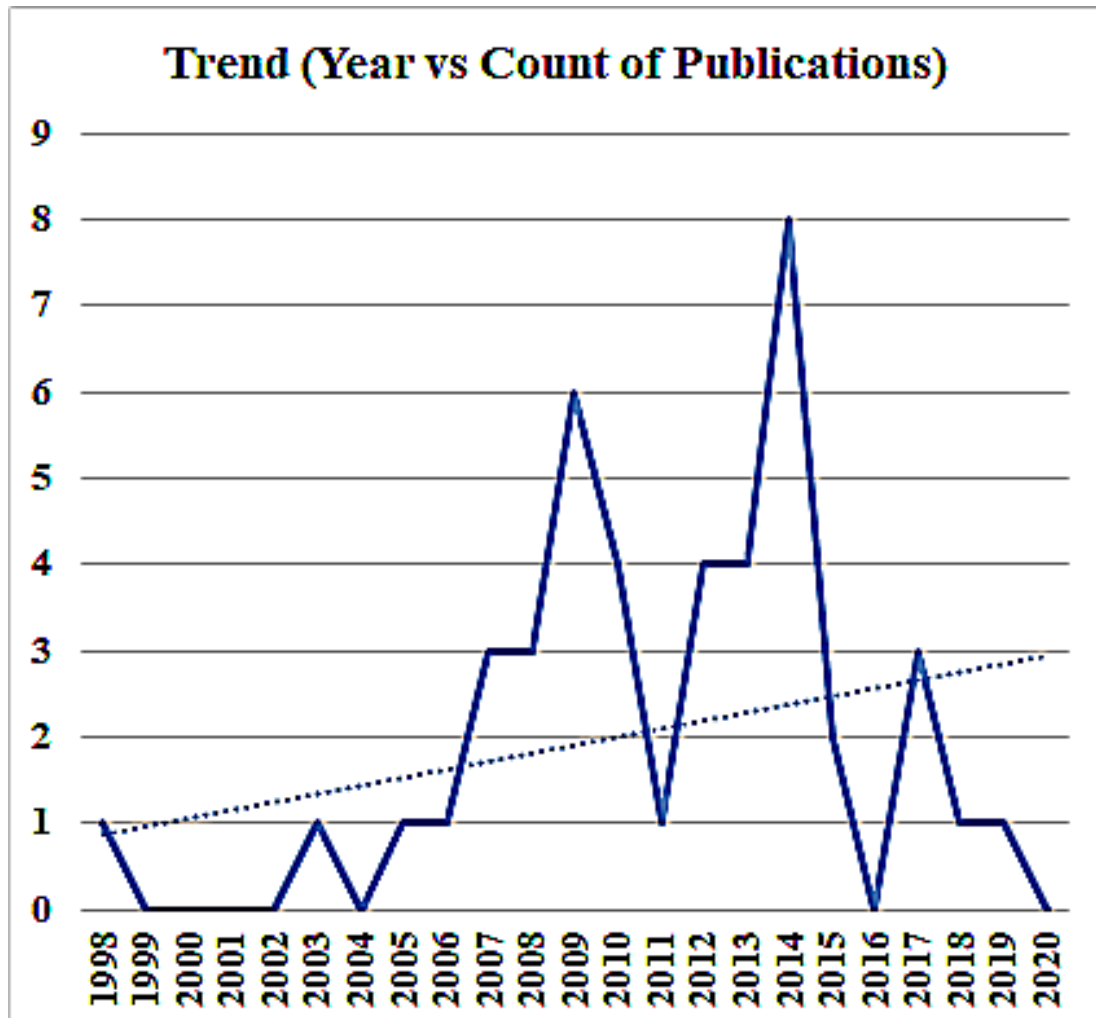


Figure 2.9: Trend of articles published per year

to address the problems of direct marketing, service marketing, customer campaigning and bank efficiency prediction. Therefore, no subsections were presented for them.

2.5.1 Credit Scoring

Financial service firms face an intriguing choice of whether to grant a loan to an applicant or not. The problem of determining the creditworthiness of a customer based on his/her data is called credit scoring. Credit scoring is a binary classification problem for which many traditional techniques exist, including both statistical and machine learning methods. However, of late, stand-alone evolutionary techniques have also been applied to solve scoring customers.

GP has attracted the attention for the following reasons (i) removal of irrelevant attributes (ii) determination of correct discriminant function (iii) used for both small and large datasets. So, [225] developed Credit Scoring models, using GP. Earlier, the same credit scoring models were developed using, Neural Networks, Decision Trees, Rough Sets and Logistic Regression. Australian credit scoring and German credit scoring datasets were used. The results obtained were better than those from earlier models. The GP-based model is better than statistical methods as there was no need for assumptions on the dataset. Also, the GP-based model is flexible and accurate.

According to [225], GP is better than the remaining models, and there is a need for intelligence rules for the decision-maker to understand the meaning of the dataset. For this purpose, Huang et al. in [131], proposed the two-stage GP for building a credit scoring model. The first stage of GP was employed to derive the IF-THEN rules for the decision maker. In the second phase of GP, the reduced dataset is applied in building the discriminant function for providing the capability of forecasting. The data sets used are the German Credit Scoring Data Set and the Australian Credit Scoring Data. The results obtained are better than those reported by [225]. Apart from finding discriminant function, IF-THEN rules have also been incorporated for even better accuracy compared to that of [225].

Behaviour Scoring is an important task in risk management. Chen et al. in [59] introduced Modified Genetic Programming (MGP) to solve behavior scoring problems on Credit Card Data provided by the Chinese Commercial Bank. MGP showed better accuracy for behavior scoring. MGP is compared with GP with normalized inputs (NGP) and back-propagation neural network (BPN). MGP performs well for small

datasets. In future, MGP can be combined with other artificial intelligence techniques in order to classify based on rules generated. The behavior of the customers who had defaulted, the timing of the default and feature selection can also be investigated.

Derived Characteristics are usually regarded as the important index in credit scoring, and the selection of derived characteristics is formulated as an optimization problem. GP and human intervention are applied to obtain derived characteristics. Liu et al. in [194] applied GP for selecting derived characteristics that are important in credit scoring. GP utilized 47,000 samples and 76 original characteristics of Finance Enterprise, and the outcomes are satisfactory. Later, discrimination analysis is used on the set of valuable derived characteristics, and the results turned out to be good.

A comparison, between conventional methods and non-conventional ones, has also been carried out in [11] where GP was implemented for Credit Scoring with a case of Egyptian Public Sector Banks. The GP model was compared with the weight of evidence measure and the probit models which are the conventional methods for credit scoring. The average correct classification (ACC) was high and expected cost of misclassification (ECM) was lower than those of other models. This study can further include mortgages, house loans, and corporate loans. Also, employing other evaluation criteria, such as Gini coefficient and ROC curve, would be useful.

Scoring models have been developed to model/predict credit risk. These scoring models are categorized based on statistical methods like logistic regression, classification trees, nonlinear regression, nearest neighbor approach, etc., and non-statistical methods like GA, neural networks, etc. Cai et al. in [45] solved the personal credit scoring problem with GA using the German Credit Dataset (<http://archive.ics.uci.edu/ml/>), and their model proved to be of high efficiency when evaluating the significance values. But, GA is more complicated than other methods, and its computing time is also high. Nevertheless, it can be optimized further to reduce computational time and increase efficiency.

SA is a local search meta-heuristic that can be used for feature selection, classification, and clustering. Dong et al. in [83] presented the SA-based rule extraction algorithm for credit scoring and performed experiments on German credit dataset. The SA-based model outperformed statistical techniques like Decision Trees (DT), K-nearest neighbor, linear discriminant analysis (LDA), and quadratic discriminant analysis (QDA). The benefits of the proposed model are that it can extract rules as

well as find attribute correlations. Its limitations are that the parameters have to be fine-tuned and training time is high.

Credit Scoring can also be performed applying Multi-Objective Evolutionary Algorithm (MOEA) by proposing new criteria for feature selection. Wang and Huang in [312] employed feature selection techniques along with MOEA feature selection with relative correlations. Then, various classifiers such as SVM, multi-layer perceptron are used on these features. German and Australian credit datasets have been analyzed. MOEA yielded superior AUC than other feature selection methods. This work can be extended by applying data pre-processing and feature selection for various classifiers on large-scale problems and also new multi-objective algorithms incorporating various performance metrics can also be considered.

In large organizations, it is very difficult to update continuously complex credit scoring systems and it is better to automate the scoring maintenance functions within firms. To this end, Fogarty in [97] developed maintenance functions based on the credit scoring system using GA. The scorecard prepared using GA performed as well as or, if not better than the traditional methods. GA is used to refine the score created by the initial model driven by traditional techniques. Data from Equifax, Experion and Trans Union Credit Bureaus is collected and used. Marketing models have the highest improvement on an auto loan in the Pacific Rim region and risk models also have an improvement for the credit cards of the Pacific Rim region. Limitations are that GA needs to have a lot of computational power, data, and clear definition of objective function for the business. Also, there is no generic model for all problems and data. Further merit of GA over other search techniques has to be investigated.

Here, we discuss a hybrid, GP + ACO, for credit scoring. Aliehyaei and Khan in [18] investigated GP, the modified ACO, and hybrid GP-ACO for credit scoring. German and Australian credit datasets were analyzed, and the hybrid approach was shown to perform the best of the other three methods. In this hybrid, the insight of ACO is input to GP for a head start. Further investigation into the better performance of ACO and impact of improving GP were to be carried out.

Disaggregation based on choice is a concept based on simple findings that, in general, real-world decision makers are unable or unwilling to provide. Corazza et al. in [68] proposed a PSO based choice disaggregation in the multi-criteria credit worthiness of a set of firms. The data set was provided by the major bank in North Eastern Italy

and the results obtained were highly consistent. The comparison was made with actual parameters of MULTicriteria RAnking MEthod (MURAME) and parameters obtained by PSO for ordering of the alternatives. Further, research using realistic computational time and finding the values of parameters of the MURAME model that make them consistent with the classification provided by the banks for credit scoring have to be performed.

Waad et al. in [310] employed the GA-based feature selection technique to the credit scoring data of German and Australian Credit Datasets, Tunisian Bank Dataset, and the HMEQ Dataset. They concluded that the proposed technique is robust and accurate, on performance metrics like precision, recall, and F-measure.

Parameter range estimation, polynomial fitness functions, and an alternate fitness function called weighted bitmask were applied to GA for building credit scoring models in [165]. A sample of 489 credit card customers' data from Venezuela was used. The sensitivity, specificity, and accuracy of the model with three fitness functions were compared. The weighted bitmask fitness function yielded better results than others as indicated by a statistical test. The major limitations are the availability of a quality database and the computational complexity of the maximum bitmask size. Different encoding schemes can also be used to reduce computational complexity.

Uthayakumar et al. [307] presented an ACO-driven rule-based classification system for credit scoring. They performed their analysis on both subjective and quantitative datasets. A multi-objective credit scoring model is proposed, namely, multi-objective PSO in [114, 115] and compared with linear discriminant analysis (LDA). The Gustafson-Kessel (GK) calculation is the most important tool for credit scoring. Nevertheless, this calculation exhibits a moderately poor capacity to distinguish a subset of features from a large dataset. Most techniques that use the GK calculation require a predefined number of clusters. In [262], GK-based modified binary particle swarm optimization (MBPSO) technique is developed to increase the classification accuracy. In [109], Evolutionary Algorithms (EA)-based models proposed during 1997 to 2018 for credit card scoring are reviewed. The focus in [109] is based on two main aspects: (i) model type with issues addressed and (ii) assessment procedures. Al-Sahaf et al. in [16], presented a survey on the applications of evolutionary machine learning, for example, classification, regression and clustering on emerging areas including combinatorial optimization, computer vision, deep learning, transfer learning, and ensemble learning.

2.5.2 Customer Segmentation

Customer Segmentation is a major strategy that entails the grouping of customers into homogeneous clusters so that products and services can be customized or tailor-made for a given cluster of customers, thereby reducing huge operational costs. Here the assumption is one size does not fit all, which is very well proven in the real-world applications. Hence, the need of clustering.

Several studies demonstrate that segmentation based on customer lifetime value (CLV), rather than descriptive variables alone, presents interesting features for marketing. Mzoughia and Limam in [217] proposed a multi-objective GA considering descriptive variables and CLV for segmentation, using the data from credit card transactions of a Retail Bank in North Africa. The proposed method characterized segments well. Finding the optimal number of clusters is difficult and needs investigation. Also, the influence of the multi-objective optimization algorithm is to be investigated.

2.5.3 Fraud Detection

Fraud is characterized as an activity that is undesired, not allowed by law, and illegal. Only a minuscule percent of the activities/transactions are illegal but can have a catastrophic impact. Fraud detection typically is a data mining problem. Data imbalance is the issue in fraud detection, where genuine transactions disproportionately outnumber the fraudulent one, posing a problem for almost all machine learning techniques.

GAs are evolutionary algorithms that aim to obtain better solutions and can also be used in detecting fraud. Hoogs et al. in [127] proposed a GA approach to detect financial statement fraud. The study comprised the target sample (fraudulent companies) of 51 companies accused by Security and Exchange Commission of improperly recognizing revenue and peer class (non-fraudulent companies) of 339 companies. GA accurately classified 63 percent of target class companies and 95 percent of peer class companies. Research can be extended to other financial statement frauds. The techniques presented here can also be extended to incorporate other qualitative indicators and events that have additional discriminative power. Further, research could also test the stability of learned patterns on a selected sample of test patterns at a later time frame than the training sample. Moreover, prediction accuracy for fraudulent companies can be increased by resorting to multi-objective and hybrid EC techniques.

Fraud detection costs are minimized if the parameters are tuned with an EC technique before applying AIS. Gadi et al. in [102] applied AIS with parameters optimized using GA for credit card fraud detection and compared its performance with that of other DM techniques. Credit card data was provided by a Brazilian card issuer, and the results were consistent with earlier applied DM techniques. Regarding future directions, we can analyze the details of optimized parameters and choose a better metaheuristic for parameter tuning.

An important point is that the problem of fraud is highly dynamic as fraudsters continually adapt. So, in some studies, AIS is preferred over other techniques. Brabazon et al. in [36], suggested the use of AIS for online credit card fraud detection. The data was provided by WebBiz, and the results suggest that AIS is good for fraud detection. The system obtained high accuracy but with high false negatives. Also, parameters can be tuned using GA, and a hybrid or multi-stage AIS detection (Negative Selection, Modified Negative Selection, and Clonal Selection) can also be considered.

Nowadays fraud analytical solutions that analyze customer behavior patterns (apart from technical solutions like cyber-security driven ones) are utilized for detecting fraud. Ozçelik et al. in [228] applied GA to improve credit card fraud detection utilizing real life data. The results are better than standard DM techniques with appropriately chosen parameter values.

Fraud detection methods are based on two approaches: 1) User behavior analysis and 2) fraud analysis. Soltani et al. in [277] presented AIS based credit card fraud detection called Artificial Immune Recognition System that considers user behavior. Data from POS registered transactional records was analyzed, and it was observed that the false positive rate decreased.

Wong et al. in [316] discovered credit card fraud by the AIS. Desensitised credit card transaction data obtained from a major Australian bank consisting of a total of 640,361 transactions for 21,746 credit cards. The proposed system achieved a detection rate of 71.3 percent on real transaction data. Many avenues of research are possible, such as the distribution of components of AIS, improvements to AIS, more sophisticated match rules, integration of AIS to transaction processing and even trying out other evolutionary techniques in order to improve the detection rate.

Research is underway for applying new meta-heuristics other than GA and GP for fraud detection. Duman and Elikucuk in [86] proposed an MBO algorithm to credit card

fraud detection. MBO analyzed original credit card fraud data (22 million records with 978 as fraudulent ones) and yielded good performance compared to GA and standard DM techniques. As an extension, MBO can be further improved by testing alternative neighborhood functions and alternative benefits mechanisms.

In online payments, if a transaction is fraudulent, then the company absorbs the loss and consequently refunds the lost amount to the customer concerned, provided the customer is not the perpetrator of the fraud. Assis et al. in [21] identified fraud in online credit card transactions using GP on the Brazilian Electronic Payment Company Data. The algorithm achieved good performance in fraud detection. In future studies, multiple objectives should be considered, including rule complexity, higher accuracy, and additional datasets.

Fraud detection should have a high detection rate, low false alarm rate, and real-time responses. So, Soltani Halvaie and Akbari in [117] detected credit card fraud using an artificial immune-based recognition system (AIRS). The Brazilian Bank Data was used for this paper. It had 3.74% fraudulent data of lost cards, stolen cards, skimming, mail orders, telephone orders, and account takeover. The accuracy increased up to 25% while cost was reduced up to 85% and response time decreased by 40%. Also, improvements can be made like weighting the features using feature selection methods in distance function, considering artificial immune networks for fraud detection (AFDM) using cloud computing.

2.5.4 Market Basket Analysis

Market Basket Analysis (MBA) is the analysis of items that are bought (or present) together by many customers in single, multiple or sequential transactions. Understanding the connections and the quality of those connections provides important knowledge that can be utilized to make suggestions, cross-sell, up-sell, offer coupons, and so forth.

Constraint-Based Mining (CBM) concentrates on mining item sets that are important. Kuo and Shih in [177] employed ACO to mine an extensive database to find the association rules efficiently. Considering multi-dimensional constraints was observed to be more efficient. The National Health Insurance Research Database was considered, and ACO yielded better results than Apriori and consumed less computational time. Performing feature selection improves the quality of the rules.

Associative Classifiers are classifiers based on associative classification rules (ACR) and are more accurate than traditional classification. However, they cannot handle numerical data. Chien and Chen in [62] built an associative classifier to discover trading rules from a GA-based algorithm on numerical data. Data from ten companies of the S&P 500 from Yahoo! Finance from the year 2008 was analyzed. The proposed model has high prediction accuracy and is highly competitive compared to the data distribution method. Further, the proposed model (GA-ACR) could be extended to more relations between numerical data and can be utilized to deal with portfolio investment problems.

Ranking of rules and classifying rules as good or bad is important after generating association rules. Yang et al. in [319] suggested the evolutionary associative classification method for both adjustments of the order of rules as well as the refinement of each single rule. The association rules are re-ranked with respect to the equations which combine support and confidence values. GNP is used to search the equation space for prior knowledge. Apart from global ranking, the local adjustment is made by giving some rewards to the right rules and penalty to bad rules. The German Credit dataset was analyzed and the proposed method improves classification accuracies.

Luna et al. in [198] extracted association rules using G3P models that enabled extraction of both numerical and nominal association rules in a single step. Proposals combine G3P with Non-dominated Sort Genetic Algorithm (NSGA-II) and Strength Pareto Evolutionary Algorithm (SPEA-2). Credit dataset was used, and a good trade-off between support and lift was obtained by the proposed multi-objective methods. Proposed methods work on any domain whether categorical or nominal. They also restrict the search space using grammar.

Sarath and Ravi in [263] applied PSO to mine M best association rules without having to specify thresholds on support and confidence. Unlike the Apriori and FP-growth algorithm, this algorithm generates the best M non-redundant rules from a given bank dataset. The quality of rules is assessed by a fitness function viz., the product of support and confidence. This algorithm can be used as an alternative to the Apriori and FP-growth algorithm. Future strategies include the usage of better fitness function.

Hybrid Multi-Objective techniques can also be applied to rule generation. Ganghishetti and Vadlamani in [103] developed three multi-objective evolutionary association rule

miners, namely MO-BPSO, MO-BPSO-TA, and MO-BFFO-TA on XYZ Bank datasets and concluded that MO-BPSO-TA outperformed other models proposed. All three consumed less time compared to standard association rule mining algorithms and do not generate redundant rules. In another work [244], Fuzzy multi-objective evolutionary association rule miners are developed, namely FMO-BPSO, FMO-BPSO-TA, and FMO-BFFO-TA on XYZ Bank datasets

2.5.5 Customer Life Time Value

Customer Lifetime Value (CLV) is a forecast of all the worth a business will get from their whole association with a customer right from the time he/she started doing business until the time he/she attrite. Since we do not know exactly the extent of each relationship with any customer, we would like to make a decent estimate and state CLV by a periodic estimate. Not much work is reported in this important aspect, which has tremendous significance in growing the business.

Mzoughia and Limam in [217] discussed the estimation of CLV along with other analytical problem taking the help of various EC techniques and these are reviewed in other subsections. Hence, in this subsection, we review the work where only CLV is estimated with an EC technique, i.e., GA.

2.5.6 Customer Churn Prediction

The ability to foresee that a specific customer is about to churn, while there is still time to retain him/her results in a potential extra income for every business. Other than the immediate loss of income that results from a customer defecting to the competition, the customer acquisition cost is a lost investment. Further, it is consistently more expensive to acquire a customer than to retain an existing one. Therefore, customer churn detection becomes a quite an essential task in analytical domain with significant revenue ramifications. Using data and text mining techniques, a service firm can predict which set of customers has a high probability to churn. The activity of customer retention involves marketing teams reaching out to the list of identified dissatisfied customers (obtained through churn models) in order to retain them.

Eiben et al. in [88] employed and evaluated the GP model on the problem of customer retention modeling using the investors' dataset of a financial company. Other

data analysis techniques like the rough set, CHAID, and logistic regression were also applied independently, but GP outperformed all of the others. EC techniques other than GP are yet to be studied. Developing models for other time periods is an ongoing research.

Au et al. in [22] proposed a novel evolutionary data mining algorithm called DMEL for churn prediction. Some databases like Credit Card Database, Social Database, and PBX Database were used. Experiments with different data sets revealed that DMEL effectively discovered interesting and robust classification rules. Feature selection can be considered as an important future research direction here.

2.5.7 Default Prediction

The default detection models make use of the present and historical data of a customer to forecast the customer's capacity to pay back on time. An accurate default detection system is essential for the company's profitability. Here, one should realize that both retail and corporate customers can default. Very few works appeared in this area.

Classification rules can be set by using GA, just like discriminative analysis for default forecasting. Bozsik in [35] developed a GA-based default forecasting model on data from 200 companies and compared it with the models currently in use. The results obtained were reliable and achieved a classification accuracy of 84.5%. A further bigger sample of the dataset can be used in the model. Hybrid EC techniques can be used, whereas formulating the problem in a multi-objective framework is also an important future research direction.

It is important to pick up the signs of financial distress on time to evaluate firms that are going to default. Gordini in [111] proposed a GA-based approach for default prediction of small enterprises. Data from defaulting 6200 firms of Italy by the end of 2009 was analysed. Results obtained are the best in terms of predicting defaulting firms. Models other than GA that capture the non-linear relationship and are capable of extracting rules can also be used.

2.5.8 Key Generation

Swarm intelligence [67, 151] based key generation algorithm was created through an ant colony optimization [286]. In [285], the particle swarm optimization based key

generation is discussed. Both utilized a single objective function that is dependent on the number of matches of keys in key streams with those in plain text. Authors in [284, 287] used the ant colony based key generation algorithm for the image encryption and in applications of cellular networks.

Tables 2.11 and 2.12 summarizes the efficacy of the EC techniques compared to conventional techniques in solve the analytical problems. Table 2.12 presents the performance of association rule mining using various measures of evaluation like support, confidence, comprehensibility, interestingness, and the number of rules considered for a mining task. Table 2.13 presents the objective functions, their properties and parameters that are set in each research article.

Table 2.11: Results of Classification Problems

Articles	Accuracy (EC Method)	Sensitivity (EC Method)	Specificity (EC Method)	Compared (Standard DM Technique)	Dataset Type	Benchmark Dataset	Rule Gener- ated (Mentioned in the Ar- ticle)
[165]	75.85%	85.28%	53.85%	-	Bank	No	No
[45]	-	91.18%	56.28%	-	Credit Scoring	Yes	No

Table 2.11: Results of Classification Problems (Continued)

[225]	88.27%, 77.34% (GP) (Australian, German)	-	-	MLP* (87.93, 75.51), CART* (85.81, 7059), C4.5 (87.06, 73.17), Rough Sets (83.72, 74.57), LR* (86.19, 75.40)	Credit Scoring	Yes	Yes

Table 2.11: Results of Classification Problems (Continued)

[131]	79.49% (2SGP)	-	-	MLP* 75.51, CART* 70.59, C4.5 73.17, Rough Sets 74.57, LR* 75.40	Credit Scoring	Yes	Yes

Table 2.11: Results of Classification Problems (Continued)

[11]	83.28%, 85.82% (GP _p , GP _t)	91.89%, 91.07% (GP _p , GP _t)	65.25%, 74.94% (GP _p , GP _t)	WOE _a * (54.99, 98.30, 34.08) WOE ₁ * (54.44, 98.30, 33.25), PA * (81.93, 67.40, 88.95), PA ₁ * (81.62, 67.15, 88.60)	Bank	No	No
[59]	89.83% (MGP), 88.73% (Normalized Inputs GP (NGP))	-	-	86.76% (Back Propagation Neural Net- work (BPN))	Bank	No	Yes

Table 2.11: Results of Classification Problems (Continued)

[312]	100% (MOEA with LDA*)	-	-	92.5%(Chi-Square), 99.3% (EAFS*), 92.0% (None), 99.4% (Relief) With LDA*	Credit Scoring (Austral- ian)	Yes	No

Table 2.11: Results of Classification Problems (Continued)

[83]	72.9% (SAREA)	91.57% (SAREA)	29.3% (SAREA)	LDA* (72.1%, 73.3%, 68.75%) QDA* (65.9%, 64.9%, 67.5%) KNN* (71.6%, 98.0%, 7.66%) C4.5 (70.4%, 78.6%, 49.9%)	Credit Scoring	Yes	Yes
[18]	77.5%	-	-	-	Credit Scoring	Yes	Yes
[86]	-	82.78% (Average) (MBO)	-	82.73% (Average) (GASS*)	Bank	No	No
[277]	-	91%, 100%	33%, 13%	-	Retail	No	No

Table 2.11: Results of Classification Problems (Continued)

[36]	90.14%	96.55% Max. of (Sen. + Spec)/2	9.35% Max. of (Sen. + Spec)/2	-	Bank	No	No
[319]	73.56% (RANK+) , 73.10% (RANK)	-	-	70.90% (FOIL*), 69.10% (CPAR*), 72.00% (CMAR*)	Credit Scoring	Yes	Yes

Table 2.11: Results of Classification Problems (Continued)

[22]	95.67%, 99.9% (DMEL)	-	-	82.6%, 94.6% (C4.5), 58.9%, 58.9% (Michigan Style (SCS*)), 59.6%, 94.2% (Pittsburgh Style (GABL*))	Bank, Company	Yes	Yes
[111]	76.9% (GA)	-	-	71.3% (MDA*), 72.1%(LR*)	Company	No	Yes
[35]	84.5%	-	-	-	Company	No	Yes

* MLP = Multi-Layer Perceptron

* WOE = Weight of Evidence

* CART = Classification and Regression Tree

-
- *PA = Probit Analysis
 - *LR = Logistic Regression
 - *EAFS = Evolutionary Algorithm for Feature Selection
 - *FOIL = First Order Inductive Learner
 - *LDA = Linear Discriminant Analysis
 - *CPAR = Classification Based on Predictive Association Rule
 - *QDA = Quadratic Discriminant Analysis
 - *CMAR = Classification Based on Multiple-class Association Rules
 - *MDA = Multiple Discriminant Analysis
 - *KNN = K-Nearest Neighbour
 - *SCS = Simple Classifier System
 - *GASS = Genetic Algorithm hybridized with Scatter Search
 - *GABL = GA Batch Concept Learner

Table 2.12: Results of Association Rule Mining (M.B.A.)

Articles	Support (EC Method)	Confidence (EC Method)	Comprehensibility (EC Method)	Interestingness (EC Method)	No. of Rules (EC Method)	Compared (Standard DM Technique)
[62] I (Associative Classification)	1%, 2%, 5%, 10%	60%, 70%, 80%, 90%	-	-	Top 3, 4, 5 rules taken with 90% Accuracy	-
[62] (II) (GA Used)	-	-	-	-	Top 3 rules from popula- tion of 20 (GA Method used) with 85% Accuracy	Data Dis- tribution Method (DDM)
[198]	70% (G3PARM)	90% (G3PARM)	-	-	Population Size of 50	-
[319]	1% (GNP Pa- rameters)	50% (GNP Pa- rameters)	-	-	Max. no. of rules is 80,000	FOIL*, CPAR*, CMAR*
[177]	-	-	-	-	Pheromone Threshold of 15 to generate 10 rules	Apriori
[263]	1.5%	40%	-	-	10	Apriori, FP-Growth*
[103] (MO- BPSO)	2.04%	54.25%	68.32%	20.95%	-	-

Table 2.12: Results of Association Rule Mining (M.B.A.) (Continued)

[103] (MO- BPSO- TA)	2.15%	53.79%	67.94%	18.41%	-	-
[103] (MO- BFFO- TA)	1.11%	28.62%	64.71%	6.93%	-	-

*FOIL = First Order Inductive Learner

*CPAR = Classification Based on Predictive Association Rule

*CMAR = Classification Based on Multiple-class Association Rules

*FP-Growth= Frequent Pattern-Growth

Table 2.13: Objective Functions, Properties of objective function and Parameters set for the EC techniques

<i>Articles</i>	<i>EC Technique</i>	<i>Objective Function Used</i>	<i>(Single- or Multi-Objective) / (Maximization or Minimization) / (Discrete or Continuous) / (Constrained or Unconstrained) / (Unimodal or Multimodal)</i>	<i>Parameter Set</i>	<i>Values</i>
[225]	GP	Mean Absolute Error	Single / Minimization / Discrete / Unconstrained / NM	Population Size (PS) = 40, Crossover Rate (CR) = 0.9, Mutation Rate (MR) = 0.01, Maximum Generations (MG) = 100	
[131]	Two Stage - GP	Mean Absolute Error	Single / Minimization / NM / Unconstrained / NM	PS = 100, CR = 0.9, MR = 0.01, MG = 1000	
[165]	GA	Fitness Function = Weighted Bit mask & Fitness Score = Accuracy	Single / Maximization / Discrete / NM / NM	PS = 200, CR = 0.8, MR = 0.15	
[97]	GA	Scoring System Maintenance Function	NM / NM / NM / NM / NM	NM	
[11]	GP	α (Sum of square error) β (Classification Error)	Single / Minimization / NM / Unconstrained / NM	CR = 50%, MR = 95%	

Table 2.13: Objective Functions, Properties of objective function and Parameters set for the EC techniques (Continued)

[111]	GA	Accuracy	Single / Maximization / NM / NM / Multimodal	NM
[127]	GA	Accuracy	Single / Maximization / NM / NM / Multimodal	NM
[177]	ACO	Support and Frequency	Single / Maximization / Discrete / Constrained / Multimodal	Pheromone Threshold = 15
[263]	BPSO	Support $\times$ Confidence	Single / Maximization / Discrete / Unconstrained / Multimodal	NM
[312]	NSGA-II	Chi-square 2. Relative correlation 3. Relief	Multi / NM / NM / Unconstrained / Multimodal	PS = 50, CR = 0.9, MR = 0.01, MG = 100 and 1000
[22]	DMEL	Probability of attribute values of the record that can be correctly determined by the rule.	Single / Maximization / Discrete / NM / Multimodal	CR = 0.25; 0.5; 0.75
[117]	AIS	Fraud Score based on 1. distance function and 2. cost	Single / Minimization / NM / NM / NM	AIS
[198]	G3P, G3P-NSGA-II, G3P-SPEA-2	Cartesian Distance 2. Support 3. Confidence	Single and Multi / NM / Discrete / Constrained / Multimodal	PS = 50, CR = 90%, MR = 15%, MG = 0.1, Support = 70%, Confidence = 90%, Pareto Front Size = 20

Table 2.13: Objective Functions, Properties of objective function and Parameters set for the EC techniques (Continued)

[319]	GNP	Accuracy of rules	Single / Maximization / Discrete / NM / NM	PS = 5, CR = 0.2, MR = 0.1, MG = 5, Support = 0.01, Confidence = 0.5, Maximum antecedent size = 3, Maximum no. of rules = 80000
[68]	PSO	Spearman rank correlation coefficient and 2. Net Flow	Single / Minimization / Continuous / Unconstrained / NM	No. of particles = 100; 200
[194]	GP	Information Value	Single / Maximization / Discrete / Unconstrained / NM	NM
[45]	GA	Error rates of judging rightly and wrongly good and bad customers	Single / Maximization / Discrete / Unconstrained / NM	PS = 1000, CR = 0.5, MR = 0.01, MG = 1000
[83]	SA	Percentage correctly classified $\times$ Similarity Function $\times e$ (e is a coefficient)	Single / Maximization / Discrete / Unconstrained / NM	Initial Temperature = 10000, Final Temperature = 0.01, Cooling rate = 0.95, Iteration at each temperature = 100
[18]	GP, ACO, GP+ACO	Mean Absolute Error	Single / Minimization / NM / Unconstrained / NM	NM

Table 2.13: Objective Functions, Properties of objective function and Parameters set for the EC techniques (Continued)

[310]	GA	$\sum_i Weight_i \times Distance(\sigma, L_i)$	Single / Maximization / Discrete / NM / NM	NM
[88]	GP	Coefficient of Concordance (distinctive power of model)	Single / Maximization / NM / NM / NM	CR = 0.5, MR = 0.9
[35]	GA	<i>Discriminant Function</i> = $\sum_i W_i \times x_i$ <i>W_i is the weight and x_i is the attribute value</i>	Single / Maximization / Continuous / Unconstrained / NM	MR=20%
[228]	GA	Correctly Classified transactions	Single / Maximization / NM / NM / NM	NM
[21]	GP	Accuracy $\times$ Sensitivity $\times$ Specificity	Single / Maximization / NM / NM / NM	PS = 500, CR = 75%, MR = 2.5%, MG = 100
[124]	GA	Optimal rules based on classification (Investment Rules)	Single / Minimization / Discrete / Constrained / NM	PS = 100, Elitist = 10
[59]	GP	Sensitivity + Specificity	Single / Maximization / Discrete / Unconstrained / NM	PS = 1000, CR = 0.9, MR = 0.1, MG = 50
[277]	AIS	Fraud Ratio	Single / Maximization / NM / NM / NM	NM
[217]	MOGA	1. No. of transactions and 2. CLTV	Multi / Maximization / NM / NM / NM	NM
[86]	MBO	Savings limit and true positive rate	Single / Maximization / NM / NM / NM	No. of birds = 15, No. of neighbors = 25, No. of flaps = 100, No. of overlaps = 5,

Table 2.13: Objective Functions, Properties of objective function and Parameters set for the EC techniques (Continued)

[316]	AIS	Detection rate, False positive rate, fraud ratio	Single / Maximization / NM / NM / NM	NM
[36]	AIS	Hetogeneous value distance metric	Single / Maximization / Continuous / NM / NM	NM
[102]	AIS, GA	Cost = \$100 $\times$ false negative + \$10 $\times$ false positive + \$1 $\times$ true positive	Single / Maximization / NM / NM / NM	Parameters of AIS optimized using GA
[103]	MO-BPSO, MO-BFFO-TA, MO-BPSO-TA	$\prod_{i=1}^8 Measurevalue(i)$ Support, Confidence, Interestingness, Comprehensibility, Lift, Leverage, Conviction, Coverage	Multi / Maximization / Discrete / Unconstrained / NM	NM
[62]	GA	1. Confidence 2. Support $\times$ Confidence 3. $\sqrt{Support \times Confidence}$ 4. Confidence $\times \log(Support)$	Single / Maximization / Discrete / NM / Multimodal	PS = 20, MG = 50, CR = 0.7, MR = 0.001, Minimum Support = 0.002, Minimum Confidence = 70%, Top K rules; K = 3

Table 2.13: Objective Functions, Properties of objective function and Parameters set for the EC techniques (Continued)

[244]	FMO-BPSO, FMO-BPSOTA, FMO-BFFOTA	1) Support, 2) Confidence, 3) Coverage, 4) Comprehensibility, 5) Leverage, 6) Interestiness, 7) Lift and 8) Conviction are	Multi / - / Discrete / NM / Multimodal	1) Max. iteration FMO-BPSO = 50-100, 2) Max. iterations of BPSO; inner and outer iterations of TA; FMO-BPSOTA = 50-200, 100-200, 25, 3) Max. iterations of BFFO; inner and outer iterations of TA; FMO-BFFOTA = 50-100, 50-300, 5
[245]	FFTA Rule Miner, FF Rule Miner, TA Rule Miner	Sensitivity $\times$ Specificity	Single / Maximization / - / NM / Multimodal	Max. iteration FF-TA = 25 - 200 Max. iteration FF = 100, Max. iteration inner and outer iterations TA = 250, 25
[262]	BPSO	Kernel Fuzzy Clustering Method	Single / Minimization / Discrete / NM / Multimodal	Iteration = 40, Maximum iterations = 300
[114]	MOPSO	1) Minimize number of misclassifications 2) Minimize the sum of deviation of observations 3) Maximize the minimal distances from critical values	Multi / - / - / NM / Multimodal	$W_G = 0.5$, $W_B = 0.5$, Cutoff = -1.1, 80, -1.1 Generations = 1000 and Population = 40

Table 2.13: Objective Functions, Properties of objective function and Parameters set for the EC techniques (Continued)

[115]	MOPSO	1) Minimize the sum of distances between the score of each sample misclassified and cutoff 2) Maximize the sum of distances between the score of each sample correctly and cutoff	Multi / - / - / NM / Multimodal	$W_G = 0.33-0.5$, $W_B = 0.5-0.7$, Cutoff = Variable or -1.1, T= 100 or 1000 and Population = 40
[307]	ACO	Sensitivity $\times$ Specificity	Single / Maximization / NM / NM / Multimodal	Number of Ants = 6
[284]	AKGA	Number of Characters in the keystream occurring in the plain text divided by keystream length	Single / Maximization / Discrete / NM / Multimodal	-
[285]	AKGA, PKGA	- do -	- do -	-
[286]	AKGA	- do -	- do -	-
[287]	AKGA	- do -	- do -	-

*NM = Not Mentioned

2.6 Discussion of the Review and Future Directions

While EC techniques have made steady inroads into the area of analytical and operational domains, they have not done as much as they have in other fields of application. Our survey indicates that analytical and operational domain still offers fertile ground, where EC techniques either in isolation or in combination have a tremendous role to play. In this context, we listed some important open research problems and future directions, as follows:

- The most striking observation from the review is that EC techniques turn out to be viable alternatives for classical DM techniques like Neural Networks, Support Vector Machines, Decision Trees, K-means, etc. as sometimes they offer better results in less time.
- As the trend of applying EC techniques to solve analytical and operational problems has been increasing from 1998-2020, it will be a booming area of research in future.
- EC techniques solve these tasks by generating ‘if-then’ rules from the underlying data. These rules can be considered as early warning expert systems in tasks such as fraud detection, churn detection, default detection, credit scoring, etc. In this sense, EC techniques, are termed as ‘transparent systems’ as opposed to black boxes like logistic regression, MLP, SVM, etc. Decision trees, rule sets, rough set based methods, fuzzy logic based methods, fall in the category of transparent systems. However, in many of the papers reviewed here, quite surprisingly, these rules are not at all presented making it difficult to judge the value of the contributions.
- It is conspicuous from the survey that for some reason, GA has been employed extensively for solving analytical and operational problems. Apart from GA, other EC techniques that are proved to be better in dealing with multi-modal, non-linear, and non-differential objective functions and non-convex search spaces can also be employed. For instance, DE, which outperformed GA in many problems in other domains, can be employed to investigate its potential for solving analytical and operational problems. Likewise, the potential of other EC techniques can also be explored.

-
- Further, hybrid EC techniques have immense potential in solving these tasks in less time with higher accuracies. However, the challenge is the correct choice of constituent EC techniques and the suitable design of the hybrid.
 - Many analytical and operational problems can be formulated as multi-objective optimization problems by suitably identifying the multiple objectives. Then, Evolutionary Multi-objective Optimization techniques can be employed leading to innovation in decision making.
 - Real-world data sets rather than synthetic data should be analyzed by the EC techniques so that the practicing community gets convinced about the utility of these techniques. Further, benchmark datasets should be used.
 - Customer Life Time Value (LTV), Default detection, and Customer Campaigning have a tremendous scope for further research involving the application of EC techniques, as not much work is reported in this area. CLV estimation can be posed as a regression problem and solved using EC.
 - In future, EC techniques can be applied to solve new analytical and operational problems such as sentiment classification and class association rule mining for sentiment analysis, fraud detection, etc.
 - Fraud Detection is of interest to many financial institutions because the world is becoming more and more technology-dependent for performing financial transactions and services. Further, the problem becomes more complicated because of inherent imbalance that the datasets are replete with. In other words, the companies have a disproportionately small number of fraudulent transactions than the number of genuine transactions. Therefore, it is very difficult to detect fraud without resorting to data balancing techniques. While one-class classification is the ideal way to solve this problem, the application of EC techniques in building one-class classifiers is worth investigating. Another direction is the application of data balancing before invoking EC techniques for binary classification.
 - Further, MBA being an important task using which one can grow one's business, EC techniques-driven class association rule mining will yield insights as to

what attributes together lead to successful cross-sell and thereby build better recommendation engines. These are alternative to traditional association rules.

- It is found that high utility itemset mining, generates high amount of return on investment as frequent itemsets are not always interesting and sometimes rare itemsets yield high profit. In this connection, we observe that evolutionary algorithms find an interesting application in this area as well, with utility as the objective function.
- Since, quantum-inspired EC techniques are gaining popularity, their potential in solving analytical and operational problems is worth investigating.
- Credit recovery analytics involves the use of predictive analytics to identify those defaulted customers, who have the propensity to repay the loans but failed to do so for some valid and legitimate reasons. This is one problem that is typically formulated as a classification problem and solved using data mining and text mining techniques, but the application of EC techniques is conspicuously absent. This problem can also be explored in future.
- Operational Risk pertains to all other risks different from credit and market risk. It is the risk emanating from people, process, and technology. Predicting the financial loss due to operational risk collapses into a regression task, wherein the potential loss from an untoward event becomes the target variable, whereas the data on losses from past events, event triggers, etc. become predictor variables. These events include fraudulent online transactions, cyber attacks, etc. Banks started collecting data associated with operational risk. This regression problem can also be solved using EC techniques.
- We did not consider pure and hybrid soft computing techniques for the review. However, EC techniques have a tremendous role to play in that direction too. Since many analytical problems are binary classification problems, the issues of wrapper-based feature subset selection, classification / regression / association / clustering can be solved using two-tiered EC architectures.
- Finally, one caveat is in order. It is well known that user-defined parameters have to be tuned carefully for the EC techniques. It consumes time and effort to fine

tune and identify the optimal combination of parameters for the EC techniques in a given application. Similar to all other non-parametric, intelligent techniques, this is one limitation of these techniques.

2.7 Conclusions

We reviewed all the articles from 1998 to 2020 that are related to the application of EC techniques to solving analytical and operational problems. This paper has provided a detailed review based on twelve 12 analytical and operational problems like MBA, Fraud Detection, Churn detection, Sentiment Analysis, Direct Marketing, Customer Segmentation, Customer Lifetime Value (LTV), Default detection, Service Marketing and Customer Campaigning. The paper also details how EC techniques can be used to improve the relationship with the customer, thereby increasing the return on investment to the company.

We conclude that EC techniques have a spectacular role to play in solving analytical and operational problems and that this survey will be useful to practitioners from service industries, academic researchers working in the analytical and operational problem domains. Survey will also be useful for data miners working in the field of optimization, in general and EC techniques, in particular. The list of future directions is going to be very useful for budding research scholars in this area.

Chapter 3

A Hybrid and the variants of Differential Evolution and Harmony Search

3.1 Introduction

Optimization refers to minimization or maximization of a objective function with associated decision variables within specified bounds that may or may not have constraints. The optimization performed with constraints is called constrained optimization while the one performed without constraints are called unconstrained optimization. The branch of optimization consists of sub areas such as multi-objective optimization or multi-modal optimization. Optimization finds its application in various areas of engineering, research, economics, biology, science, etc. There are three methods of optimization, Classical optimization, Heuristic-based optimization, and Metaheuristic-based optimization. The difference between these methods is their application. For example, classical optimization techniques are used in convex, continuous and differential search problems. Whereas, the other two methods could be utilized for non-convex, discontinuous and non-differential problems. Moreover, the heuristic-based optimizations have some associated assumptions with them while other two methods don't have any assumption [105, 128].

The optimization techniques are further divided into different categories. For example, meta-heuristic based techniques are further divided into evolutionary com-

puting, swarm intelligence, stochastic optimization, etc. Similarly, metaheuristic algorithms are divided into population-based or point-based algorithms. Reliability of Complex Systems (RCS) has a huge impact on reliability engineering. The aim of reliability optimization is to develop a continuous cost-effective system [214, 251, 302]. The purpose of RCS is to determine the optimal component reliabilities in the whole process. The present study considered three RCS problems.

Three optimization algorithms are proposed namely a variant of Differential Evolution (DE) i.e. Modified Differential Evolution (MDE), a variant of Harmony Search (HS) i.e. Improved Modified Harmony Search (IMHS) and a hybrid optimization algorithm i.e. Improved Modified Harmony Search + Modified Differential Evolution (IMHS + MDE). These algorithms are tested on three problems associated with Reliability of Complex Systems using static extensive constraint handling. The hybrid is compared to HS, IMHS, DE, MDE and other existing algorithms. The experimental results revealed that IMHS + MDE takes less function evaluations (FE) as to the compared algorithms.

3.2 Motivation

- Develop new variants of existing EC techniques and also a hybrid EC technique.
- Test the EC techniques, the developed EC variants, and the hybrid EC technique on the test problems.

3.3 Related Work

RCS have a great impact in the reliability engineering which are non-convex problems. Many techniques are developed for finding solutions of RCS. Improved non-equilibrium simulated annealing (INESA) [251] was developed by Ravi et al. to solve some of the RCS problems. In [252], have applied TA [85] for solving the fuzzy problem. Furthermore, Shelokar et al. in [269], found better optima than that of [251] with ACO algorithms. Later in [250], RCS was solved with even better optima than in [269] by Modified Great Deluge Algorithm (MGDA). Furthermore, Jaya Krishna and Ravi [167] solved the reliability redundancy problems by using IMHS with best FE.

There are a class of meta-heuristics which use different combinations of meta-heuristics. These various combinations of algorithms are called a memetic algorithm (MA). The 1st generation MAs were hybrid optimization algorithms. They used the strength of one metaheuristic to perform exploration and the strength of other for local searching. Chen et al. in [60] have provided multi-facets of MAs. In [226], the history, current trend and future of MAs are presented.

In Bhat et al. (2006) [31], improved DE is proposed in combination with the reflection property of the simplex method for the efficient parameter estimation of biofilter modeling. In Chauhan and Ravi (2010) [55], a hybrid, based on DE and TA named DETA, is developed for the unconstrained problems and is compared with DE. The results regarding FE show that DETA outperforms DE. Choudhuri and Ravi (2010) [64], proposed a hybrid combining Modified Harmony Search (MHS) and the Modified Great Deluge Algorithm (MGDA) for the unconstrained problems and compared them with MHS. The results prove that MHS+MGDA perform better than MHS. In Maheshkumar and Ravi (2011) [202], a modified harmony search and threshold accepting hybrid was proposed and compared with HS and Modified HS on the unconstrained problems. In Maheshkumar et al. (2013) [203], a hybrid combining both PSO and TA, called PSOTA was developed for unconstrained optimization problems, which gave best optimization results than PSO. Finally, a hybrid optimization algorithm combining ACO, and the classical optimization algorithm called Nelder-Mead simplex was developed to train a neural network for bankruptcy prediction [268].

Several hybrid optimization algorithms are developed by employing either HS or DE like DE+TS [289], DE+TA [55], HS and PSO (NHPSO) [187], HS, PS and ACO (HPSACO) [148], HS and MGDA (MHS+MGDA) [64], HS and TA (MHSTA) [202].

3.4 Improved Modified Harmony Search + Modified Differential Evolution

3.4.1 Improved Modified Harmony Search

Harmony Search (HS) is a process imitating metaheuristic presented by Geem et al. in [104]. It imitates the technique used by a musician to develop new beats of music from old music or developing new music from nothing. The HS employs four parameters namely PAR, HMCR, BW and HMS. In HS, harmony update is a significant part of HS

that produces new solution vectors. The harmony update is performed in a systematic manner using following steps:

1. It picks up the solution from harmony memory if the generated random number lies between 0 and 1 i.e. it is below HMCR.
2. (a) Look into the local neighbourhood of selected solution vector if the generated number is less than PAR.
(b) Else if the acquired number is bigger than the PAR, stick with the generated solution vector.
3. If the generated number is higher than HMCR, pick the random value from the domain of decision variable.

The procedure for Improved Modified HS (IMHS) is given below:

1. Decide on the size of harmony memory (HM).
2. Start with HM number of solution vectors and measure the related fitness value.
3. HMCR is incremented in a linear way ranging from 0.7 to 0.99. The value of PAR is fixed between 0 and 1.
4. The process of updating is performed i.e. old vectors are used or new vectors are generated from the older ones or totally new unique vectors are generated by using the HMCR and PAR.
5. The fitness value is analyzed at this point. If the fitness value is better than worst solution, then substitute the worst solution vectors and the fitness values with the newly generated solution.
6. The step number 3, 4, and 5 should be repeated until the maximum of iterations are met.

3.4.2 Modified Differential Evolution

It is a population based evolutionary metaheuristic. It is designed for real input parameter-based optimization problems. There are four major parts of DE i.e. the first step is the initialization or assignment of all solutions with random values; mutate the solutions; recombine the solutions and select the final solution (replace the old solution with new one if it is better than the previous one). The new solutions are generated from the existing solution by mutation and crossover. A slight modification is done in the selection operation of the modified DE (MDE). **In selection, the newly found solution's fitness is compared with the worst solution instead of current solution and the solution with better fitness among the two is selected.** There are two parameters in DE namely CR and F. The working of MDE is elaborated in the following steps:

Step-1: Initialization

1. Initialize 'NP', the number solution vectors of dimensionality size, 'Dim'.
2. Check whether the solution vectors are within bounds.
3. Compute the fitness values of the 'NP' number of solution vectors.

Step-2: Mutation

4. Let x_1 , x_2 , x_3 and $x^{current}$ be the three unique solution vectors chosen randomly from the population 'NP'.
5. Perform the mutation using the following equation:

$$u'_i = x_{1,i} + F * (x_{2,i} - x_{3,i})$$

The above equations are of a variant of DE i.e. DE/rand/1/bin. Differential weight 'F' is between 0 and 2

Step-3: Recombination

6. Using the mutation vector and crossover rate parameter CR recombination is performed as follows:

$$v_i = \begin{cases} u_i & \text{if } (rand(0,1) \leq CR \mid i = I_{rand}) \\ x_i^{current} & \text{if } (rand(0,1) > CR \text{ and } i \neq I_{rand}) \end{cases}$$

CR is between 0 and 1. I_{rand} is a random dimension between 1 to Dim and I_{rand} is used so that v and $x^{current}$ are different.

Step-4: Selection

7. If $Fitness(v) \leq Fitness(x^{worst})$ replace worst vector with solution vector ‘ v ’ in the population.
8. Else do not replace the worst vector.
9. Repeat steps 4-8, until a maximum number of iterations.

3.4.3 Improved Modified Harmony Search + Modified Differential Evolution

Das et al. in [72], show that the explorative power of HS is initially is very good. But, as the number of iterations increase, it gets stuck at the local optimal basin, without progressing further. To overcome this, HS is to be perturbed with a good and powerful meta-heuristic. In [153], various performance measures such as dimensionality, the number of local optima, interval span of side constraints, the ratio of local optima, valley structure coefficient, and peak density ratio were compared to a set of good meta-heuristics. These compared meta-heuristics are as follows: Water Cycle Algorithm (WCA), Random Search (RS), GA, DE, PSO, HS, CS and SA. The compared DE gave best results than all those with a good coverage, and HS gave comparatively better results than some with second-best coverage when a radar plot was created with various performance measures. This gave us the motivation to combine HS with DE to obtain a best optimization algorithm, as DE gave best results for all performance measures.

IMHS, which was developed for complex reliability systems in [167], is combined with differential evolution. A modified DE is developed with an alternate selection strategy in [169]. This MDE does the exploitation well as the selection of DE is replaced with that of HS where a good solution replaces the worst solution instead of the current solution vector. Selection operation usually suggests the exploitation power of any evolutionary algorithm as suggested in Das et al. in [72]. So, the DE selection strategy is replaced with that of HS, as it replaces the worst solution vector, which is better than DE regarding exploitation.

The proposed IMHS+MDE hybrid optimization algorithm runs in two critical stages cyclically. In the first stage, IMHS is run for a fixed number of generations. The resulting population from IMHS is then passed to MDE in the second stage, where MDE is run for a fixed number of generations. The population from MDE is then passed on to IMHS. Both these stages are repeated (cycled) till the maximum number of iterations is reached or optimum is reached (see Figure 3.1).

3.5 Constraint Handling

Constraint handling for the optimization problems means that constraints are handled by special procedures to search the search space effectively thereby satisfying the constraints. The search space is split into feasible region and unfeasible regions depending upon the satisfaction of the constraints. It is divided into feasible region if constraint is satisfied and unfeasible region if constraints are not satisfied. Approaches such as i) Repair operators; ii) Penalty Functions; iii) Hybrid methods; iv) special operators; etc. are used for handling constraints.

In case of problems associated with the evolutionary computing, penalties are usually used. The penalties are applied for handling constraints i.e. penalties are imposed on constraints when the optimization algorithm searches in infeasible region. The common approaches utilized for employing penalties are the following: 1. Static Penalty 2. Adaptive Penalty 3. Dynamic Penalty 4. Death Penalty.

3.5.1 Static Extinctive Penalty

Static extinctive penalties were used for addressing the problem of constraint handling. In this technique the applied penalty remains constant and independent of the current generated solution in the whole process. Static penalty is applied in different ways i.e. 1) extinctive penalty, 2) binary penalty, and 3) distance-based penalty, etc.

A low penalty value will be employed for maximization of the objective function and high penalty value will be employed for minimization of the objective function. The process of using a small or large constant removes the infeasible solution in further evaluations. The approach utilized for dealing with constraints in solving the current problems, is to employ large static value for the infeasible solutions.

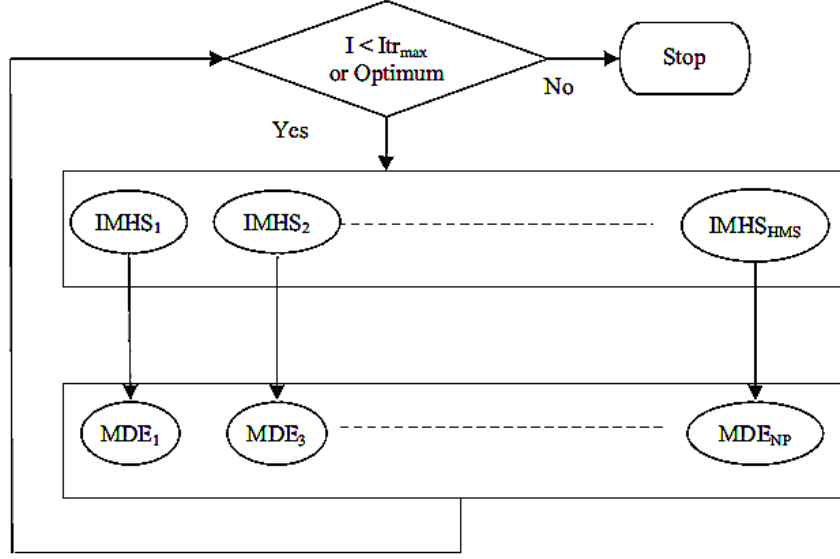


Figure 3.1: The Process of coupling between IMHS memory and MDE population vectors

3.6 Problem Definitions

Problem-1: Life support system in a space capsule [302] is shown in Figure 3.2. The component and system reliabilities are, R_i , $i = 1, 2, \dots, 4$ and R_s , respectively

$$R_s = 1 - R_3[(1 - R_1)(1 - R_4)]^2 - (1 - R_3)[1 - R_2\{1 - (1 - R_1)(1 - R_4)\}]^2$$

Objective function is C_s i.e. system cost which is to be minimized. The constraints are the component and system reliabilities which are R_i i.e. between $0.5 \leq R_i \leq 1.0$, $i = 1, 2, 3, 4$ and R_s i.e. between $R_{s,min} \leq R_s \leq 1.0$ respectively. There are two variants of system cost for problem-1, which are given below:

Case (i):

$$C_s = 2M_1R_1^{\delta_1} + 2M_2R_2^{\delta_2} + M_3R_3^{\delta_3} + 2M_4R_4^{\delta_4}$$

where, $M_1 = 100$, $M_2 = 100$, $M_3 = 200$, $M_4 = 150$, $R_{s,min} = 0.9$ and $\delta_i = 0.6$ for $i = 1, 2, 3, 4$.

Case (ii):

$$C_s = \sum_{i=1}^4 M_i \left[\tan \left(\frac{\pi}{2} R_i \right) \right]^{\delta_i}$$

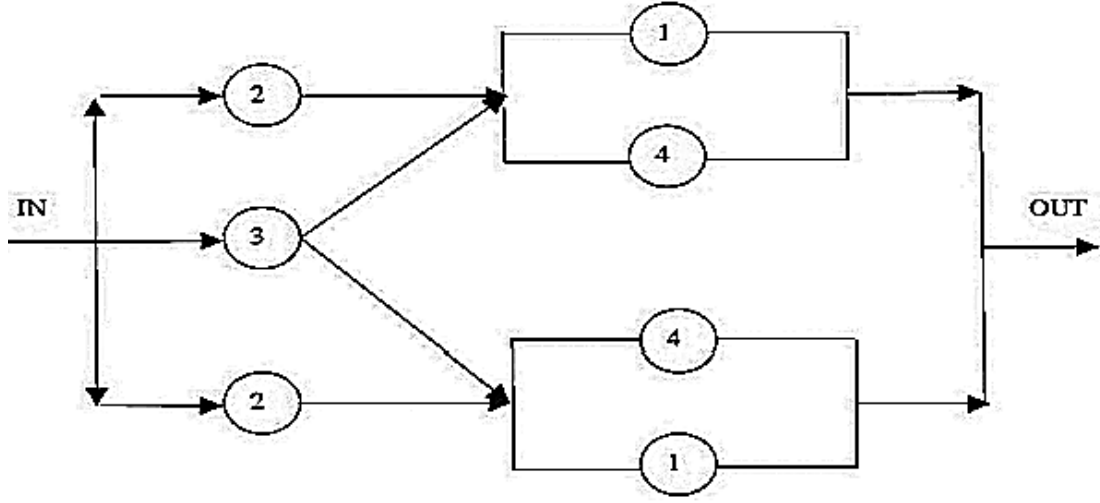


Figure 3.2: Complex System of Life Support System in a space capsule

where, $M_1 = 25$, $M_2 = 25$, $M_3 = 50$, $M_4 = 37.5$, $R_{s,min} = 0.99$ and $\delta_i = 1.0$ for $i = 1, 2, 3, 4$.

Problem-2: Complex Bridge System [214] is shown in Figure 3.3

The component and system reliabilities are, R_i , $i = 1, 2, \dots, 5$ and R_s , respectively.

$$R_s = R_1R_4 + R_2R_5 + R_2R_3R_4 + R_1R_3R_5 + 2R_1R_2R_3R_4R_5 - R_1R_2R_4R_5 \\ - R_1R_2R_3R_4 - R_1R_3R_4R_5 - R_2R_3R_4R_5 - R_1R_2R_3R_5$$

$$\text{Minimize: } C_s = \sum_{i=1}^5 l_i \left[\exp \left(\frac{m_i}{1-R_i} \right) \right]$$

Objective function is C_s i.e. system cost which is to be minimized. The constraints are the component and system reliabilities which are R_i i.e. between $0 \leq R_i \leq 1.0$, $i = 1, 2, 3, 4, 5$ and R_s i.e. between $0.99 \leq R_s \leq 1.0$ respectively. Here $l_i = 1$ and $m_i = 0.0003$ for $i = 1, 2, 3, 4, 5$.

3.7 Results and Discussion

The experimental results generated for problem 1 are presented in Table 3.1, 3.2 and 3.3. According to the results and statistics, it is evident that IMHS + MDE reached

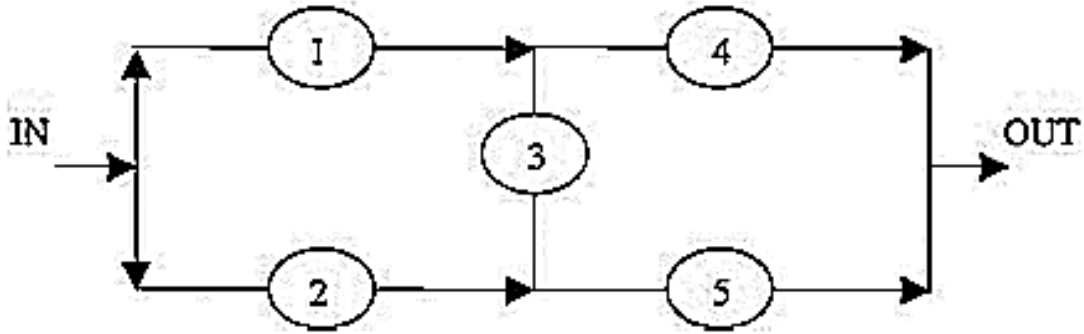


Figure 3.3: Reliability Complex Bridge System

optimum in problem 1 case 1 and in problem 2 with best FE. The numerical values from the results show a 2.6%, 27.6%, 99.4%, 99.4%, 92.3%, 75.1% indicated improvement with respect to FE, compared to MDE, DE, IMHS, HS, MGDA, ACO respectively for the problem 1 case 1. For problem 1 case 2, the hybrid IMHS + MDE gained 24.4%, 22.9%, 99.7%, 99.7%, 98.8%, 96.0% improvement in terms of FE, compared to MDE, DE, IMHS, HS, MGDA, ACO respectively. IMHS+MDE achieved 57.0%, 89.5%, 97.5%, 91.1%, 93.7%, 96.0% improvement for the problem-2 in terms of FEs, compared to MDE, DE, IMHS, HS, MGDA, ACO respectively.

The statistics in terms of median FE, success rate and standard deviation of the 25 runs results of the converged optima are presented in Table 3.4. Even though comparing the standard deviation and success rate of IMHS+MDE with DE, MDE, IMHS, HS doesn't convey much except in one case of MDE and in two cases of IMHS, HS. Therefore, FEs is the apt metric in this case and IMHS+MDE performed better than all the compared.

From Table 3.4, it can be seen that DE and IMHS+MDE are able to achieve 100 percent success rate than the MDE, IMHS, HS and other contrasted algorithms. Therefore, when it comes to picking the best EC algorithm, it turns out that DE is winner as it has only two parameters and simple. IMHS+MDE has three parameters and it is also a memetic algorithm which involves coupling of two algorithms. Therefore, considering simplicity and number of parameters, DE is the best choice.

The reason why IMHS+MDE performed best is that it involves two EC algorithms, which are used to perturb one another when stuck at local optimum. Therefore,

IMHS+MDE performed the best.

3.8 Conclusions and Future Directions

This study focused on three continuous RCS problems with four and five component reliabilities for the RCS problems. The results depicted that IMHS + MDE gave best results i.e. it provided enhancement in optima with best FE. The study concludes that the proposed hybrid meta-heuristic is effective on RCS problems. The results of experiment also revealed that the proposed meta-heuristic provided best results in some problems. The study further suggests that hybrid and adaptive forms of optimization algorithms could give best results.

Table 3.1: Summary of the Results: Problem-1 Case (i)

Algorithm	R_1^b	R_2^b	R_3^b	R_4^b	R_s^b	C_s^b	FE^a
IMHS+MDE	0.5	0.838920	0.5	0.5	0.9	641.823562	5,003
MDE	0.5	0.838920	0.5	0.5	0.9	641.823562	5,133
DE	0.5	0.838920	0.5	0.5	0.9	641.823562	6,911
IMHS [167]	0.506316	0.832600	0.5	0.5	0.9	642.005607	100000
HS [104]	0.505804	0.832803	0.5	0.500311	0.9	642.025896	100000
MGDA [250]	0.50001	0.838919	0.5	0.5	0.9	641.823608	65,603
ACO [269]	0.5	0.838920	0.5	0.5	0.9	641.823562	20,100
INESA [251]	0.50006	0.83887	0.50001	0.50002	0.90001	641.8332	NA
SA [251]	0.50095	0.83775	0.50025	0.50015	0.90001	641.903	NA
[302]	0.50001	0.84062	0.5	0.5	0.90005	642.04	NA

^a Function evaluations (FE)

^b Median value of 25 runs

Table 3.2: Summary of the Results: Problem-1 Case (ii)

Algorithm	R_1^b	R_2^b	R_3^b	R_4^b	R_s^b	C_s^b	FE^a
IMHS+MDE	0.825570	0.890103	0.628105	0.728733	0.99	390.570889	2,121
MDE	0.825436	0.890103	0.627929	0.729073	0.99	390.570881	2,808
DE	0.825639	0.890207	0.627411	0.728745	0.99	390.570867	2,752
IMHS [167]	0.825437	0.890179	0.627901	0.728797	0.99	390.570880	887,718
HS [104]	0.852543	0.890114	0.627904	0.729057	0.99	390.570876	720,236
MGDA [250]	0.825808	0.890148	0.627478	0.728662	0.99	390.570190	188,777
ACO [269]	0.825895	0.890089	0.627426	0.728794	0.99	390.570893	54,140
INESA [251]	0.82516	0.89013	0.62825	0.72917	0.99	390.572	NA
SA [251]	0.82529	0.89169	0.62161	0.72791	0.990003	390.6327	NA
[302]	0.825895	0.890089	0.627426	0.728794	0.99041	397.88	NA

^a Function evaluations (FE)

^b Median value of 25 runs

Table 3.3: Summary of the Results: Problem-2

Algorithm	R_1^b	R_2^b	R_3^b	R_4^b	R_5^b	R_s^b	C_s^b	FE ^a
IMHS+MDE	0.934708	0.935126	0.792849	0.934505	0.935168	0.99	5.019918	3,192
MDE	0.934283	0.935384	0.791964	0.935249	0.934672	0.99	5.019918	7,432
DE	0.934948	0.935064	0.791613	0.935035	0.934582	0.99	5.019918	30,593
IMHS [167]	0.934791	0.934732	0.796893	0.934792	0.934792	0.99	5.019918	132,104
HS [104]	0.935206	0.934877	0.788511	0.935146	0.934699	0.99	5.019918	36,196
MGDA [250]	0.935400	0.935403	0.788027	0.93506	0.934111	0.99	5.019919	50,942
ACO [269]	0.933869	0.935073	0.798365	0.935804	0.934223	0.990001	5.019923	80,160
INESA [251]	0.93747	0.93291	0.78485	0.93641	0.93342	0.99	5.01993	NA
SA [251]	0.93566	0.93674	0.79299	0.93873	0.92816	0.99001	5.01997	NA
[214]	0.93924	0.93454	0.77154	0.93938	0.92844	0.99004	5.02001	NA

^a Function evaluations (FE)

^b Median value of 25 runs

Table 3.4: Statistics of the employed optimizations algorithms

Method	RCS	Problem	Median NFEs ¹	Success Rate ³	SD ²	Max. NFEs	Parameters
HS [104]	Problem-1: Case (i)		1,000,000	0.12	0.2660	1000000	PAR=0.2, BW=1e-06, HMC=0.75
	Problem-1: Case (ii)		720,236	0.84	7.01e-05	1000000	PAR=0.45, BW=0.001, HMC=1.0
	Problem-2		36,196	1.0	1.38e-07	1000000	PAR=0.5, BW=0.001, HMC=0.9
IMHS [167]	Problem-1: Case (i)		1,000,000	0.04	2.2250	1000000	PAR=0.3, BW=1e-04
	Problem-1: Case (ii)		887,718	0.76	8.05e-05	1000000	PAR=0.5, BW=0.001
	Problem-2		132,104	1.0	1.13e-07	1000000	PAR=0.4, BW=0.001
DE	Problem-1: Case (i)		6,911	1.0	7.93e-08	1000000	CR=0.9, F=0.5
	Problem-1: Case (ii)		2,752	1.0	3.08e-05	1000000	CR=0.95, F=0.6
	Problem-2		30,593	1.0	1.29e-07	1000000	CR=0.35, F=0.45
MDE	Problem-1: Case (i)		5,133	1.0	8.18e-08	1000000	CR=0.65, F=0.65
	Problem-1: Case (ii)		2,808	1.0	3.35e-05	1000000	CR=0.8, F=0.7
	Problem-2		7,432	0.52	0.3577	1000000	CR=0.8, F=0.95
IMHS+MDE	Problem-1: Case (i)		5,003	1.0	8.70e-08	1000000	PAR=0.15, BW=1e-04, CR=0.7, F=0.75
	Problem-1: Case (ii)		2,121	1.0	2.32e-05	1000000	PAR=0.35, BW=1e-04, CR=0.9, F=0.7
	Problem-2		3,192	1.0	1.42e-07	1000000	PAR=0.25, BW=0.001, CR=0.8, F=0.7

¹Median value of the FE of 25 runs

²Standard deviation of the converged optima for 25 runs

³Percentage of the number times optimum is converged for 25 runs

Chapter 4

Large Scale Global Optimization via Improved Modified Harmony Search and Modified Differential Evolution Hybrid

4.1 Introduction

Metaheuristic-based optimization techniques are classified into Evolutionary Computing (EC), Swarm Intelligence-based optimization (SI), Stochastic-based optimization, Physics-based Optimization, Artificial Immune System (AIS)-based optimization, etc. Metaheuristic algorithms are further classified into the population, or point-based, optimization algorithms. Evolutionary computing techniques include the Genetic Algorithm [110], Genetic Programming [164], and Differential Evolution (DE) [110], etc., which are all single objective optimization algorithms. However, Multi-Objective Genetic Algorithm [159], Strength Pareto Evolutionary Algorithm-2 [342], Nondominated Sorting Genetic Algorithm-II (NSGA-II) [78], etc. are examples of multi-objective evolutionary computing algorithms. Swarm Intelligence-based optimization algorithms include Particle Swarm Optimization (PSO) [151], Ant Colony Optimization [67], and Bee Swarm-based Optimization [146], etc. Stochastic-based optimization algorithms include Tabu Search [107, 108], Stochastic Hill Climbing [260], and Threshold Accepting (TA) [85], etc. Physics-based optimizations include Simulated Annealing [156], Har-

mony Search (HS) [104], etc. The Artificial Immune System [48, 75] based optimization, includes Negative Selection, Clonal Selection, etc.

There is also another class of meta-heuristics that combine the power of more than one meta-heuristic. These are called Memetic Algorithms (MA). MAs have evolved over three generations. First generation MAs are the hybrid optimization algorithms. These use the power of one class of metaheuristic to carry out the exploration and the power of another class of metaheuristic to do a local search. In a survey, presented by Ong and Chen (2011) [226] multi-facets of memetic algorithms are discussed. In [226], the past, present, and future of memetic algorithms is discussed. In this work; we are proposing a first-generation hybrid-iterative two-stage memetic algorithm by combining Harmony Search and Differential Evolution. Harmony Search does the job of exploration and exploitation, and Differential Evolution does the job of giving a perturbation to the HS for coming out of basin (local optima) near optimal value. But here, we are utilizing an Improved Modified Harmony Search (IMHS) proposed by [167] for finding optimal values in reliability complex systems problems. Also, Modified Differential Evolution (MDE) with alternate selection strategy proposed by [169] to detect outliers using evolutionary computing.

4.2 Related Work

Large Scale Global Optimization (LSGO) benchmarks were proposed in a special session of Congress on Evolutionary Computation (CEC) 2008 [299]. Later these benchmarks were upgraded in CEC 2010 LSGO special session, and the same was utilized in CEC 2012 LSGO special session [298]. Some of the excellent LSGO algorithms applied on 20 benchmarks LSGO functions were presented in CEC 2010 LSGO special session [40, 160, 216, 224, 313, 314]. Another set of LSGO algorithms applied on the same LSGO functions was presented in CEC 2012 LSGO special session [38, 96, 182, 297, 331]. Later, another benchmark called CEC 2013 LSGO special session was proposed in 2013 and this benchmark still needs to be solved effectively.

In CEC 2010 LSGO special session, Memetic Algorithm and Solis-Wets' with local Chains (MA-SW-Chains) of Molina et al.'s became the winner of the LSGO competition. In Molina et al. [216] developed the MA-SW-Chains algorithm, by chaining different local search applications, assigns to every individual a local search intensity

that depends on its features. In [314], a two-stage-based ensemble optimization evolutionary algorithm (EOEA) is designed to handle LSGO problems. Omidvar et al. [224] proposed Differential Evolution with Cooperative Co-evolution - Delta Method (DECC-DML) to capture interacting variables for more effective problem decomposition suitable for cooperative coevolutionary frameworks to solve LSGO problems. In [334], the dynamic multi-swarm particle swarm optimizer (DMS-PSO) and a sub-regional harmony search (SHS) are hybridized to obtain DMS-PSO-SHS. In DMS-PSO-SHS, sub-swarms are dynamic and small, which is also appropriate to be the population of the harmony search. In Brest et al. [40], authors present the self-adaptive differential evolution algorithm jDElsgo on large-scale global optimization. In [160], Differential Ant-Stigmergy Algorithm (DASA) is developed for LSGO, which transforms a real-parameter optimization problem into a graph-search problem, and then the parameters' differences assigned to the graph vertices are used to navigate through the search space. In Wang et al. [313], sequential Differential Evolution (SDE), enhanced by neighborhood search (SDENS) is proposed, which consists of two steps. In the first step, for every individual, we create two trial individuals by local and global neighborhood search strategies. Second, we select the fittest one among the current individual and the two created trial individuals as a new current individual. The above algorithms in this paragraph are all presented in CEC 2010 LSGO special session.

The Multiple Offspring Sampling (MOS) [182] algorithm became the winner of the CEC 2012 LSGO competition. The Multiple Offspring Sampling framework has been used to combine two different heuristics: the first one of the local searches of the Multiple Trajectory Search (MTS) algorithm, and the well-known Solis and Wets heuristic. In [38], the authors present the self-adaptive differential evolution algorithm (jDEsps) with a small and varying population size on large-scale global optimization. In [297], the performance of Differential Evolution with Landscape Modality Detection and a Diversity Archive (LMDEa) reported on the set of benchmark functions provided for the CEC 2012 Special Session on LSGO using a small population size and a large archive for diversity. In [96], a Memetic ABC (MABC) algorithm was developed and hybridized with two local search heuristics: the Nelder-Mead algorithm (NMA) and the Random Walk with Direction Exploitation (RWDE). The former tends more towards exploration, while the latter more towards exploitation of the search space. In [331], a Cooperative Coevolution Evolutionary Algorithm (CCEA) with Global Search (CCGS)

is presented to handle the premature convergence of CCEA and applied to LSGO problems.

Recently, two new LSGO algorithms, namely the Memetic Algorithm with Adaptive Local Search Depth (MA-ALSD) in [192] and the Joint Operations Algorithm (JOA) in [295] have been proposed with CEC 2010 LSGO benchmarks. The MA-ALSD presented in CEC 2014 was with the CEC 2010 LSGO benchmarks. In [295], JOA was compared with six other metaheuristics. Of all the algorithms that were compared, JOA gave the best results with regards to the mean optimal value.

Mahdavi et al. in [201] surveyed the LSGO problems, with a good classification of LSGO algorithms that is based on the approach used for large scale optimization. A description of the application of LSGO algorithms used for the various real-world problems, like data mining, mechanics, etc., is also surveyed in the above article. In [184], a comprehensive comparison of all the winning LSGO algorithms in various LSGO competitions, was made. A comparison of the algorithms of all the LSGO winners (along with other competition winners) is also made on the CEC 2013 LSGO benchmarks.

4.3 CEC 2013 LSGO Benchmarks

A large number of meta-heuristics are developed in the recent times. For all these meta-heuristics, performance gradually decreases as dimensionality increases. This problem with high dimensionality is termed “Curse of Dimensionality” [188]. Also, many real world problems are high dimensional in nature. The following describes why high dimensional problems are hard to solve:

1. Search space explodes as dimensionality increases
2. High dimensional problem evaluation takes a lot of computational power and is usually costly
3. There may be a dependency between variables
4. Search space properties may change as dimensionality increases.

To address these issues of high dimensional problems, CEC, in 2008, started a special session to come up with large-scale global optimizing algorithms. This initial

LSGO special session uses only simple functions in their test suite. Later, in CEC 2010 and CEC 2012, it has been extended to make this special session a suitable evaluation platform for testing and comparing large-scale global optimization. CEC 2013 LSGO special session [188] was developed by keeping in mind the following four types of high-dimensional problems: 1) separable functions, 2) partially-separable functions, 3) functions that consist of overlapping subcomponents and 4) fully-nonseparable functions. In CEC 2013 LSGO special session are developed by adding the following features to the high dimensional problems:

- 1) Nonuniform subcomponent sizes;
- 2) Imbalance in the contribution of subcomponents;
- 3) Functions with overlapping subcomponents;
- 4) New transformations to the base functions:
 - Ill-conditioning;
 - Symmetry breaking;
 - Irregularities.

The basic functions used in the CEC 2013 LSGO special session are:

1. The Sphere Function
2. The Elliptic Function
3. Rastrigin’s Function
4. Ackley’s Function.
5. Schwefel’s Problem 1.2
6. Rosenbrock’s Function

To these basic functions, 1000 dimensional input vectors are passed, which are computed from the optimization algorithms. These input vectors are shifted, permuted, and rotated (rotated by multiplying with the orthogonal matrix). By applying these shifting, permuting, and rotation operations, the job of the optimization algorithm is made difficult [188]. Each of these benchmark functions are to be run for three million function evaluations (FEs) and for 25 runs. Then median results need to reported for each of the benchmark function.

Table 4.1: Summary of functions used in CEC 2013 LSGO competition

a. Fully-separable Functions	
1	F1: Elliptic Function
2	F2: Rastrigin Function
3	F3: Ackley Function
b. Partially Additively Separable Functions	
i) Functions with a separable subcomponent:	
4	F4: Elliptic Function
5	F5: Rastrigin Function
6	F6: Ackley Function
7	F7: Schwefels Problem 1.2
ii) Functions with no separable subcomponents:	
8	F8: Elliptic Function
9	F9: Rastrigin Function
10	F10: Ackley Function
11	F11: Schwefels Problem 1.2
c. Overlapping Functions	
12	F12: Rosenbrock's Function
13	F13: Schwefels Function with Conforming Overlapping Subcomponents
14	F14: Schwefels Function with Conflicting Overlapping Subcomponents
d. Non-separable Functions	
15	F15: Schwefels Problem 1.2

Of these basic functions, some are unimodal and others are multimodal. Sphere, Elliptic and Schwefel's functions are unimodal, as they have unique local optima. Rosenbrock's, Rastrigin's and Ackley's functions are multimodal, as they have multiple local optima. Some are separable, partially separable and fully non-separable. Sphere, Elliptic, Rastrigin's and Ackley's functions are separable. Some of these are made partially separable by multiplying some parts of the input vector with the orthogonal matrix. However, Schwefel's and Rosenbrock's functions are fully non-separable. Some of the functions are scalable, which means that they can be applied to problems with a dimension of more than 1000.

By considering the above factors 15 benchmarks have been designed. All of these benchmark functions are unconstrained. These 15 benchmark functions of the CEC 2013 LSGO special session, with all the above factors, are summarized in Table 4.1.

4.4 Proposed Hybrid Optimization Algorithm

4.4.1 Nature of Optimization Problem

The current optimization problem has 15 objective functions of the form $F: \mathbb{R} \rightarrow \mathbb{R}$. When given input from a set of bounded real number domains, it gives an output in the real number range. All 15 objective functions are minimization problems by nature. For minimization problems, the solution vector generated from the optimization algorithm, ' X_{new} ', when given to the function, ' F ', should be less than ' X_{old} ', i.e., $F(X_{new}) \leq F(X_{old})$.

4.4.2 Proposed Improved Modified Harmony Search + Modified Differential Evolution

Das et al. (2011) [72] show that the explorative power of Harmony Search (HS) is initially is very good, but that it falls, as the number of iterations increase, and it gets stuck at the local optimal basin, without progressing further. To overcome this, HS is to be perturbed with a good and powerful meta-heuristic. In [153], various performance measures such as dimensionality, the number of local optima, interval span of side constraints, the ratio of local optima, valley structure coefficient, and peak density ratio were compared to a set of good meta-heuristics. These compared meta-heuristics are as follows: WCA, RS, GA, DE, PSO, HS, CS and SA. The compared DE gave better results than all those with a good coverage, and HS gave comparatively better results than some with second-best coverage when a radar plot was created with various performance measures. This gave us the motivation to combine HS with DE to obtain a better optimization algorithm as DE gave better results for all performance measures.

We combined IMHS, which was developed for complex reliability systems in [167], with differential evolution. We also developed a modified DE with an alternate selection strategy in [169]. This MDE does the exploitation well as the selection of DE is replaced with that of HS where a better solution replaces the worst solution instead of the current solution vector. Selection operation usually suggests the exploitation power of any evolutionary algorithm as suggested in Das et al. (2011) [72]. So, we replaced the DE selection strategy with that of HS, as it replaces the worst solution vector, which is better than DE regarding exploitation.

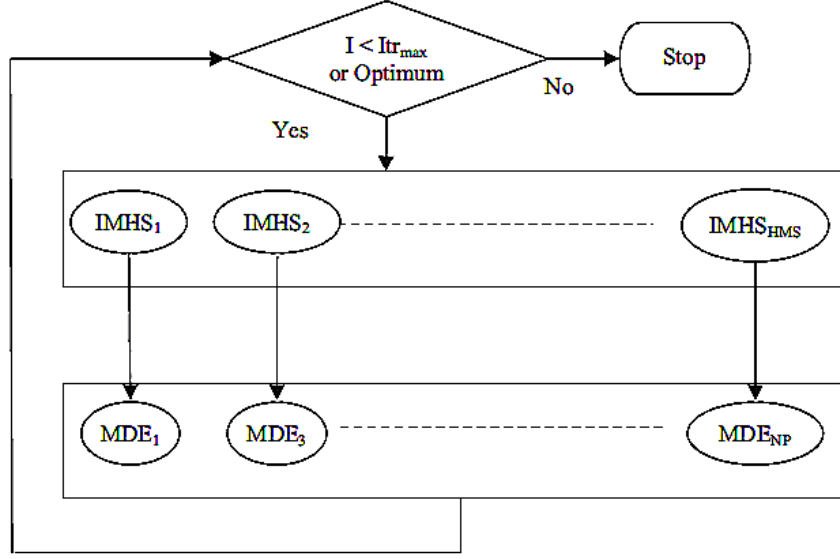


Figure 4.1: The Process of coupling between IMHS memory and MDE population vectors

Also, we tested our hybrid against particle swarm optimization and threshold accepting (PSOTA), on the 30-dimensional problems [203]. Our hybrid gave better results with regard to the success rate and function evaluations, as compared to that of PSOTA, which gave us the initial fuel to test on LSGO problems of CEC 2013 special session. Other criteria for choosing the concerned algorithms for hybridization are that they have fewer parameters to tune. IMHS has dynamic tuning of one parameter while another parameter is fixed and MDE has two parameters to tune. Thus, a total of two parameters are needed to be tuned in the hybrid.

The proposed IMHS+MDE hybrid optimization algorithm runs in two critical stages cyclically. In the first stage, IMHS is run for a fixed number of generations. The resulting population from IMHS is then passed to MDE in the second stage, where MDE is run for a fixed number of generations. The population from MDE is then passed on to IMHS. Both these stages are repeated (cycled) till the maximum number of iterations is reached or optimum is reached (see Figure 4.1).

PHASE I: IMHS

The process flow of IMHS is depicted in Figure 4.2.

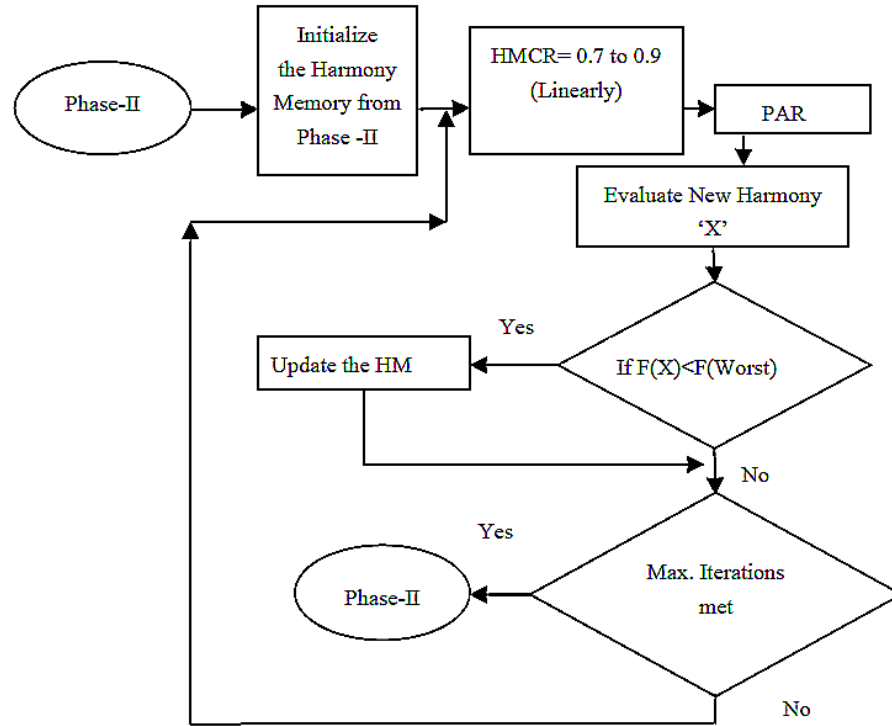


Figure 4.2: IMHS Process Flow

PHASE II: MDE

The process flow of MDE is depicted in Figure 4.3.

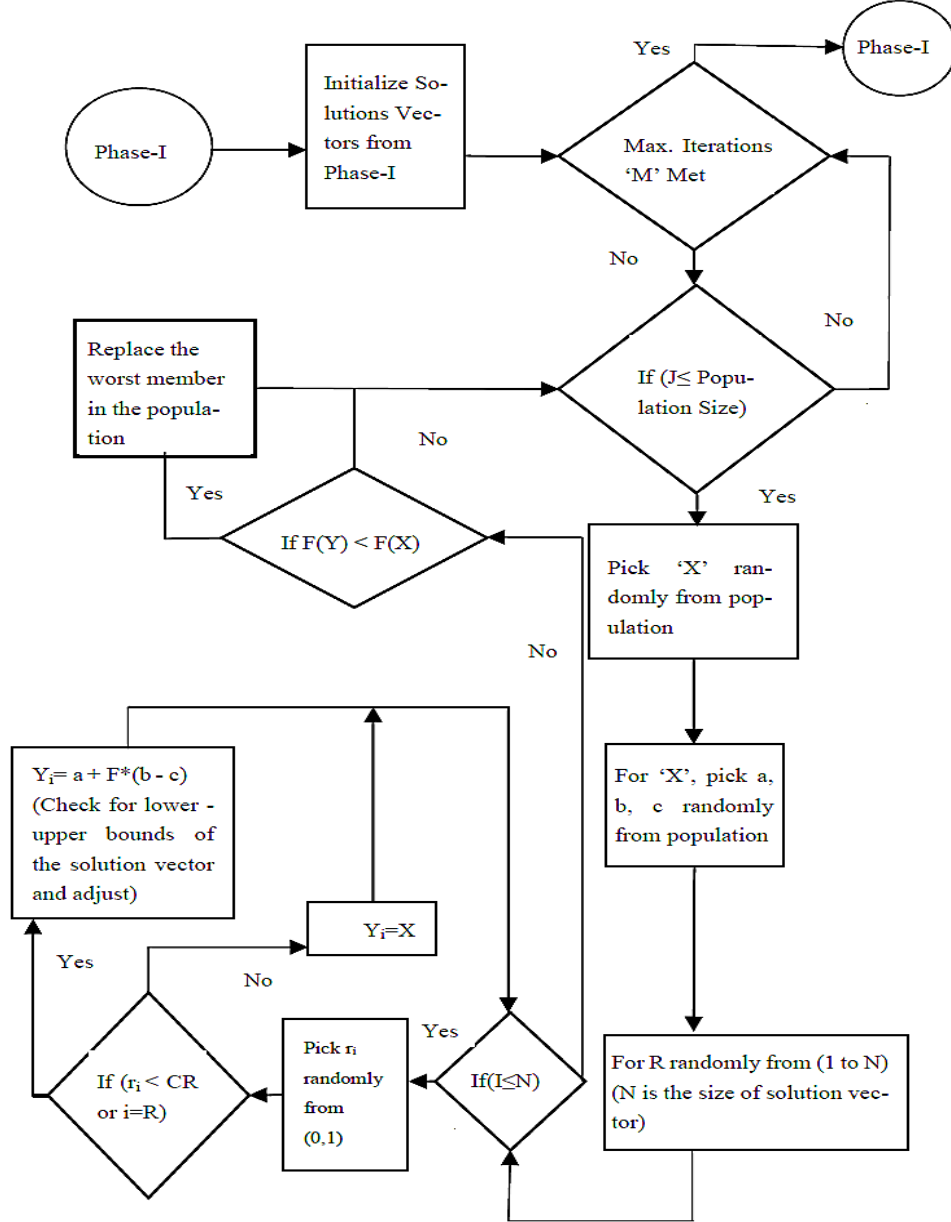


Figure 4.3: MDE Process Flow

In the proposed hybrid optimization algorithm, the Improved Modified Harmony Search (IMHS) is proposed in [167], which has an 'hmcr' value linearly increasing from

0.7 to 0.9. This is taken and is coupled with Modified Differential Evolution (MDE), proposed in [169], where the selection strategy of DE is replaced with that of HS and where the worst solution will be replaced instead of current solution. The reason for using MDE is perturbing the solutions generated by IMHS as the IMHS has the limitation of getting stuck at the local optima. Also, Kim et al. in [153] show that HS and DE are the powerful metaheuristics when compared with various performance measures. In Das et al. (2011) [72], it is shown that the power of HS depends on the selection strategy of HS, and also, the limitation of HS is that it gets stuck at the local optima. So, these readings gave us an intuition to combine IMHS with MDE. In the proposed hybrid the population of solution vectors from IMHS is passed to MDE iteratively in a loop (see chapter 3.4 for more details).

The proposed hybrid optimization algorithm is employed in two phases cyclically. IMHS is run for a fixed number of generations in first phase. The resulting population from IMHS is then passed to MDE in the second phase, where MDE is run for a fixed number of generations. The population from MDE is then passed on to IMHS. Both these phases are repeated (cycled) till the maximum iterations is met. This population is swapped after a fixed number of generations between IMHS to MDE and also from MDE to IMHS till the maximum outer iterations. At first, IMHS, is run for 10000 generations, and then MDE is run for 20000 generations. So, a total of $(10000 \text{ (IMHS)} + 20000 \text{ (MDE)}) = 30000$ generations. One pass through IMHS and MDE is termed one outer iteration. The algorithm is run for 100 passes that is for a total of $(100 \times 30,000) = 3,000,000$ inner iterations or FE's take place as set by the CEC 2013 LSGO benchmark. The proposed hybrid large scale optimization algorithm is presented in the Algorithms 1, 2 and 3

In the proposed hybrid optimization, the modified version of HS is coupled with the modified version of DE. The proposed hybrid has the following modifications to the basic algorithm of HS and DE:

1. Increment HMCR linearly from 0.7 to 0.9 as HS performs better within this range [321]. This modification is needed for good scope for exploration when HMCR is at 0.7 and when HMCR reaches 0.9, exploitation is performed with slight scope for exploration. In the hybrid MHS+TA discussed in the review above, experimentation performed with the HMCR value is incremented linearly from 0

Algorithm 1: Improved and Modified Harmony Search (IMHS)

Input: Initialized Harmony Memory (HM), HMCR and PAR
 $I = 1$
IM-HarmonySearch(HM, Y)
{
 while $I \leq \text{MaxIterations}_{IMHS}$ **do**
 Increment HMCR from 0.7 to 0.9 in steps
 $x' = \text{HarmonyUpdate}(HM, hmcr)$
 $y' = \text{FitnessF}_x(x')$
 if y' better than worst solution vector in HM **then**
 Replace worst vector by x'
 Also replace the corresponding fitness value with y'
 end
 $I = I + 1$
 return (HM, Y)
end
}
HarmonyUpdate($HM, hmcr$)
{
 if $\text{rand}(0, 1) < hmcr$ **then**
 Randomly pick a solution vector from the harmony memory
 if $\text{rand}(0, 1) < \text{par}$ **then**
 Modify the solution vector values
 end
 end
 Generate a new random solution vector
 Check whether the solution vector is within Bounds
}

Algorithm 2: Modified Differential Evolution (MDE)

```
Input: Population ( $P_{DE}$ ), NP, F and C
 $I = 1$ 
 $J = 1$ 
M-DifferentialEvolution( $P_{DE}$ ,  $Y$ )
{
  while  $I \leq \text{MaxIterations}_{MDE}$  do
    while  $J \leq NP$  do
       $\text{int } x = \text{floor}(\text{rand}(1, NP + 1))$ 
       $\text{int } a, b, c$ 
      repeat
         $a = \text{floor}(\text{rand}(1, NP + 1))$ 
      until  $a = x$ ;
      repeat
         $b = \text{floor}(\text{rand}(1, NP + 1))$ 
      until  $b = x \mid b = a$ ;
      repeat
         $c = \text{floor}(\text{rand}(1, NP + 1))$ 
      until  $c = x \mid c = a \mid c = b$ ;
       $R = \text{floor}(\text{rand}(1, Dim + 1))$ 
       $Ind1 = P_{DE}[a]$ 
       $Ind2 = P_{DE}[b]$ 
       $Ind3 = P_{DE}[c]$ 
       $Current = P_{DE}[x]$ 
       $Candidate = P_{DE}[x]$ 
      while  $K \leq Dim$  do
        if  $\text{floor}(\text{rand}(1, Dim + 1)) = R$  or  $\text{rand}(0, 1) < CR$  then
           $ck = Ind1[K] + F * (Ind2[K] - Ind3[K])$ 
        else
           $ck = Current[K]$ 
        end
        if  $ck$  with in bounds then
           $Candidate[K] = ck$ 
           $K = K + 1$ 
        end
      end
       $K = 1$ 
       $x' = Candidate$ 
       $y' = \text{FitnessF}_x(x')$ 
      if  $y'$  better than worst solution vector in  $P_{DE}$  then
        Replace worst vector by  $x'$ 
        Also replace the corresponding fitness value with  $y'$ 
      end
       $J = J + 1$ 
    end
     $I = I + 1$ 
  end
  return ( $P_{DE}$ ,  $Y$ )
}
```

Algorithm 3: IMHS+MDE

Input: $PopulationSize = 200$, $MaxIterations_{IMHS} = 10000$, $MaxIterations_{MDE} = 100$,
 $MaxIterations_{Total} = 100$
Compute for once Fitness ‘Y’ of Randomly Initialized Population Vectors ‘P’
 $P_{HM} = P$
 $Y_1 = Y$
while $K \leq MaxIterations_{Total}$ **do**
 Reset ‘HMCR’ and $MaxIterations$ Counters of IMHS and MDE
 $(P_{DE}, Y_2) = \text{IM-HarmonySearch}(P_{HM}, Y_1)$
 $(P_{HS}, Y_1) = \text{M-DifferentialEvolution}(P_{DE}, Y_2)$
 if $K = 4 \mid K = 20 \mid K = MaxIterations_{Total}$ **then**
 $P = P_{HS}$
 $Y = Y_1$
 $Index = IndexofMin(Y)$
 Print($P[Index]$, $Y[Index]$)
 end
 $K = K + 1$
end

to 1. Though the hybrid MHS+TA gave good results, the convergence was slow as a lot of exploration is performed, which is unnecessary. And, also, when the HMCR value reaches one (1) no more exploration is performed, which causes the algorithm to get stuck at local optima. So, we came up with a better version of HS, where the HMCR is incremented from 0.7 linearly up to 0.9.

2. PAR value is fixed by parameter tuning using ‘GARreal’ function of ‘gaoptim’ package in R. The PAR helps in local improvements, and some of the previous solution vector values are retained without getting a totally new vector out of exploitation.
3. The population size is another parameter which has to be carefully chosen as hybrid yields better results only when the coupling is performed well with right population sizes. Various experimentations with different population sizes i.e. 30, 200, and 300, was performed for a fixed number of function evaluations and 1000 dimensions. Initially, the coupling between population sizes, 30, and 200, yielded incrementally better results. But, after 200, the coupling yielded a decrease in performance as well as no improvement (meaning a lot of overhead for CPU

without much improvement in the betterment of the Optima). Also, the reason for choosing 200 as population size is that DE will have a better scope of the alternate selection strategy (discussed below) for more number of diverse solution vectors in the population than less number of solutions.

4. As discussed, the HS algorithm's selection strategy is utilized in DE making it MDE. The reason is that HS has better exploitative power, with worse vector replacement than DE's current vector replacement. By doing so, if the current vector is better than worst, it remains in the population, rather than getting replaced. By doing so, the next time the DE will have a chance to search the local neighborhood near the current vector.
5. Parameters F and CR are set by careful parameter tuning using 'GARReal' function of 'gaoptim' package in R. Choosing a low value of F gives a chance to search slowly and finely. If CR is set not too high or too low, then it helps in finding the right mix between mutation and recombination by searching the local neighborhood better.
6. The maximum number of inner iterations for the MDE is set to double those of IMHS as more exploitation is needed than exploration. The above helps in fine-tuning to the solution, when more exploitation is performed rather than skipping the solution. Another reason for setting MDE iterations to double those of IMHS is that the step size of MDE is kept low so that it can do fine the tuning well. Also, this cycle of first performing IMHS and then performing MDE is repeated as one contributes to the diversity of the population of the other by repeating iteratively. We have also experimented with different settings of inner iterations to set for IMHS and MDE, and the current setting of inner iterations gave better results with faster convergence.
7. The coupling happening between IMHS and MDE is a form of loose coupling, as the optimization algorithms can be separated from each other, whereas in the tight coupling the optimization algorithms cannot be separated which is not the case here in the proposed approach.

4.4.3 Time complexity of the proposed hybrid

The approach proposed in Das and Suganthan (2011) [74] for DE to compute the time complexity is employed for both the algorithms. The algorithm runs for a total of three million function evaluations (FE's). IMHS, at first, runs for 10000 FE's, and MDE, next, runs for $(100 \times 200) = 20000$ FE's where 100 represents inner iterations and 200 represents the population size. So, a total of $(10000 \text{ (IMHS)} + 20000 \text{ (MDE)}) = 30000$ FE's (this is one outer iteration). These outer iterations repeated 100 times (for a total of $(100 \times 30,000) = 3,000,000$ FE's).

The time complexity for IMHS is (number of inner iterations (I_{IMHS}) * number of outer iterations (OI) * dimensionality of the objective function (D)) = $10000 \times 100 \times D = 1,000,000 \times D = 1/3^{rd}$ of Total FE's (TFE_{IMHS}) * D. So, the time complexity is represented as $O(TFE_{IMHS} \times D)$. The time complexity for MDE is (number of inner iterations (I_{MDE}) * number of outer iterations (OI) * Population Size (PS) * dimensionality of the objective function (D)) = $100 \times 100 \times 200 \times D = 2,000,000 \times D = 2/3^{rd}$ of Total FE's (TFE_{MDE}) * D. So, the time complexity is represented as $O(TFE_{MDE} \times D)$. Combining both we get $O((TFE_{IMHS} + TFE_{MDE}) \times D)$.

4.4.4 Parameter tuning for the proposed hybrid

There has been much debate on using a generalist vs. a specialist way of parameter tuning for Evolutionary Algorithms (EA) [89, 274]. In the current work, we employed a specialist way of parameter tuning. The reason is that according to No Free Lunch Theorem [315] there cannot be an EA, which is uniformly the best for all the optimization problems. As the decision space changes, the parameters used for the EA for the decision space also change. Therefore, a generalist way of parameter tuning is not suitable here. Hence, we employed a specialist way of parameter tuning.

Further, according to Eiben and Smit [89, 274]

“Tuning an EA on a set of functions delivers a generalist, that is, an EA that is good at solving various problem instances. Obviously, a true generalist would perform well on all possible functions. However, this is impossible by the no-free-lunch theorem. Therefore, the quest for generalist EAs is practically limited to less general claims that still raise serious methodology issues”

Due to the afore-mentioned serious issues, as quoted in the above quoted paragraph, we choose to adopt the specialist methodology for the parameter tuning of the non-adaptive parameters. As it can be seen, most of the parameters of the EAs used for most LSGO algorithms are adaptive (i.e. no need of parameter tuning). But, our proposed hybrid EA is a mix of linear increment and non-adaptive parameters.

4.4.4.1 Parameter List and Tuning Methodology

The parameter tuning for the proposed hybrid method for each CEC 2013 LSGO function is performed by running a stochastic optimization on top of the hybrid method. We ran the function called ‘GAReal’ of ‘gaoptim’ package of ‘R’ on top of the proposed hybrid for a fixed number of iterations. Three parameters namely PAR, CR, and F are fine-tuned by the above R function (GAReal). The bounds for the parameters PAR and CR are between 0 and 1. The bounds for the parameter F is between 0 and 2. The HMCr is not tuned as it is linearly changed between 0.7 and 0.9. Tables 4.2 and 4.3 presented, shows the different parameter values of the proposed hybrid for the 15 benchmark functions of the CEC 2013 LSGO.

We can visualize the specialist methodology of parameter values present in Table 4.3 using a box plot depicted in Figure 4.4, which gives readers an intuitive idea of the distribution of the parameters. Also the box plot indicates that our proposed algorithm is more susceptible to the CR compared to other two because of its higher variability and therefore tuning the parameter CR is harder.

Another facet of our proposed algorithm for LSGO is that it is suitable for single objective unconstrained optimization framework. Therefore, the specialist methodology of parameter tuning for each of the problems is apt.

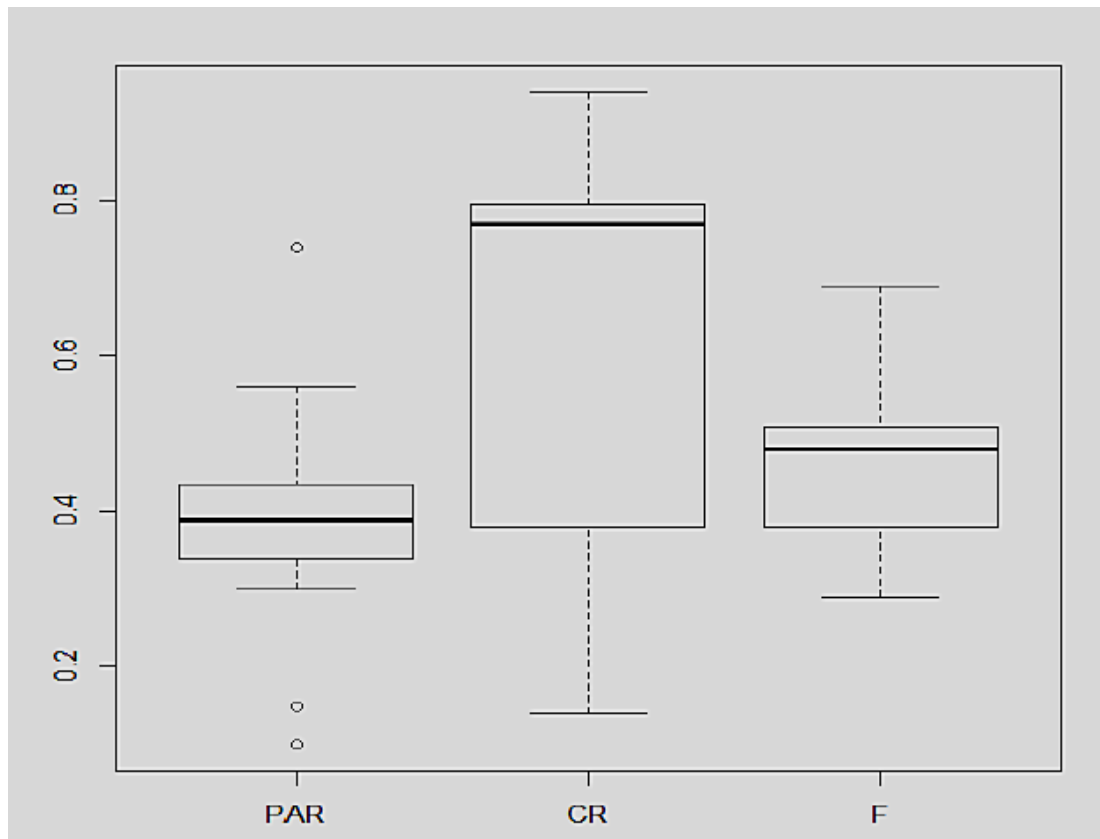


Figure 4.4: Box Plot of the non-adaptive parameters

Table 4.2: Parameters for the IMHS+MDE

Parameters	Values and Ranges
Population Size	200
HMC	0.7 to 0.9 (Linear Increment)
PAR	0.3 to 0.6 (Non-adaptive) (Fixed Values in the Range by Specialist Methodology - See Table 4.3)
CR	0.15 to 0.95 (Non-adaptive) (Fixed Values in the Range by Specialist Methodology- See Table 4.3)
F	0.3 to 0.7 (Non-adaptive) (Fixed Values in the Range by Specialist Methodology - See Table 4.3)

Table 4.3: Specialist Method of Parameter tuning for PAR, CR, and F

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15
PAR	0.3	0.74	0.4	0.34	0.10	0.56	0.42	0.34	0.37	0.15	0.49	0.45	0.39	0.41	0.34
CR	0.36	0.42	0.78	0.94	0.40	0.83	0.79	0.94	0.15	0.20	0.80	0.77	0.76	0.79	0.14
F	0.3	0.3	0.51	0.47	0.29	0.69	0.48	0.39	0.37	0.56	0.46	0.49	0.51	0.48	0.51

4.5 Results Analysis

The proposed algorithm along with the CEC 2013 LSGO benchmarks are implemented in R. The algorithms are executed on Intel(R) Core(TM) i7-6700 @ 3.4 GHz processor and 32 GB RAM. The Operating System used is Windows 8.1 Pro 64 bit. The proposed algorithm are compared with the winners of the CEC 2010 LSGO, SOCO 2011, CEC 2012 LSGO and CEC 2013 LSGO competitions on the CEC 2013 LSGO benchmarks. The results for comparison are taken from [184]. The algorithms of winners of the four competitions listed below (along with competition ranks):

1. CEC 2010 LSGO

- (a) Memetic Algorithm and Solis-Wets' with local Chains (MA-SW-Chains) (Rank 1) [216]
- (b) Two Stage Ensemble (2S-Ensemble) (Rank 2) [314]

2. SOCO 2011

- (a) Multiple Offspring Sampling (MOS) - 2011 (Rank 1) [181]
- (b) Self-adaptive differential evolution algorithm for LSGO (jDElscoP) (Rank 2) [39]
- (c) Generalized adaptive Differential Evolution (GaDE) (Rank 3) [319]

3. CEC 2012 LSGO

- (a) Multiple Offspring Sampling (MOS) - 2012 (Rank 1) [182]
- (b) Self-adaptive Differential Evolution algorithm with small and varying population size (jDEsps) (Rank 2) [38]

4. CEC 2013 LSGO

- (a) Multiple Offspring Sampling (MOS) - 2013 (Rank 1) [183]
- (b) Differential Evolution with Cooperative Co-evolution - Grouping (DECC-G) (Rank 2) [322]

-
- (c) Cooperative Co-evolution with Co-variance Matrix Adaptation Evolutionary Strategy (CC-CMA-ES) (Rank 3) [193]

Statistical analysis is performed to test the effectiveness of the proposed algorithm. Table 4.4 presents how many times numerical rank 1 was obtained by all the algorithms (nBest) over 15 functions. In Table 4.4, the nBest value for the proposed algorithm is one and for the MOS-CEC2013 is eight, which is the highest.

In order to confirm whether the superior performance of algorithms is due to the strength of the algorithms or merely due to chance of happening. Therefore, a statistical ranking of all the algorithms is performed using two multiple comparison statistical tests viz., Friedman’s test and Quade’s test. Friedman’s test ranked the proposed algorithm at an absolute rank of 7.87.. The lowest absolute rank for Friedman’s test is 3.23 for MOS-CEC2013. Now to consider variability in ranking, Quade’s test was performed. Quade’s test gave a absolute rank of 6.65 for the proposed algorithm and a lowest absolute rank of 2.65. There is no absolute rank of 1 in both tests which shows that no algorithm is a clear winner. Quade’s test also suggests the same as Friedman’s test from p-values that there is a difference among the algorithms, which can be seen in the rankings of the tests presented in Table 4.4. Therefore, from the above two statistical tests we conclude that our proposed algorithms is in absolute rank of 7.87 and 6.65 which is close to 8 or 7 out of 11, which is creditworthy performance.

The results of the proposed algorithm along with the other compared algorithms are presented in Tables 4.5 and 4.6. The functions, wherein an algorithm beats the other algorithms, are marked in bold. The proposed algorithm is run for fixed FEs as per the CEC 2013 LSGO benchmark (i.e. 3 million FEs for each of the 25 independent runs). Median value of 25 independent runs (i.e. 13th) is used for the statistical analysis as per the benchmark. The winner is decided based on the median value of 25 independent runs (i.e. 13th in the increasing order) for each function. See Appendix C for additional results analysis.

Apart from statistical ranking, a score-based ranking of algorithms was also performed. As proposed in CEC 2015 LSGO, the ranking was performed based on a median value of fitness value over 25 runs for each benchmark function for all algorithms. The median rankings are presented in Tables 4.5 and 4.6.

Table 4.4: Number of best values and ranking based on statistical tests

Algorithm	nBest	Friedman's Test (Ranking)	Quade's Test (Ranking)
IMHS+MDE	1	7.87	6.65
MA-SW-Chains	2	4.47	3.98
2S-Ensemble	1	4.9333	4.9625
MOS-SOCO2011	2	6.7667	7.7125
jDElscoP	2	6.0667	6.9375
GaDE	2	7.7	8.5125
MOS-CEC2012	2	5.5	4.5625
jDEsps	0	5.4667	5.9417
MOS-CEC2013	8	3.2333	2.6542
DECC-G	0	8.4	8.5583
CC-CMA-ES	0	5.6	5.5333
Statistic Value		33.50	4.71
p-value		0.0001946	1.275e-05

Table 4.5: Ranking based on Medians (As performed in CEC 2015 LSGO) for benchmarks F1 to F8

Algorithm	F1	R1	F2	R2	F3	R3	F4	R4	F5	R5	F6	R6	F7	R7	F8	R8
IMHS+MDE	4595495	11	1.56E+03	10	2.06E+01	11	9.74E+09	7	9.32E+05	1	1.05E+06	11	1.42E+08	7	1.53E+14	6
MA-SW-Chains	6.15E-13	8	1.13E+03	7	6.79E-13	7	2.70E+09	5	1.98E+06	3	6.24E+02	3	3.99E+06	5	4.65E+13	4
2S-Ensemble	0.00E+00	1	9.95E-01	2	3.98E-13	6	1.13E+10	9	1.28E+06	2	2.00E+05	6	1.54E+06	4	3.94E+14	9
MOS-SOCO2011	0.00E+00	1	1.96E+01	4	0.00E+00	1	1.20E+10	10	1.09E+07	9	9.85E+05	8	1.23E+07	6	1.13E+15	10
jDElscoP	0.00E+00	1	9.95E-01	2	2.77E-13	5	7.18E+09	6	4.83E+06	5	1.99E-01	1	3.17E+09	11	1.05E+14	5
GaDE	0.00E+00	1	0.00E+00	1	2.56E-13	4	1.01E+10	8	1.02E+07	8	1.03E+06	10	2.68E+09	10	1.57E+14	7
MOS-CEC2012	0.00E+00	1	1.75E+03	11	2.35E-12	9	3.41E+08	2	2.37E+07	10	9.88E+05	9	1.78E+05	2	5.31E+11	1
jDEsps	2.88E-23	7	9.15E+01	5	1.10E-13	2	2.56E+09	4	3.11E+06	4	7.07E-01	2	1.14E+09	9	4.76E+12	3
MOS-CEC 2013	0.00E+00	1	8.24E+02	6	1.69E-12	8	7.80E+07	1	6.95E+06	6	1.39E+05	4	1.10E+03	1	2.82E+12	2
DECC-G	2.33E-06	10	1.31E+03	8	1.19E+00	10	2.08E+11	11	8.19E+06	7	1.77E+05	5	9.10E+08	8	6.10E+15	11
CC-CMA-ES	5.56E-09	9	1.34E+03	9	1.49E-13	3	1.88E+09	3	7.28E+14	11	2.06E+05	7	1.36E+06	3	2.92E+14	8

Table 4.6: Ranking based on Medians (As performed in CEC 2015 LSGO) for benchmarks F9 to F15

Algorithm	F9	R9	F10	R10	F11	R11	F12	R12	F13	R13	F14	R14	F15	R15
IMHS+MDE	2.52E+08	3	92948392	11	1.72E+10	7	33536.33	11	4.27E+09	7	4.5E+10	7	36003393	8
MA-SW-Chains	1.16E+08	1	3.32E+02	1	2.10E+08	4	1.24E+03	8	1.91E+07	3	1.47E+08	5	5.76E+06	3
2S-Ensemble	1.34E+08	2	1.44E+07	6	2.45E+08	5	1.66E+03	9	2.14E+08	4	5.68E+07	4	2.37E+06	2
MOS-SOCO2011	9.16E+08	10	8.78E+07	9	1.35E+09	6	7.01E+01	2	8.37E+08	6	2.30E+09	6	1.21E+08	11
jDElscoP	3.67E+08	6	1.01E+03	3	1.22E+11	9	9.74E+02	5	3.48E+10	11	4.89E+11	11	3.43E+07	7
GaDE	6.78E+08	9	2.29E+07	8	2.41E+11	11	1.11E+03	7	2.87E+10	9	4.19E+11	10	1.09E+08	10
MOS-CEC2012	1.65E+09	11	9.01E+07	10	2.53E+07	2	1.68E+02	3	5.90E+06	2	4.80E+07	3	7.25E+06	4
jDEsps	2.74E+08	4	9.21E+02	2	2.74E+10	8	7.89E+02	4	3.12E+10	10	1.80E+11	9	4.59E+07	9
MOS-CEC2013	4.18E+08	7	1.17E+06	5	1.71E+07	1	1.56E+01	1	1.02E+06	1	1.28E+07	1	1.68E+06	1
DECC-G	5.32E+08	8	2.15E+07	7	1.25E+11	10	4.20E+03	10	8.66E+09	8	1.23E+11	8	1.19E+07	5
CC-CMA-ES	3.57E+08	5	7.48E+05	4	9.79E+07	3	1.03E+03	6	4.12E+08	5	3.66E+07	2	3.05E+07	6

Table 4.7: F1 racing points table for score calculation

Formula One (F1) Points Table	
Place	Points
11th onwards	0
10th	1
9th	2
8th	4
7th	6
6th	8
5th	10
4th	12
3rd	15
2nd	18
1st	25

Formula one (F1) based scoring (see Table 4.7) was performed after the algorithms were ranked according to median. Any function that is ranked above ten is given a zero score. A summation, (all the scores of the benchmark functions for a particular algorithm) was performed, and later ranking was done based on these scores. The highest F1 score has the lowest rank (1), and the lowest F1 score has the last rank.

To better understand in which functionality grouping (separable, partially separable or non-separable) the proposed algorithm is performing well, we have also calculated the F1 score for each algorithm based on functionality grouping. The F1 scores have been presented in Table 4.8. The F1 score for functionality grouping suggests that the proposed algorithm is performing moderately well for partially separable functions, for overlapping functions, and for non-separable functions, not performing well for separable functions.

In this score-based ranking, MOS-CEC2013 algorithm, which is the winner of all the competitions, is higher in the ranking than the proposed algorithm. Our proposed hybrid is better than MOS-SOCO2011 (Rank 1 of SOCO 2011 Competition), GaDE (Rank 3 of SOCO 2011 Competition), and DECC-G (Rank 2 of CEC 2013 LSGO competition).

Table 4.8: F1 Score based on Median Ranking (As given in CEC 2015 LSGO)

Functions	Separable Functions		Partially Separable Functions and Overlapping Functions										Non-separable Functions	
	F1, F2, F3		F4, F5, F6, F7	F8, F9, F10, F11		F12, F13, F14		F15						
Algorithm	S1	R1	S2	R2	S3	R3	S4	R4	S5	R5				
IMHS+MDE	1	11	37	7	29	8	12	8	4	8				
MA-SW-Chains	16	9	50	2	74	1	29	5	15	3				
2S-Ensemble	51	4	40	5	38	5	26	6	18	2				
MOS-SOCO2011	62	1	15	10	12	9	34	4	0	11				
jDElscop	53	3	43	4	35	7	10	9	6	7				
GaDE	62	1	10	11	12	9	9	10	1	10				
MOS-CEC2012	27	7	39	6	44	4	48	2	12	4				
jDEsps	34	6	44	3	49	3	15	7	3	9				
MOS-CEC2013	37	5	70	1	59	2	75	1	25	1				
DECC-G	6	10	20	9	11	11	9	10	10	5				
CC-CMA-ES	19	8	36	8	37	6	36	3	8	6				

4.6 Conclusions and Future Directions

We proposed a novel hybrid meta-heuristic and tested its effectiveness on CEC 2013 LSGO benchmarks. Statistical tests such as Quade's and Friedman's tests were conducted on the compared algorithms. Apart from these statistical tests, F1 based scoring was also performed on the compared algorithms. The experimental results consistently show that our proposed hybrid meta-heuristic performs statistically on par with some algorithms in a few problems, while it turned out to be the best in a couple of problems. In future, adaptive versions of the optimization algorithms need to be utilized with less parameters to tune as the proposed optimization algorithm even though able to converge to optimum fast but only at the expense of extensive tuning of parameters. Also, algorithms with less or no parameters and faster convergence to the optimum need to be investigated.

Chapter 5

Feature Subset Selection using Adaptive Differential Evolution: An application to banking

5.1 Introduction

The Feature Subset Selection (FSS) refers to the method that involves selecting significant subset of important features for the purpose of model development. The FSS is required for reasons [116] such as model simplification, human comprehensibility, smaller model building times, maintaining immunity from the curse of dimensionality and lastly, for the models to generalize well and reduce over-fitting. Apart from the above reasons, the FSS techniques can be employed when there exist several features greater than the data points.

FSS techniques are employed together with any evaluation measure that basically involves scoring the subsets [116]. Moreover, it calls for an exhaustive exploration of search space which is considered computationally costly for searching/selection of feature subsets. The performance metric has an impact upon the FSS approach and it's with such measures one could easily group the FSS techniques as wrapper-based, embedded and lastly, filter-based.

- Wrapper-based FSS makes use of an optimization technique and a classification model for evaluating the feature subsets. Additionally, feature subset selection technique makes use of a ten-fold cross-validation, hold-out and others. The

wrapper based FSS is computationally expensive. Nevertheless, they offer superior results.

- Filter-based FSS uses a statistical measure rather than a classification framework for finding feature subsets. Such techniques are computationally less expensive and also less precise. For instance, please consider the t-statistic based model in [255].
- Embedded FSS makes use of models that internally, have the ability of performing feature selection. An example is LASSO regression for developing linear models, where the features with non-zero regression coefficients are chosen by the LASSO [301]. Other techniques that fall in this category are Decision Tree [37], Neural Networks [219, 253], and etc.

Algorithms used to search the features can be meta heuristics like Simulated annealing (SA), Genetic algorithm (GA), Particle swarm optimization (PSO), etc. Among all features, only a subset of features is chosen by the search algorithm. Models are built essentially based upon the training dataset and verified on the test data by making use of the evaluation measure. This procedure is repeated until suitable features set is found based upon the evaluation measure.

5.2 Motivation

The motivation is as follows:

- Development of an adaptive and novel wrapper-based FSS algorithm, where there will be no need to manually tune parameters of the wrapper and also comparing it with extant FSS techniques in [57, 256].
- To employ an effective evaluation measure i.e. Matthews Correlation Coefficient (MCC) for the wrapper.
- Suggesting a host of classifiers for the FSS.

5.3 Related Work

In this section, only Evolutionary Computing (EC) -based FSS are considered. Moreover, only examined research wherein FSS is applied on credit scoring particularly Australian along with German datasets and on Financial Statement fraud dataset [256].

5.3.1 EC-based FSS using classification techniques

There exist many EC-based FSS techniques, making use of considerable classification techniques and the list is comprehensive. As a result, few suitable FSS techniques are reviewed in this section. In [241], wrapper based on GA was basically devised with All paired Support Vector Machine and classification accuracy as an evaluation measure. Moreover, in [223], a hybrid GA-based wrapper was devised by making use of K-Nearest Neighbors and classification accuracy as an evaluation measure. Additionally, in [266], GA-based wrapper was developed by making use of Decision Tree and classification accuracy (10-fold) as an evaluation measure. Then, within [142], GA-based wrapper was developed by making use of K-Nearest Neighbors and classification accuracy as the evaluation measure. Within [17], PSO along with GA-based wrapper was developed with Support Vector Machine and classification accuracy (10-fold) as an evaluation measure. Further, within [66], Tabu Search + PSO-based wrapper was devised by making use Support Vector Machine and K-Nearest Neighbors along with Euclidean Distance as an evaluation measure. Within [306], Simulated Annealing-based wrapper was devised by making use of Naive Bayes with classification accuracy (5-fold) as an evaluation measure. In [318], the GA-based wrapper was devised by making use of Support Vector Machine and specificity considered as the evaluation measure. In [333], Binary PSO + Mutation-based wrapper was devised by making use of Decision tree along with weighted cost as an evaluation measure. Within [288], Quantum-inspired Differential Evolution wrapper was devised by making use of Logistic Regression as the classifier along with accuracy as the fitness function or evaluation measure.

Taking into consideration the above discussion, it can be clearly stated that the wrappers based on GA, classification accuracy as the evaluation measure along with Support Vector Machine as classifier are frequently employed for feature selection by the researchers.

5.3.2 Credit Scoring and Financial Statement Fraud

These two [57, 129] are basically regarded as the seminal papers wherein feature selection is employed for solving the credit scoring problem. The following works wherein data mining approaches are employed for solving the financial statement fraud: [58, 92, 119, 155, 157, 222, 232, 256, 272, 325, 338]. Within [245], rule-based classifiers are developed by making use of evolutionary computing methods detecting financial statement frauds.

5.4 Classification Algorithms and Evaluation Measures

5.4.1 Classifiers Employed for the proposed FSS

5.4.1.1 Probabilistic Neural Network

Probabilistic Neural Network (PNN) was developed in the year 1990 by Specht [283]. It deals with the problem of automatic learning in one pass. From the samples of feature vectors, PNN classifies the samples into classes based on the distribution of samples. Moreover, this is a neural network that holds the advantage of speedy generalization. PNN employs four different layers i.e. Input layer, Pattern layer, Summation layer and Output layer.

5.4.1.2 Logistic Regression

Logistic Regression (LR) is a statistical model, which is usually employed for a two class problems [311]. It based upon estimating regression model's parameters ' β '. This model is one in which the probability of log-odds of any occurrence as a linear combination of independent variables i.e $x_1, x_2, \dots, x_n$. Additionally, the two classes of the dependent or output variable is often transformed to 0 and 1 that for instance are yes/no. It must be noted that $p(x)$ is the possible likelihood of the output variable leveled to either a '1' or a '0'.

$$p(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \dots + \beta_n x_n)}}$$

5.4.1.3 Naive Bayes

Naive Bayes (NB) [258] finds the conditional a-posterior probabilities associated with a output variable provided independent predictor variables with the Bayes rule. Moreover, the algorithm is developed on a highly powerful assumption that the features independent of one another. It holds a benefit of being developed upon a robust probabilistic basis. The following equation assigns a class label to every sample i.e. $\hat{y} = C_k$.

$$\hat{y} = \underset{k \in \{1, \dots, K\}}{\operatorname{argmax}} p(C_k) \prod_{i=1}^n p(x_i | C_k)$$

5.4.1.4 Support Vector Machine

Support Vector Machine (SVM) finds a maximum margin separating hyperplane [265]. As a result, this issue is modeled as a constrained optimization problem. Solving the primal is difficult, thus optimization problem is transformed into dual form having linear constraints and resolved by making use of quadratic programming. Additionally, for non-linear classification issues kernel trick is adopted. Within the kernel trick, kernels like linear kernel, radial-basis kernel, polynomial kernel and others can be utilized for the purpose of solving the non-linear classification. The dual form of the kernel is mentioned within the below equations.

$$\begin{aligned} & \text{maximize } f(c_1 \dots c_n) \\ & = \sum_{i=1}^n c_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n y_i c_i k(\vec{x}_i, \vec{x}_j) y_j c_j \\ & \text{subject to } \sum_{i=1}^n c_i y_i = 0, \text{ and } 0 \leq c_i \leq C \text{ for all } i \end{aligned}$$

In this $k(\cdot)$ is considered as being the kernel function, ‘ $\mathbf{x}$ ’ are the input variables, ‘ C ’ is the regularization parameter and lastly, ‘ y ’ is the output variable. The dual variables c_i ’s are resolved by making use of SMO (Sequential Minimal Optimization) and radial basis kernel is adopted in the proposed FSS.

5.4.2 Evaluation Measures used in the Proposed FSS

As per the confusion matrix [276], TP basically refers to the genuine positives, TN refers to the genuine negatives, FP the false positives and lastly, FN the false negatives.

5.4.2.1 Sensitivity

Sensitivity [276] (termed as the true positive rate (TPR) or the recall), calculates the percentage of positives which are correctly recognized.

$$\text{Sensitivity} = \frac{TP}{TP + FN}$$

5.4.2.2 Specificity

Specificity [276] (also termed as the true negative rate (TNR)), calculates the percentage of negatives which are correctly recognized.

$$\text{Specificity} = \frac{TN}{TN + FP}$$

5.4.2.3 Accuracy

Accuracy [276] essentially calculates the percentage of negatives and positives which are correctly recognized.

$$\text{Accuracy} = \frac{TP + TN}{TP + FN + TN + FP}$$

5.4.2.4 Area Under Curve

The area under the ROC curve (AUC) [276] is average of specificity and sensitivity for binary classification problem and is highly robust as compared to the accuracy measure for imbalanced datasets.

$$\text{AUC} = \frac{\text{Sensitivity} + \text{Specificity}}{2}$$

5.4.2.5 Matthews Correlation Coefficient

Matthews Correlation Coefficient (MCC) [32, 208] is computed between the actual and predicted binary classes. This generates a value between -1 and +1. Additionally, a

coefficient of +1 conveys an exact classification, 0 conveys an arbitrary classification and -1 conveys an exact opposite classification.

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

5.5 Evolutionary Computing

5.5.1 Differential Evolution

Differential Evolution (DE) [293] has fundamentally four sections initialization, recombination, mutation and lastly, selection. Moreover, at individual iteration, new solutions are generated with the combination of solutions. Suppose a, b, c are the three solution vectors that have been selected randomly from any population of solutions ‘NP’ and the subsequent equations are employed for the purpose of crossover and mutation.

$$If(rand(0,1) < CR \parallel i = R) \text{ then } u_i = a_i + F \times (b_i - c_i)$$

$$If(rand(0,1) > CR \& i \neq R) \text{ then } u_i = x^{current}$$

In this, DE/rand/1/bin variant is adopted. Differential weight, ‘F’, lies between 0 and 2. Crossover Rate ‘CR’, lies between 0 and 1.

5.5.2 Adaptive Differential Evolution

Adaptive Differential Evolution (ADE) is considered as changing DE with basically two new concepts. First concept is Roulette Wheel Selection [25] of a parameters set i.e. Differential Weight (F) along with Crossover Rate (CR) from 12 sets based on the success rate of the selection and second concept is random localization [144]. Within [304], the Roulette Wheel Selection is employed in order to select the better parameter set based on success rate of the selection of each parameter set. Additionally, all the twelve parameter sets use random localization at the time of mutation. At the time of adaptation procedure, the twelve different F and CR values are recognized through 12 different probabilities q_h , $h = 1, 2, \dots, H$ (where H signifies 12). Such likelihoods are changed through the success rate during the preceding iterations of the considered 12

parameters. In case if $f(y) > f(x_i)$ then the h -th setting is regarded as successful.

$$q_h = \frac{n_h + n_0}{\sum_{j=1}^H (n_j + n_0)}$$

where n_h is regarded as the present success count of h -th setting and $n_0 > 0$ is regarded as a constant. In addition, the $n_0 > 1$ setting avoids a dramatic change in q_h through one random successful application of the h -th parameter settings. Further, the q_h values essentially are reset to $1/H$ when the values of probability go down.

In [144], a random localization approach is introduced for DE. Instead of selecting randomly the current vector, select the best among the three randomly selected vectors. This saves around 35% of the function evaluations [144]. The mutant vector ‘u’ is basically computed from three randomly chosen vectors including ‘a’, ‘b’ and ‘c’. Amongst these, ‘a’ is seen as the most effective vector with respect to fitness value.

$$u_i = a_i + F \times (b_i - c_i)$$

x_{new} is calculated from the mutant vector ‘u’ and the original vector through carrying out either exponential or binomial crossover. The reason behind selecting the adaptive version of DE is dynamic automatic selection of DE’s parameters. Further, this assists in the better and faster convergence along with performance.

5.6 Proposed ADE-based Feature Subset Selection

The above suggested feature subset selection based on ADE makes use of four different classification algorithms i.e. PNN, LR, NB and SVM with MCC that is the evaluation measure or fitness function. Besides calculating MCC for the chosen features set, evaluation measures like accuracy, AUC, sensitivity and specificity are also calculated for comparison with other research works. Please refer Algorithm 4 for the detailed implementation steps and for the process flow the schematic diagram in Figure 5.1.

Further, the binary discretization is carried out through means of Sigmoid function. The same has been provided in the below equation:

$$Sigmoid(x) = \frac{1}{1 + e^{-x}} = \frac{e^x}{e^x + 1}$$

Algorithm 4: Adaptive Differential Evolution (ADE)–FSS

Input: Population, NP, F and C

$I = 1$

$J = 1$

ADE-FSS(X , Y)

{

while $I \leq \text{MaxIterations}_{\text{ADE-FSS}}$ **do**

while $J \leq NP$ **do**

if $y_J < MCC_{Th}$ **then**

*Perform Roulette Wheel Selection to choose between
the 12 parameter sets (of CR and F) based*

on the success rate of each set of parameters

Choose solution vectors a, b, c randomly

as mutant vectors from $X_{Continuous}$

Select the best among three

as the current solution vector (Random Localization)

Perform Binomial or Exponential Crossover

Perform Binary Discretization of the

new solution vector x_{new} using Sigmoid Function

$\hat{x}$ is the new binary solution vector

(Apply Classification Algorithm on Selected Features)

$\hat{y} = \text{ComputeFitness-MCC}(\hat{x})$

if $\hat{y}$ better than the fitness of current solution vector in X **then**

Replace current vector by $\hat{x}$ in X_{Binary}

Replace current vector by x_{new} in $X_{Continuous}$

Also replace the corresponding fitness value with $\hat{y}$

end

end

$J = J + 1$

end

$I = I + 1$

end

return (X , Y)

}

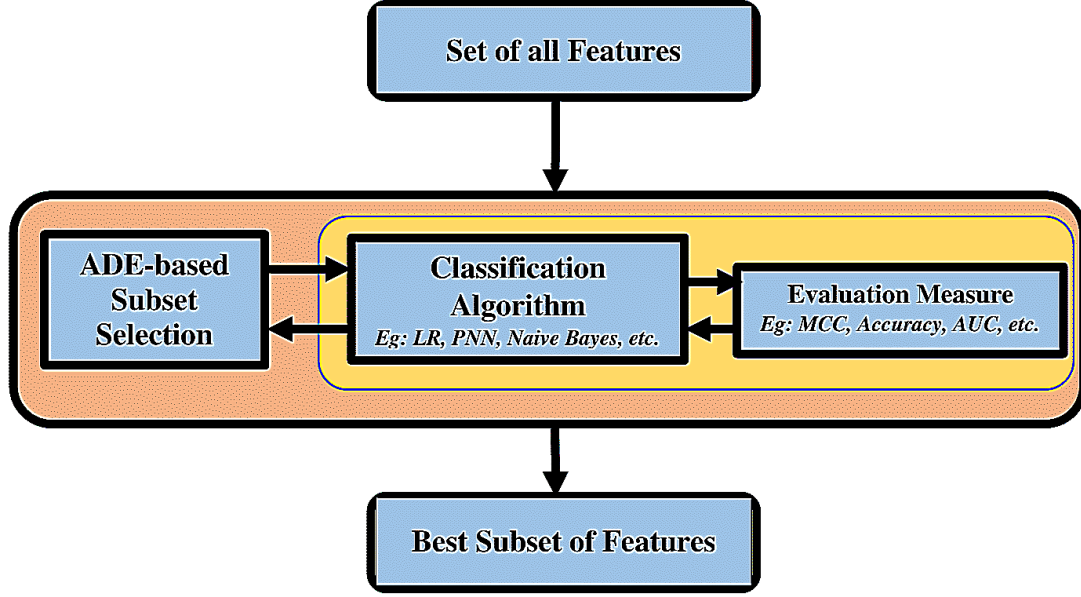


Figure 5.1: Adaptive Differential Evolution based FSS

If the $Sigmoid(x_{new}^i) > Rand(0, 1)$ then $\hat{x}$ is assigned a value of '1' or otherwise it is assigned a '0' value.

5.7 Dataset Description

Three different datasets are utilized for the suggested ADE-FSS algorithm. Moreover, out of three, two of them involve credit scoring and the third involves detection Financial Statement Fraud dataset. Such datasets have an implication to the banking domain. Two credit scoring datasets belonging to UCI Machine Learning Repository are utilized i.e. German Credit Scoring dataset [7] and Australian Credit Scoring dataset [6]. First, the German Credit Scoring dataset involves around 700 diverse creditworthiness samples along with 300 non-credit worthiness samples. This means that an overall of 1000 samples together with 24+1 features are present within the German Credit Scoring dataset. Numeric version of the German Credit Scoring dataset is considered for the experimental study. Moreover, the Australian Credit Scoring dataset includes 383 'bad' credit scoring samples and 307 'good' credit scoring samples. An overall of 690 samples together with 14+1 features are present in the Australian Credit

Scoring dataset. The Financial Statement Fraud dataset involve 35 financial items as mentioned within [57, 256]. The dataset consists of 101 deceitful and 101 non-deceitful companies. As a result, the dataset has an overall of 202 samples. During the pre-processing phase, it is witnessed that just few of the financial ratios within the dataset hold a large range. As a result, a natural logarithmic transformation is applied followed by normalization.

5.8 Results and Discussion

The ADE and DE-based FSS are run for 30 runs as they are stochastic. The maximum number of iterations for both DE and ADE-based FSS are set at 3000. For DE, CR is set at 0.47 and F is set at 0.78. But for ADE, there is no need to tune the control parameters as it is adaptive. The population size is kept at 30. The proposed FSS techniques are contrasted with the techniques in [57, 256]. The results considerably highlight that the proposed FSS technique are superior to the compared techniques. MCC is employed as the objective for the ADE. Besides calculating MCC, other evaluation measures like accuracy and AUC are too calculated for the comparison with other FSS methods. It is observed that ADE-SVM yielded superior results as compared to the other suggested approaches. Moreover, ADE-SVM yielded the MCC values of 0.64, 0.88 and 0.96 for German, Australian credit scoring and financial statement fraud datasets correspondingly. This could be clearly observed from the Tables 5.1, 5.2 and 5.3. ADE-SVM yielded an AUC values of 0.791, 0.921 and 0.98 for German, Australian credit scoring along with financial statement fraud datasets correspondingly. ADE-SVM yielded accuracies of 85.3, 94.19 and 98.02 for German, Australian credit scoring and financial statement fraud datasets correspondingly. ADE-based wrapper techniques generated less cardinality, superior AUC and accuracy values as compared to the state-of-art techniques along with DE based wrappers. This essentially establishes that ADE is an effective optimizer for the purpose of finding the optimum feature subsets. In the ten folds, the best evaluation measures are averaged.

Taking a step ahead, t-Test is carried out on MCC values of 10 different folds. It is observed that ADE-SVM provides better results in comparison with the other chosen frameworks. As a result, a pairwise t-test is carried out at one percent significance value along with 18 degree of freedom with ADE-SVM contrasted with all the other

Table 5.1: Results of German Credit Scoring Dataset

Model	MCC (Avg.)	AUC (Avg.)	Acc. % (Avg.)	Average Cardinality
ADE-PNN	0.60	0.817	80.80	8.3
ADE-LR	0.6	0.767	84.10	12.3
ADE-NB	0.60	0.797	83.3	9.9
ADE-SVM	0.64	0.791	85.3	7.4
DE-PNN	0.61	0.821	81.8	8.5
DE-LR	0.61	0.772	84.6	12.4
DE-NB	0.61	0.803	83.9	12.2
DE-SVM	0.63	0.789	85.0	7.9
All+SVM [57]	-	0.729	75.40	24
LDA+SVM [57]	-	0.736	76.10	12
DT+SVM [57]	-	0.712	73.70	12
RST+SVM [57]	-	0.730	75.60	12
F-score+SVM [57]	-	0.745	76.70	12

Table 5.2: Results of Australian Credit Scoring Dataset

Model	MCC (Avg.)	AUC (Avg.)	Acc. % (Avg.)	Average Cardinality
ADE-PNN	0.83	0.916	91.45	6.9
ADE-LR	0.82	0.911	91.02	6.9
ADE-NB	0.82	0.910	91.16	5.2
ADE-SVM	0.88	0.921	94.19	6.3
DE-PNN	0.82	0.915	91.30	6.5
DE-LR	0.82	0.912	91.15	6.8
DE-NB	0.82	0.912	91.302	5.7
DE-SVM	0.88	0.943	94.19	6.7
All+SVM [57]	-	0.843	84.34	14
LDA+SVM [57]	-	0.881	86.52	7
DT+SVM [57]	-	0.862	86.29	7
RST+SVM [57]	-	0.854	85.22	7
F-score+SVM [57]	-	0.852	85.10	7

Table 5.3: Results of Financial Statement Fraud Dataset

Model	MCC (Avg.)	AUC (Avg.)	Acc.% (Avg.)	Average Cardinality
ADE-PNN	0.89	0.941	94.07	12.3
ADE-LR	0.88	0.935	93.62	9.6
ADE-NB	0.77	0.877	87.69	9.6
ADE-SVM	0.96	0.980	98.02	9.9
DE-PNN	0.90	0.950	95.04	13.4
DE-LR	0.90	0.945	94.61	11.3
DE-NB	0.80	0.890	88.14	9.7
DE-SVM	0.96	0.980	98.02	10
t-stat+MLFF [256]	-	0.791	78.77	18
t-stat+SVM [256]	-	0.735	73.41	18
t-stat+GP [256]	-	0.929	92.68	18
t-stat+GMDH [256]	-	0.908	90.68	18
t-stat+LR [256]	-	0.708	70.36	18
t-stat+PNN [256]	-	0.927	95.64	18

Table 5.4: t-Test Results of German Credit Scoring Dataset

Compared FSS Techniques	P-value	Sig. T/F
ADE-SVM vs ADE-PNN	0.0054	T
ADE-SVM vs ADE-LR	0.0274	F
ADE-SVM vs ADE-NB	0.0037	T
ADE-SVM vs DE-SVM	0.5785	F
ADE-SVM vs DE-PNN	0.04	F
ADE-SVM vs DE-LR	0.2186	F
ADE-SVM vs DE-NB	0.1	F

Table 5.5: t-Test Results of Australian Credit Scoring Dataset

Compared FSS Techniques	P-value	Sig. T/F
ADE-SVM vs ADE-PNN	0.0876	F
ADE-SVM vs ADE-LR	0.0717	F
ADE-SVM vs ADE-NB	0.0674	F
ADE-SVM vs DE-SVM	0.9471	F
ADE-SVM vs DE-PNN	0.0932	F
ADE-SVM vs DE-LR	0.0969	F
ADE-SVM vs DE-NB	0.1075	F

Table 5.6: t-Test Results of Financial Statement Fraud Dataset

Compared FSS Techniques	P-value	Sig. T/F
ADE-SVM vs ADE-PNN	0.0588	F
ADE-SVM vs ADE-LR	0.0219	F
ADE-SVM vs ADE-NB	0.0001	T
ADE-SVM vs DE-SVM	0.8509	F
ADE-SVM vs DE-PNN	0.0955	F
ADE-SVM vs DE-LR	0.0639	F
ADE-SVM vs DE-NB	0.0006	T

suggested techniques considered two at a time. In case if p-value is lower than 0.01, the null hypothesis would be rejected thus asserting that the pair of FSS frameworks compared are statistically not equal. Otherwise, in case if the p-value is above 0.01, the null hypothesis isn't rejected, suggesting that the pair of FSS frameworks compared are equal statistically. For the outcomes to be statistically significant, the p-value needs to be below 0.01. Results of the t-Test are provided in Tables 5.4, 5.5 and 5.6, having significance value of 'T', when p-value is lower than 0.01 and 'F' as false, when the p-value is above 0.01. Further, for all three datasets, t-Test outcomes in 4 cases (i.e. ADE-SVM vs ADE-LR, ADE-SVM vs DE-SVM, ADE-SVM vs DE-PNN and ADE-SVM vs DE-LR) of the contrasted FSS frameworks are found to be statistically equal.

Within [65, 336], computational complexities of LR, NB, PNN and SVM are presented. The computational complexity associated with NB is $O(Nd + dC)$, wherein 'N' signifies the training samples, 'd' signifies the overall features and 'C' signifies the overall classes within the dataset [65]. Additionally, the computational complexity associated with LR is $O(Nd^2 + d^3)$, wherein 'N' is regarded as the overall training samples, 'd' is considered as the overall features [65]. LR too performs optimization such as steepest descent, Newton Raphson and others for the purpose of discovering LR parameters that essentially makes it slow with respect to running time. In contrary, PNN's time complexity relies upon the selection of the kernel along with the smoothing factor. In case, if the smoothing factor exceeds one it requires exponential time even though it is considered as a single pass algorithm. Additionally, PNN involves high memory requirement since all the samples are kept at every node within the pattern layer and storage of samples at individual node also acquires some time that is ignored.

Apart from this, the time complexity for SVM relies completely upon parameter ‘C’ together with training samples i.e. for the large ‘C’, time complexity is about N^3 and for small ‘C’, time complexity is about N^2 . As a result, training SVM on large datasets is considered as highly complex as compared to training SVM with small datasets. Based upon the above discussion related to computational complexity and discussion in section 5.4.1, the expert could adopt any of the four classification algorithms he/she prefers for the purpose of FSS, since few are found to be statistical equal.

Further, a contrast between the evaluation measures (i.e. MCC, AUC and accuracy) is carried out within [32], in which the imbalanced datasets are studied and the results suggest that AUC along with MCC are the robust evaluation measures. Additionally, researchers in [32] also point towards the fact that MCC is preferred for the reason that it has a closed form formula and is robust while AUC has no closed form formula.

5.9 Conclusions and Future Directions

To conclude, ADE-based FSS is proposed making use of four classification algorithms with MCC as the fitness function. Additionally, the ADE-FSS is essentially employed on three different datasets of banking domain i.e. fraud detection and credit scoring. This is certainly the first case wherein MCC is employed as a fitness function or evaluation measure for FSS. Further, one among the classification algorithms can be fixed based upon the results obtained within this paper and several EC techniques can be employed. In future, multi-objective optimization techniques can be applied to the problem, where apart from MCC other objective functions can be considered in the framework of Minimum Redundancy and Maximum Relevance (MRMR).

Chapter 6

Rule-based Classification via Differential Evolution for the banking applications

6.1 Introduction

The term rule-based classification refers to a classification method that makes use of ‘IF-THEN’ rules for class detection. Rule-based classification extracts ‘IF-THEN’ rules from the training data. An ‘IF-THEN’ rule is expressed as: “IF antecedent conditions THEN Class”. The benefits of rule-based models are: 1) easy to interpret, 2) easy to generate, 3) can classify new instances rapidly, etc.

Credit scoring is an important task in the banking sector, where the customer information is collected and analyzed for avoiding incorrect decisions in granting loans. A sound model rather than intuitive experience would assist the managers. Customers’ credit forecast is paramount within the credit sector. Moreover, with the rapid progress within this sector, credit scoring frameworks have been extensively adopted for the credit assessment. Additionally, credit scoring frameworks are devised for distinguishing consumers belonging to the non-creditworthy class (rejected) or creditworthy class (accepted) by analyzing their associated geographic, demographic, transactional, psychographic and behavioural attributes including bureau information. If credit scoring is performed meticulously with the help of data-driven analytical models, it will result in less non-performing assets, later on, thereby saving the banking system from huge

losses.

Additionally, financial fraud is another scourge that is plaguing the industry. There has been a significant rise in the overall financial scams over the last decade. The Lehman Brothers Scandal [1, 5] during the year 2008 within the United States as well as Satyam scandal [4] during the year 2009 within India are amongst the nastiest accounting calamities that have taken place during the latest times. During the year 2007, Lehman Brothers were positioned at the leading position within the “America’s Most Admired Companies 2007” [2] through Fortune Magazine. Only after a year, the corporation had gone insolvent, and it was discovered that the company had concealed around 50 billion dollars in loans masked as sales. Moreover, it was a stark situation of an accounting deception. The Satyam Computer Services calamity was a corporate scandal, which appeared within India during the year 2009 where the Chairman admitted the fact that the accounts of the company had been counterfeit. The global corporate society was surprised at the time when the chairman declared that he had made changes to the accounts by around US\$ 1.47 billion [4]. Such frauds could have been avoided had appropriate audits been in place. Although there are auditors for the job, because of human subjectivity and other circumstances in which they handle the information, it is virtually impossible for them to provide accurate assessment always.

Fortunately, here, the Data Mining (DM) techniques come to our rescue. Various DM techniques require optimisation at different stages, for example, in feature selection or in tuning a classifier/regression model or in deciding the ideal parameter that would bring about precise outcomes for the technique [231]. Also, evolutionary computing (EC) techniques can be utilised independently for solving different data mining tasks, for example, classification as elucidated in [168]. Given the inherent flexibility of EC techniques, it is clear that EC techniques alone can be utilised to solve different data mining tasks.

Likewise, in this chapter, develop, i.e. a Differential Evolution and an Adaptive Differential Evolution based rule miners for generating ‘IF-THEN’ rules, which classify the customers into creditworthy or not and corporations into fraudulent or not. Rules are of high interest for the end users because they are easily interpretable and offer an easy understanding of the knowledge produced by the models. With appropriate rules, examined features that are more significant in deciding creditworthiness or frauds and also taking proper measures from the early indicators given by the rules.

6.2 Motivation

The motivation of the present work is as follows:

1. Noticed a gap in using EC techniques for mining classification rules with lower and upper bounds specified for the attributes present in the antecedent of the rules. Therefore, a novel EC based rule miner is proposed with a new rule encoding scheme.
2. The earlier studies could not show the power and versatility of EC techniques. Therefore, these techniques are explored in this front.
3. The use of Differential Evolution and its variants for developing a rule-based classifier to solve banking applications, namely credit scoring and detecting financial statement fraud is conspicuously absent. This chapter addresses this gap.
4. The reasons for choosing DE are [74]: 1) DE can work in non-linear, non-differentiable, and multi-modular search spaces. 2) It is genuinely robust and quick. 3) Though intended to take care of real-valued problem spaces, it can also be utilized for discrete problem spaces. Owing to the above advantages of DE, found that DE is the best choice for EC-based rule mining.

6.3 Contribution

1. Proposed novel EC-based rule miners with a new encoding scheme for solving classification tasks of credit scoring and detecting financial statement fraud in the banking industry.
2. Further, the proposed rule miners yielded just a couple of rules with very high AUC compared to a baseline Chi-Square Automatic Interaction Detector (CHAID) which yielded a much higher number with reduced AUC. This is the significant achievement of the study.
3. Intervals for the attributes present in the antecedent of the rules yield more information and human comprehensibility compared to other extant rule-based classifiers.

6.4 Related Work

Dash and Liu in [77] proposed a thorough survey of the existing frameworks for feature selection and advocated a feature selection procedure, which includes four tasks namely feature creation, feature evaluation, testing and stopping criteria. The standard evaluation measure, like accuracy, was employed for eliminating inappropriate features. Machine learning techniques have been employed for solving such decision-forming issues. Conventional data mining methods have been employed for building credit scoring frameworks [130]. Investigators have devised several models that include linear discriminate strategy [29, 185, 300], decision tree concept [131], Rough sets theory strategy [44], F-score model [61] and lastly, the genetic programming concept [225].

Moving ahead, lately, the researchers proposed a hybrid data mining strategy in the development of a sound credit scoring framework. Lee and his colleagues [186] incorporated neural network with standard discriminate analysis concept and Chou [63] employed machine learning approaches like Artificial Neural Networks (ANN or NN), Support Vector Machines (SVM) and Decision Trees (DT). Past research studies emphasized increasing the accuracy of the credit scoring framework. Any improvement in the accuracy would result in notable savings of cost. As per past research studies, machine learning practices turned out to be better than traditional practices in handling credit scoring issues, particularly when non-linearity exists in data. Additionally, for conventional statistical methods, a probability distribution needs to be assumed. But, in reality, these assumptions may not be satisfied. Therefore, non-parametric or distribution-free methods are needed. Machine learning methods fill this gap more than adequately. [130].

On the other hand, there exists limited research in the context of financial statement fraud discovery with the help of data mining practices. The most often used methods have been case-based reasoning, decision tree techniques, text mining, neural networks, logistic regression and others. Kirkos et al. [155], state that auditing scams cost over 400 billion dollars in value yearly to the businesses of the United States. Spathis et al. [282] matched multi-criteria decision assistance with statistical methods like discriminant and logit analysis in identifying deceitful financial statements. An innovative financial core for the detecting management scam is devised by making use of a support vector machine on the financial information by Cecchini et al. [49].

Moreover, Huang et al. [132] devised a ground breaking scam discovery mechanism based on Zipf's Law. Additionally, Kirkos et al. [155] employed the ID3 decision tree as well as Bayesian belief network for detecting financial statement frauds. Sohl and Venkatachalam [275] employed back-propagation NN for this task. Cerullo and Cerullo [50, 51] elucidated the nature of financial statement fraud with the usage of NN. Calderon and Cheh [46] scrutinised the effectiveness of NN being a prospective detector of business fraud auditing.

Further, Koskivaara [161] examined the effect of several pre-processing frameworks upon the prediction ability of NN at the time when auditing financial records. Busta and Weinberg [43] made use of six diverse NN designs through several subsets of 34 variables for distinguishing between 'normal' and 'manipulated' financial information. They scrutinised numbers' with the digit distribution in the crucial financial information based on Benford's regulation [43]. This regulation confirmed that the digits of logically occurring numbers are dispersed and predictable on a pattern. Feroz et al. [94] detected that the comparative effectiveness of the NN frameworks was because of their powerful capability of 'learning' from examples.

Fanning and Cogger [93] made use of NN (AutoNet) to distinguish management scams on the vital openly accessible predictors of deceitful financial statements. Moreover, Ramamoorti et al. [248] contrasted the outcome of the multilayer perceptron with the Delphi research. In the research conducted by Zhang et al. [330], the review of papers reported the usage of NN within forecasting during the years 1988–98. Furthermore, Deshmukh and Talluru [81] appointed a 15 rule-based fuzzy reasoning approach for assessing the threat of management frauds. Then, Pacheco et al. [230] devised a hybrid system involving NN as well as a fuzzy system for this purpose.

Further, Magnusson et al. [200] made use of text mining and confirmed that the language of the quarterly report offered a sign of the modification within the corporation's financial position. Further, the variable selection was employed within the rule-based system by means of removing factors, which were either terminated or which are less influential on predictive data. Several investigators suggested rule extraction algorithms built on top of standard meta-heuristic optimization methods like GA [110], PSO [151], ACO [67] and finally, DE [293].

Firstly, Mahfoud and Mani [205] made use of GA to generate rules for predicting the outcome of individual stocks. Then, Shin and Lee [273] generated rules using GA

for forecasting bankruptcy of companies. After that, Parpinelli et al. [239] suggested Ant-Miner that makes use of ACO to extract classification rules. Afterwards, Kim and Han [154] made use of GA for determining the rules for forecasting firms' bankruptcy. Sousa et al. [281] suggested Constricted PSO (CPSO) for rule mining. Afterwards, Liu et al. [196] suggested PSO-trained rule extraction approach, where they put forward fitness function different from that of [281]. Thereafter, Ji et al. [141] enhanced the Ant-miner suggested within [239]. Zhao et al. [335] suggested Fuzzy-PSO, in which the binary PSO develops fuzzy regulations. After that, Holden and Freitas [126] hybridised ACO and PSO to determine the grouping rules. Then, Su et al. [294] adopted DE for rule extraction. Furthermore, Ravisankar et al. [256] made use of Classification and Regression Trees (CART) together with other approaches to examine fraudulent corporations falling within a Chinese Bank dataset. Most recently, Pradeep et al. [245] suggested a new Rule Extraction method by making use of Firefly and threshold accepting optimization algorithm to detect financial statement fraud.

Uthayakumar et al. [307] presented an ACO-driven rule-based classification system for credit scoring. They performed their analysis on both subjective and quantitative datasets. A multi-objective credit scoring model is proposed, namely, multi-objective PSO in [114, 115] and compared with linear discriminant analysis (LDA). The Gustafson-Kessel (GK) calculation is the most important tool for credit scoring. Nevertheless, this calculation exhibits a moderately poor capacity to distinguish a subset of features from a large dataset. Most techniques that use the GK calculation require a predefined number of clusters. In [262], GK-based modified binary particle swarm optimization (MBPSO) technique is developed to increase the classification accuracy. In [109], Evolutionary Algorithms (EA)- based models proposed during 1997 to 2018 for credit card scoring are reviewed. The focus in [109] is based on two main aspects: (i) model type with issues addressed and (ii) assessment procedures. Al-Sahaf et al. in [16], presented a survey on the applications of evolutionary machine learning, for example, classification, regression and clustering on emerging areas including combinatorial optimization, computer vision, deep learning, transfer learning, and ensemble learning.

The search resulted in 3300 articles for DE and 43 for binary DE during 2016 to 2019. These search results of DE include design innovation as well as applications. Therefore, referred to the updated and latest surveys on DE, which include the innovations along with theoretical analysis. In 2009, a state-of-art survey on DE [74] was

presented, and an updated version [73] was presented in 2016. In [227], a theoretical analysis of the DE was presented. In 2019, another survey on DE along with analysis, is presented [90]. There are a few binary optimization algorithms, and most of the optimization algorithms are continuous optimization algorithms which offer good results in continuous spaces. These continuous algorithms can be adapted to solve binary and combinatorial optimization problems. Crawford et al. [69] reviewed articles on the binarization of meta-heuristics, which are originally intended for continuous optimization. Banitalebi et al. [27] proposed a new self-adaptive binary version of DE (SabDE), which employs an adaptive methodology for generating solutions and a chaotic procedure for adjusting parameters. Aslan et al. in [20], developed a discretization method for Jaya [234] by utilizing exclusive-or (XOR). They present a basic, yet a viable strategy where the update rule of Jaya is replaced with the XOR operator. Arora and Ravi in [309], developed ACO-based miner called ADACOM for extracting rules for classification task.

6.5 Differential Evolution (DE) and Adaptive Differential Evolution (ADE)

6.5.1 Differential Evolution

Storn and Price proposed Differential Evolution [293] during the year 1997 for addressing the optimization. DE integrates four significant segments that are initialization, mutation, selection and recombination. At every iteration, a new solution is developed with the combination of available solutions of the current set of solutions that are termed as mutations. The solutions, which at this level are combined with a target solution within a process termed as a crossover. Further, taking into consideration fitness, the solutions are selected. Further, let a, b, c be considered as being the three solutions vectors selected at random from the population ‘NP’ along with the associated equation operatives used for crossover and mutation mentioned below.

If $(\text{rand}(0, 1) < \text{CR} \parallel i = R)$ then $u_i = a_i + F \times (b_i - c_i)$

If $(\text{rand}(0, 1) > \text{CR} \ \& \ i \neq R)$ then $u_i = x_{\text{current}}$

DE/rand/1/bin is adopted. Differential weight, ‘F’ falls in the range of 0 and 2. Crossover Rate ‘CR’, is in the range of 0 and 1.

The main reasons for selecting DE include [74]: Firstly, DE can work on non-differentiable, non-linear as well as multi-modular search spaces. Secondly, it is powerful and fast.

6.5.2 Adaptive Differential Evolution

Adaptive Differential Evolution modifies the DE in two aspects, i.e. 1) Roulette Wheel Selection [25] is adopted on the 12 sets of CR (Crossover Rate) as well as Differential Weight (F) upon rate of success of the chosen selection and 2) Random Localization (RL) [144]. Moving ahead, the modification in the parameters of ADE takes place by way of Roulette Wheel Selection (RWS). Moreover, within [304], RWS is meant for better solutions on the efficiency rate of the choice of individual parameters sets. Additionally, there are twelve sets of parameters. Out of the twelve, six make use of binomial crossover as well as other six make use of exponential crossover. The complete twelve sets of parameters make use of random localization during the mutation process. During the adaptation process, the twelve sets of ‘CR’, as well as ‘F’, are acknowledged through means of 12 different probabilities q_h , $h = 1, 2, \dots, H$ (here H is 12). Such likelihoods are modified through the rate of progress during the last iterations. During the occurrence which $f(y) > f(x_i)$ after that the h^{th} setting is chosen.

Within [304], RL concept is proposed for DE. Instead of selecting the present vector arbitrarily, choose the most effective amongst the three selected vectors. Moreover, this essentially saves about 35 percent of the overall objective function evaluations [304]. The mutant vector ‘u’ is created from three arbitrarily selected vectors a, b, c. Among the three, the most effective vector is selected as ‘a’.

$$u_i = a_i + F \times (b_i - c_i)$$

The new vector is essentially processed from mutant vector ‘u’ along with the original vector by carrying out either the exponential or binomial crossover. The main intent behind selecting the adaptive version of DE is the forceful determination of most effective DE control parameters settings that trim down the tuning time of the control parameters.

Benchmark	LB	UB

Figure 6.1: Rule Encoding for the Antecedent

Dimension 1			Dimension 2			Dimension 3						Dimension D		
1	0.1	0.5	0	0.4	0.7	1	0.2	0.9	...	...	...	0	0.7	0.9

Figure 6.2: Antecedent part of the rule expanded

6.6 Proposed Methodology

6.6.1 Rule Encoding

A new rule encoding scheme is proposed taking cue from Pradeep et al. [245], wherein here each attribute of the dataset is represented by means of 3 bits. As a result, for the overall attributes, i.e. 35, 24 and 14, available in the Financial Statement Fraud, German and Australian credit scoring datasets respectively, and ended up with, i.e. 105, 72, 42 bits respectively. The first bit specifies whether to involve a given attribute within the rule or not. Thus, ‘0’ indicates that the attribute is not present within the rule, and ‘1’ shows otherwise. Moreover, the second and third bit represents the attribute’s lower and upper bounds [84, 101]. D is the number of financial attributes, which is 35, 24, 14 respectively, for the three problems in the case.

An example rule can be, *if the Attribute1 = [0.17, 0.87] and Attribute5 = [0.34, 0.64] then CLASS_LABEL* which has a specific cover over the dataset. In the current case, the rule encoding scheme is depicted in Figures 6.1 and 6.2.

6.6.2 DE and ADE based Rule-based Classification

The process flow of DE and ADE based rule miners is depicted in Figure 6.3. Algorithms 5 and 6 present the steps of implementation of BDE and ABDE rule miners.

Algorithm 5: Differential Evolution (DE) based Rule Miner

```
DE based Rule Miner( $D_{TrainFolds}$ ,  $D_{TestFold}$ )
{
   $Population$ ,  $NP$ ,  $CR$ ,  $F$ 
   $I = 1$ 
   $J = 1$ 
  while  $I \leq MaxIterations_{DE}$  do
    while  $J \leq NP$  do
      Choose solution vectors  $x_1$ ,  $x_2$ ,  $x_3$  randomly
      as mutant vectors from  $X_{Continuous}$ 
      Perform Mutation
      Perform Binomial Crossover
      Perform Binary Discretization of the
      Perform Binary Discretization on each of the first bit
      of the three bits representing each attribute in the dataset of the
      new solution vector  $x_{new}$  using Sigmoid Function
       $\hat{x}$  is the new binary solution vector representing selection of attribute
      If second bit greater than third bit then swap the second and third bits
      representing each attribute in the dataset of the
      new solution vector  $x_{new}$ 
       $\hat{y} = \mathbf{ComputeFitness}\text{-}F(\hat{x}, x_{new})$ 
      if  $\hat{y}$  better than fitness of current solution vector in  $X$  then
        Replace current vector by  $\hat{x}$  in  $X_{Binary}$ 
        Replace current vector by  $x_{new}$  in  $X_{Continuous}$ 
        Also replace the corresponding fitness value with  $\hat{y}$ 
      end
       $J = J + 1$ 
    end
     $I = I + 1$ 
  end
  return ( $X$ ,  $Y$ )
}
Main()
{
  Repeat
  {
    ( $X$ ,  $Y$ ) = DE based Rule Miner( $D_{TrainFolds}$ ,  $D_{TestFold}$ )
    if  $\hat{y}_{best}$  greater than minimum coverage threshold then
      Add the rule encoding to the rule base  $M$ .
      Break and come out
    end
  }
  Apply the rule base on 20 percent test data and select the best rule
}
```

Algorithm 6: Adaptive Differential Evolution (ADE) based Rule Miner

```
ADE based Rule Miner( $D_{TrainFolds}$ ,  $D_{TestFold}$ )
{
  Population,  $NP$ 
   $I = 1$ 
   $J = 1$ 
  while  $I \leq MaxIterations_{ADE}$  do
    while  $J \leq NP$  do
      Perform Roulette Wheel Selection to choose between
      the 12 parameter sets (of CR and F) based
      on the success rate of each set of parameters
      Choose solution vectors  $x_1$ ,  $x_2$ ,  $x_3$  randomly
      as mutant vectors from  $X_{Continuous}$ 
      Select the best among three
      as the current solution vector (Random Localization)
      Perform Mutation
      Perform Binomial or Exponential Crossover
      Perform Binary Discretization on each of the first bit
      of the three bits representing each attribute in the dataset of the
      new solution vector  $x_{new}$  using Sigmoid Function
       $\hat{x}$  is the new binary solution vector representing selection of attribute
      If second bit greater than third bit then swap the second and third bits
      representing each attribute in the dataset of the
      new solution vector  $x_{new}$ 
      Apply Fitness Function (F)
       $\hat{y} = \mathbf{ComputeFitness}\text{-}F(\hat{x}, x_{new})$ 
      if  $\hat{y}$  better than fitness of current solution vector in  $X$  then
        Replace current vector by  $\hat{x}$  in  $X_{Binary}$ 
        Replace current vector by  $x_{new}$  in  $X_{Continuous}$ 
        Also replace the corresponding fitness value with  $\hat{y}$ 
      end
       $J = J + 1$ 
    end
     $I = I + 1$ 
  end
  return ( $X$ ,  $Y$ )
}
Main()
{
  Repeat
  {
    ( $X$ ,  $Y$ ) = ADE based Rule Miner( $D_{TrainFolds}$ ,  $D_{TestFold}$ )
    if  $\hat{y}_{best}$  greater than minimum coverage threshold then
      Add the rule encoding to the rule base  $M$ 
      Break and come out
    end
  }
  Apply the rule base on 20 percent test data and select the best rule
}
```

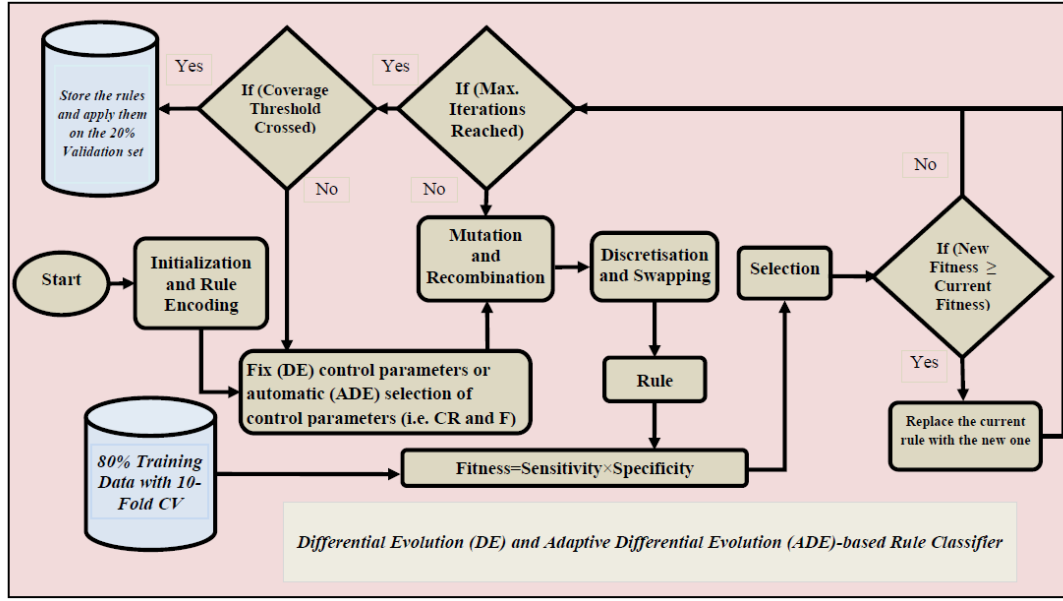


Figure 6.3: Flow Diagram of DE and ADE based Rule Miners

6.6.3 Initialisation

Initialise NP, the number of solution vectors of dimensionality size, $3 \times \text{Dim}$ (wherein Dim refers to the total number attributes). Examine if the solution vectors are within bounds, i.e. between 0 and 1. Compute the values of fitness of the NP solution vectors.

6.6.4 Mutation

6.6.4.1 Mutation for DE

Consider x_1 , x_2 , and x_3 , as the three solution vectors selected randomly from the population of size NP. Carry out the mutation by making use of the equation 1a

$$u'_i = x_{1,i} + F * (x_{2,i} - x_{3,i}) \quad (1a)$$

The equation above is a DE variant, i.e. of DE/rand/1/bin. Differential weight, 'F', lies within the range of 0 and 2.

6.6.4.2 Mutation for ADE

Consider x_1, x_2, x_3 as the three solution vectors selected randomly from the population of size NP. Among the three, the most suitable vector for fitness value is selected as ' x_1 '. Carry out the mutation by making use of the equation below,

$$u'_i = x_{1,i} + F * (x_{2,i} - x_{3,i}) \quad (1b)$$

The equation above is a DE variant, i.e. of DE/randrl, wherein 'randrl' implies the random localisation [144]. Differential weight, 'F', is either 0.5 or 0.8.

6.6.5 Recombination

6.6.5.1 Recombination in DE

Making use of the mutation vector along with the crossover rate parameter C, the recombination is carried out, as mentioned in equation 2:

$$v_i = \begin{cases} u_i & \text{if } (\text{rand}(0, 1) \leq CR \mid i = I_{rand}) \\ x_i^{current} & \text{if } (\text{rand}(0, 1) > CR \text{ and } i \neq I_{rand}) \end{cases} \quad (2)$$

CR is within the range of 0 and 1. Moreover, I_{rand} is a random integer drawn from 1 to Dim and I_{rand} is used to ensure that v and $x^{current}$ are diverse.

6.6.5.2 Recombination in ADE

1. Within the binomial crossover (DE/././bin expresses the exponential crossover), the modified solution values are distributed randomly from 1 to D. The binomial crossover within ABDE basically makes use of three values of CR ($CR = \{0, 0.5, 1\}$) directed towards two values of F (i.e. $F = 0.5$ and $F = 0.8$). As a result, an overall of 6 combinations.
2. For exponential crossover (DE/././exp expresses the exponential crossover), the starting place of crossover is selected randomly from 1 to D, and L successive solution values are obtained from mutant vector v . Probability of substituting the k^{th} solution values from 1, 2, . . . , L with $L \leq D$, reduces exponentially with increasing k.

-
3. The three CR estimations for exponential crossover are principally obtained from the origins of a specific polynomial [305] that intern makes use of three values of the probability, i.e. p_m that is established equidistantly in the $(1/D, 1)$ limits. The exponential crossover within ABDE makes use of three projected CR values together with the two F values (i.e. $F = 0.5$ and $F = 0.8$). As a result, an aggregate of 6 combinations.
 4. The 12 diverse parameter sets offer a reasonable opportunity for regulating exploitation and exploration during different stages of the optimisation procedure [304].

6.6.6 Discretisation and Swapping

DE is developed for real input parameter-based optimisation problems. Therefore, the discretization of the first bit of each attribute to binary values is necessary according to the rule encoding scheme.

$$w_i = \begin{cases} 1 & \text{if}(\text{rand}(k, 1) \leq \text{Sigmoid}(v_{3i-2})) \\ 0 & \text{if}(\text{rand}(k, 1) > \text{Sigmoid}(v_{3i-2})) \end{cases} \quad (3)$$

Where w is a binary vector and k is the bias which lies in $(0,1)$.

The second and third bits are swapped if the second bit is higher than third or otherwise left as it is and taken into v . These binary values (i.e. w) and continuous values (i.e. v) are taken into ‘R’, which represents a rule generated by DE or ADE.

6.6.7 Fitness Evaluation

For DE or ADE based rule miners:

For the rule ‘R’, compute the fitness function (F), i.e. F is to be maximized.

$$F = \text{maximize}\{\text{Sensitivity} \times \text{Specificity}\} \quad (4)$$

In the confusion matrix [276], TP implies the detected true positives class, TN refers to the detected true negatives class, FP the detected false positives class and lastly, FN the detected false negatives, class.

Sensitivity

Sensitivity [276] (termed as the true positive rate (TPR) or the recall), calculates the rate of positive class samples, which are precisely detected.

Specificity

Specificity [276] (also termed as the true negative rate (TNR)), calculates the rate of negative class samples, which are correctly detected.

Area Under Curve

The area under the ROC curve (AUC) [276] is the average of specificity and sensitivity for two-class problems and is highly useful for unbalanced datasets.

6.6.8 Selection

If $\text{Fitness}(\mathbf{R}) \leq \text{Fitness}(\mathbf{y}^{\text{current}})$ replace current vectors of $\mathbf{y}$, $\mathbf{x}_{\text{binary}}$, $\mathbf{x}_{\text{continuous}}$ in the population (i.e. both binary and continuous) with solution vectors $\mathbf{w}$ and $\mathbf{v}$ (i.e. both binary and continuous). Else do not replace the current vectors.

DE and ADE based rule classifiers are proposed as general-purpose rule miners and employ them for financial statement fraud detection and credit scoring. Firstly, the datasets are grouped into Training, Validation and Testing datasets. The ADE and DE are then employed for getting the rules. This procedure is continued until the entire classes covered by the rule are substantially higher than the pre-specified threshold percent of the coverage count.

6.7 Dataset Description and Experimental Setup

Two benchmark credit scoring datasets, namely German Credit Scoring dataset [7] and Australian Credit Scoring dataset [6] are taken from the UCI Machine Learning Repository. Another dataset, is namely Financial Statement Fraud dataset. Such datasets hold a broad inference for the banking sector.

The German Credit Scoring dataset involves around 700 customer samples of credit worthiness along with around 300 customer samples of non-credit worthiness. An overall of 1000 samples with 24 attributes and a class attribute are there within the German Credit Scoring dataset. Further, the German Credit Scoring dataset's numeric version is taken into consideration for investigation. Australian Credit Scoring dataset includes around 383 'bad' credit scoring candidates' samples along with 307 'good' credit scoring candidates' samples. An overall of 690 samples, with 14 attributes and a class attribute exist within the Credit Scoring dataset of Australia.

The Financial Statement Fraud dataset involves around 35 financial factors, as stated within [256]. Such financial factors belong to 101 fraudulent as well as 101 non-fraudulent corporations. As a result, the dataset includes an overall of 202 samples. During the pre-processing phase, it is highly observed that a few of the financial attribute values within the dataset have quite larger values. Therefore, a natural logarithmic change was carried out, followed through normalization.

Taking into cue from Krishna and Ravi [172], further examined the attributes selected from the evolutionary computing-based wrapper approach. The 10 features from Australian credit scoring dataset, 18 features from German credit scoring dataset and 24 features from financial statement fraud which are subscripted ‘FS’ (i.e. feature selected) are selected based on the number of times the features are repeated in the results presented in [172] of the 10 fold cross-validation.

Features names from Australian credit scoring datasets are “A1, A2, A3_{FS}, A4_{FS}, A5, A6, A7_{FS}, A8_{FS}, A9_{FS}, A10_{FS}, A11_{FS}, A12_{FS}, A13_{FS}, A14_{FS}”.

Features names from German credit scoring datasets are “BalanceCheque_{FS}, Loan_Num Month_{FS}, CreditHistory_{FS}, CreditAmt_{FS}, SavingsBalance_{FS}, Mths-employ, PersonStatusSex, Present Residence_{FS}, Property_{FS}, AgeInYears, Other-InstPlans_{FS}, Num Credits ThisBank_{FS}, NumPplLiabl-Maint_{FS}, Telephone_{FS}, ForeignWorker, Purpose_Car New_{FS}, Purpose_Car Old_{FS}, other debtor_ noneVsGuar-_{FS}, otherdebt.coapplVs Guar_{FS}, house_rent Vs Free, house_ owns Vs Free_{FS}, job-unemployedVsMgt_{FS}, jobs_unskilled Vs Mgt, job_ skilledVs Mgt_{FS}”.

Features names from financial statement fraud dataset are “Debt_{FS}, Total assets, Gross profit, Net profit, Primary business income_{FS}, Cash and deposits, Accounts receivable, Inventory/Primary business income, Inventory/Total assets_{FS}, Gross profit/Total assets_{FS}, Net profit/Total assets_{FS}, Current assets/Total assets_{FS}, Net profit/Primary business income_{FS}, Accounts receivable/Primary business income_{FS}, Primary business income/Total assets_{FS}, Current assets/Current liabilities_{FS}, Primary business income/Fixed assets_{FS}, Cash /Total assets, Inventory/Current liabilities_{FS}, Total debt/ Total equity_{FS}, Long term debt/Total assets, Net profit/Gross profit_{FS}, Total debt/Total assets_{FS}, Total assets/Capital and reserves, Long term debt/Total capital and reserves_{FS}, Fixed assets/Total assets_{FS}, Deposits and cash/Current assets_{FS}, Capitals and reserves/Total debt_{FS}, Accounts receivable/Total assets, Gross profit/business profit_{FS}, Undistributed profit/Net profit_{FS}, Primary business profit/ Primary

business profit of last year_{FS}, Primary business income/Last year's primary business income_{FS}, Account receivable /Accounts receivable of last year_{FS}, Total assets/Total assets of last year".

6.8 Results and Discussion

The present work is run on Intel Core i7 processors with 32 GB RAM and implemented in R. Min/Max normalization is performed on all the datasets. Both DE and ADE based rule miners are run for 4000, 6000 and 600 iterations with a population size of 20 for Australian Credit Scoring, German Credit Scoring and Financial Statement Fraud datasets respectively. Additionally, three datasets are split into 80 percent, and 20 percent sets and the latter one is selected as being the validation set. After that, 10-fold cross-validation is carried out on the 80 percent dataset. Due to the evolutionary aspect of the ADE and DE Algorithms, the outcomes are different to some extent, when the random seed is modified. For that reason, every miner is run around 25 times for each fold, and after that, the average outcomes are calculated as well as tabulated. In the training phase, 80 percent of the dataset is utilized, and the best rule is selected in terms of the objective function after all the iterations are completed by the DE or ADE. This process is performed for 25 times for each of the ten folds data. Therefore, there are 25×10 , i.e. 250 runs for every individual algorithm. From the 250 runs, i.e. 250 different best rules from each run are collected and labelled based on the maximum cover. The 25 rules of each of the ten folds are applied on 20 percent validation set, and the best rule from the 25 rules is picked based on the maximum coverage of the rule or best objective function value on the validation set. These results are presented in Tables 6.1, 6.2 and 6.3. For DE, various parameter combination for CR and F were tried out. The parameter combination of $CR = 0.15$ and $F = 0.1$ gave the best results with all three datasets for DE. But for ADE, the parameters are tuned in adaptive manner.

Taking into consideration Krishna and Ravi [172], further examined the attributes selected from the evolutionary computing-based wrapper approach. As listed above in section 6.7, considered 65% to 75% features based on the two or more repetitions from the best ADE-SVM results of 10-fold cross-validation. The 10 features from Australian Credit Scoring dataset, 18 features from German Credit Scoring dataset and 24 features

from Financial Statement Fraud dataset, i.e. 71.4% of features, 75% of features and 68.5% percent of features respectively are considered for the study. These results are presented in Tables 6.4, 6.5 and 6.6. To show that the features selected in the previous study are useful in the current study, extended our earlier study in this work. The rule-based classifier also entails feature selection in an implicit way because of the combinatorial optimization framework of the current study.

Table 6.1: Results of DE and ADE Rule Miners on 20 % Validation Set of Australian Credit Scoring Dataset

Results over 10 Folds	Australian Credit Scoring Dataset						
	DE			ADE			
	Objective Function Values	Sensitivity	Specificity	Objective Function Values	Sensitivity	Specificity	
Maximum	0.8447	0.9122	0.9259	0.8559	0.9473	0.9382	
Minimum	0.8174	0.8947	0.9012	0.7719	0.8771	0.8148	
Median	0.8221	0.9122	0.9074	0.8284	0.9035	0.9197	
Mean	0.8273	0.9105	0.9086	0.8217	0.9035	0.9098	
Std. Dev.	0.0084	0.0055	0.0086	0.0265	0.0206	0.0349	

Table 6.2: Results of DE and ADE Rule Miners on 20 % Validation Set of German Credit Scoring Dataset

Results over 10 Folds	German Credit Scoring Dataset						
	DE			ADE			
	Objective Function Values	Sensitivity	Specificity	Objective Function Values	Sensitivity	Specificity	
Maximum	0.9705	0.9852	1.0	0.9698	0.9852	0.9843	
Minimum	0.9698	0.9705	0.9843	0.8529	0.8970	0.9062	
Median	0.9705	0.9705	1.0	0.9326	0.9669	0.9609	
Mean	0.9705	0.9720	0.9984	0.9267	0.9625	0.9625	
Std. Dev.	0.0002	0.0046	0.0049	0.0430	0.0265	0.0246	

Table 6.3: Results of DE and ADE Rule Miners on 20 % Validation Set of Financial Statement Fraud Dataset

Results over 10 Folds	Financial Statement Fraud Dataset					
	DE			ADE		
	Objective Function Values	Sensitivity	Specificity	Objective Function Values	Sensitivity	Specificity
Maximum	0.81	0.9	1.0	0.8075	0.85	0.95
Minimum	0.525	0.75	0.7	0.595	0.7	0.8
Median	0.8037	0.85	0.925	0.68	0.8	0.9
Mean	0.835	0.835	0.895	0.6912	0.78	0.885
Std. Dev.	0.1015	0.0095	0.0411	0.0800	0.0632	0.0529

Table 6.4: Results of DE and ADE Rule Miners on 20 % Validation Set of Australian Credit Scoring Dataset with Feature Subsets from [172]

Australian Credit Scoring Dataset						
Results over 10 Folds	DE			ADE		
	Objective Function Values	Sensitivity	Specificity	Objective Function Values	Sensitivity	Specificity
Maximum	0.7792	0.8771	0.8888	0.7797	0.8771	0.9843
Minimum	0.5596	0.6666	0.8395	0.5596	0.6666	0.9687
Median	0.7792	0.8771	0.8888	0.7797	0.8771	0.9765
Mean	0.7566	0.8561	0.8827	0.7577	0.8561	0.9765
Std. Dev.	0.0692	0.0665	0.0156	0.0695	0.0665	0.0082

Table 6.5: Results of DE and ADE Rule Miners on 20 % Validation Set of German Credit Scoring Dataset with Feature Subsets from [172]

German Credit Scoring Dataset						
Results over 10 Folds	DE			ADE		
	Objective Function Values	Sensitivity	Specificity	Objective Function Values	Sensitivity	Specificity
Maximum	0.9626	0.9779	0.9843	0.9626	0.9779	0.9382
Minimum	0.9626	0.9779	0.9843	0.8761	0.9044	0.8148
Median	0.9626	0.9779	0.9843	0.9478	0.9705	0.9197
Mean	0.9626	0.9779	0.9843	0.9308	0.9529	0.9098
Std. Dev.	0.0	0.0	0.0	0.0395	0.0339	0.0349

Table 6.6: Results of DE and ADE Rule Miners on 20 % Validation Set of Financial Statement Fraud Dataset with Feature Subsets from [172]

Results over 10 Folds	Financial Statement Fraud Dataset					
	DE			ADE		
	Objective Function Values	Sensitivity	Specificity	Objective Function Values	Sensitivity	Specificity
Maximum	0.85	0.9	1.0	0.81	0.9	0.95
Minimum	0.7125	0.75	0.85	0.3575	0.55	0.6
Median	0.7862	0.85	0.95	0.72	0.8	0.9
Mean	0.7805	0.84	0.93	0.6687	0.78	0.85
Std. Dev.	0.0435	0.04594	0.0421	0.1515	0.1632	0.1290

The t-Test is performed on objective function values on the 20 percent of the validation set obtained from the rules obtained from the training phase of 10 fold cross-validation of 80 percent data. DE performs better than ADE. Therefore, a pairwise t-test is performed at 1% significance and 18 degrees of freedom with DE versus ADE. In the event that the p-value is under 0.01, at this point the null hypothesis is rejected, thereby conveying that the pair of optimizers are not equal. Else, if the p-value is more than 0.01, then the null hypothesis isn't rejected, and the pair of optimizers are equal. t-Test results are presented in Tables 6.7 and 6.8, with the significance value of 'T', i.e. as true, when the p-value is underneath 0.01 and 'F', i.e. as false, when the p-value is higher than 0.01. Out of the six p-values from Tables 6.7 and 6.8, it is evident that in four out of six p-values are indicating that both the optimizers are statistically equal (i.e. null hypothesis is not rejected), and in only two cases DE is found to be statistically not equal to ADE (i.e. null hypothesis is rejected). However, when it comes to choosing between the two optimisers, ADE is preferred as there no additional control parameter tuning is needed as ADE is adaptive.

Table 6.7: t-Test results of the DE vs ADE

Compared Methods	t-Values	p-Values	Sig. T/F
DE vs ADE (Australian Credit Scoring)	0.63592	0.53284	F
DE vs ADE (German Credit Scoring)	3.21885	0.00476	T
DE vs ADE (Financial Statement Fraud)	3.51591	0.00247	T

Table 6.8: t-Test results of the DE vs ADE with feature subsets from [172]

Compared Methods	t-Values	p-Values	Sig. T/F
DE vs ADE (Australian Credit Scoring)	0.03513	0.97236	F
DE vs ADE (German Credit Scoring)	2.55070	0.02007	F
DE vs ADE (Financial Statement Fraud)	2.24055	0.03790	F

The rules obtained by DE and ADE rule miners are compared with CHAID decision tree [147]. All rules obtained by DE or ADE rule miners are with two antecedents and best AUC than CHAID. Also, the number of rules obtained by DE or ADE rule miners are less than CHAID. CHAID produced 15, 31 and 11 rules for Australian Credit

Scoring, German Credit Scoring and Financial Statement Fraud datasets. These rules and results are summarised in Tables 6.9, 6.10 and 6.11.

Table 6.9: Rules obtained by DE, ADE Rule Miners and CHAID on Australian Credit Scoring Dataset

Model	Rules	AUC
DE Rule Miner	IF A2 $\in$ [0.0003, 0.9947] and A12 $\in$ [0.3236, 0.6942] then Good IF A2 $\in$ [0.0102, 0.9654] and A3 $\in$ [0.0002, 0.9704] then Bad	0.9191
ADE Rule Miner	IF A2 $\in$ [0.0014, 0.9702] and A12 $\in$ [0.0562, 0.5182] then Good IF A2 $\in$ [0.0007, 0.9844] and A3 $\in$ [0.0012, 0.9054] then Bad	0.9165
DE Rule Miner (with reduced features from [172])	IF A3 $\in$ [0.00037, 0.9646] and A12 $\in$ [0.1043, 0.6025] then Good IF A3 $\in$ [0.00123, 0.9626] and A13 $\in$ [0.0048, 0.92657] then Bad	0.8583
ADE Rule Miner (with reduced features from [172])	IF A3 $\in$ [0.0002, 0.9866] and A12 $\in$ [0.07488, 0.79494] then Good IF A3 $\in$ [0.00102, 0.9640] and A13 $\in$ [0.0078, 0.83366] then Bad	0.8583

Table 6.9: Rules obtained by DE, ADE Rule Miners and CHAID on Australian Credit Scoring Dataset (Continued)

CHAID [147]	If A8 ≤ 0 and A12 > 0.5 then Good If A8 ≤ 0 and A12 ≤ 0 and A5 > 0.615385 then Good If A8 ≤ 0 and A12 $\in (0, 0.5]$ and A3 ≤ 0.321429 then Bad If A8 ≤ 0 and A12 $\in (0, 0.5]$ and A3 > 0.321429 then Bad If A8 > 0 and A9 ≤ 0 and A5 ≤ 0.538462 then Bad If A8 > 0 and A9 > 0 and A14 $\in (3e-05, 0.00284]$ then Good If A8 > 0 and A9 > 0 and A14 > 0.00284 then Good If A8 ≤ 0 and A12 ≤ 0 and A5 ≤ 0.615385 and A7 ≤ 0.0789474 then Bad If A8 ≤ 0 and A12 ≤ 0 and A5 ≤ 0.615385 and A7 > 0.0789474 then Bad If A8 > 0 and A9 ≤ 0 and A5 > 0.538462 and A13 ≤ 0.175 then Good If A8 > 0 and A9 > 0 and A14 $\leq 3e-05$ and A13 ≤ 0.078 then Good If A8 > 0 and A9 > 0 and A14 $\leq 3e-05$ and A13 $\in (0.078, 0.106]$ then Good If A8 > 0 and A9 > 0 and A14 $\leq 3e-05$ and A13 > 0.106 then Good If A8 > 0 and A9 ≤ 0 and A5 > 0.538462 and A13 > 0.175 and A14 ≤ 0 then Bad If A8 > 0 and A9 ≤ 0 and A5 > 0.538462 and A13 > 0.175 and A14 > 0 then Good	0.8623

Table 6.10: Rules obtained by DE, ADE Rule Miners and CHAID on German Credit Scoring Dataset

Model	Rules	AUC
DE Rule Miner	IF Loan_NumMonth $\in [0.0144, 0.9625]$ and AgeInYears $\in [0.0113, 0.8977]$ then Good IF CreditAmt $\in [0.002, 0.9720]$ and AgeInYears $\in [0.0117, 0.9885]$ then Bad	0.9926

Table 6.10: Rules obtained by DE, ADE Rule Miners and CHAID on German Credit Scoring Dataset (Continued)

ADE Miner	Rule	IF Loan_NumMonth $\in$ [0.0045, 0.9287] and AgeInYears $\in$ [0.0019, 0.9127] then Good IF CreditAmt $\in$ [0.0063, 0.9037] and AgeInYears $\in$ [0.01057, 0.8832] then Bad	0.9848
DE Miner	Rule (with reduced features from [172])	IF Loan_NumMonth $\in$ [0.0003, 0.9237] and CreditAmt $\in$ [0.0005, 0.9289] then Good IF Loan_NumMonth $\in$ [0.0058, 0.9929] and CreditAmt $\in$ [0.0132, 0.9193] then Bad	0.9811
ADE Miner	Rule (with reduced features from [172])	IF Loan_NumMonth $\in$ [0.0127, 0.7940] and CreditAmt $\in$ [0.0129, 0.8474] then Good IF Loan_NumMonth $\in$ [0.00187, 0.9682] and CreditAmt $\in$ [0.0026, 0.9423] then Bad	0.9733

Table 6.10: Rules obtained by DE, ADE Rule Miners and CHAID on German Credit Scoring Dataset (Continued)

CHAID [147]	If BalanceCheque ≤ 0 and Loan_NumMonth > 0.470588 then Bad If BalanceCheque ≤ 0 and Loan_NumMonth ≤ 0.25 and CreditHistory ≤ 0.25 then Bad If BalanceCheque ≤ 0 and Loan_NumMonth ≤ 0.25 and CreditHistory $\in (0.25, 0.75]$ then Good If BalanceCheque ≤ 0 and Loan_NumMonth $\in (0.25, 0.470588]$ and Purpose_CarOld > 0 then Good If BalanceCheque $\in (0, 0.666667]$ and Loan_NumMonth ≤ 0.294118 and CreditHistory ≤ 0.25 then Bad If BalanceCheque $\in (0, 0.666667]$ and Loan_NumMonth > 0.294118 and SavingsBalance > 0.75 then Good If BalanceCheque > 0.666667 and OtherInstPlans ≤ 0.5 and Purpose_CarNew > 0 then Bad If BalanceCheque > 0.666667 and OtherInstPlans > 0.5 and CreditHistory > 0.75 then Good If BalanceCheque ≤ 0 and Loan_NumMonth ≤ 0.25 and CreditHistory > 0.75 and house_rentVsFree ≤ 0 then Good If BalanceCheque ≤ 0 and Loan_NumMonth ≤ 0.25 and CreditHistory > 0.75 and house_rentVsFree > 0 then Good	0.73

Table 6.10: Rules obtained by DE, ADE Rule Miners and CHAID on German Credit Scoring Dataset (Continued)

	If BalanceCheque $\in (0, 0.666667]$ and Loan_NumMonth ≤ 0.294118 and CreditHistory > 0.25 and NumPplLiablMaint > 0 then Good If BalanceCheque $\in (0, 0.666667]$ and Loan_NumMonth > 0.294118 and SavingsBalance ≤ 0.75 and job_skilledVsMgt ≤ 0 then Bad If BalanceCheque > 0.666667 and OtherInstPlans ≤ 0.5 and Purpose_CarNew ≤ 0 and Mths_employ ≤ 0.5 then Good If BalanceCheque ≤ 0 and Loan_NumMonth $\in (0.25, 0.470588]$ and Purpose_CarOld ≤ 0 and Telephone ≤ 0 and CreditAmt ≤ 0.0659341 then Bad If BalanceCheque ≤ 0 and Loan_NumMonth $\in (0.25, 0.470588]$ and Purpose_CarOld ≤ 0 and Telephone ≤ 0 and CreditAmt > 0.0659341 then Bad If BalanceCheque ≤ 0 and Loan_NumMonth $\in (0.25, 0.470588]$ and Purpose_CarOld ≤ 0 and Telephone > 0 and Mths_employ ≤ 0.25 then Bad If BalanceCheque ≤ 0 and Loan_NumMonth $\in (0.25, 0.470588]$ and Purpose_CarOld ≤ 0 and Telephone > 0 and Mths_employ > 0.25 then Good If BalanceCheque $\in (0, 0.666667]$ and Loan_NumMonth ≤ 0.294118 and CreditHistory > 0.25 and NumPplLiablMaint ≤ 0 and PersonStatusSex ≤ 0.333333 then Good	

Table 6.10: Rules obtained by DE, ADE Rule Miners and CHAID on German Credit Scoring Dataset (Continued)

	If BalanceCheque $\in (0, 0.666667]$ and Loan_NumMonth ≤ 0.294118 and CreditHistory > 0.25 and NumPplLiablMaint ≤ 0 and PersonStatusSex > 0.333333 then Good If BalanceCheque $\in (0, 0.666667]$ and Loan_NumMonth > 0.294118 and SavingsBalance ≤ 0.75 and job_skilledVsMgt > 0 and AgeInYears ≤ 0.142857 then Bad If BalanceCheque $\in (0, 0.666667]$ and Loan_NumMonth > 0.294118 and SavingsBalance ≤ 0.75 and job_skilledVsMgt > 0 and AgeInYears $\in (0.142857, 0.267857]$ then Good If BalanceCheque $\in (0, 0.666667]$ and Loan_NumMonth > 0.294118 and SavingsBalance ≤ 0.75 and job_skilledVsMgt > 0 and AgeInYears > 0.267857 then Bad If BalanceCheque > 0.666667 and OtherInstPlans ≤ 0.5 and Purpose_CarNew ≤ 0 and Mths_employ > 0.5 and CreditAmt ≤ 0.032967 then Good If BalanceCheque > 0.666667 and OtherInstPlans ≤ 0.5 and Purpose_CarNew ≤ 0 and Mths_employ > 0.5 and CreditAmt > 0.032967 then Good If BalanceCheque > 0.666667 and OtherInstPlans > 0.5 and CreditHistory ≤ 0.75 and NumCreditsThisBank ≤ 0 and Loan_NumMonth ≤ 0.161765 then Good If BalanceCheque > 0.666667 and OtherInstPlans > 0.5 and CreditHistory ≤ 0.75 and NumCreditsThisBank ≤ 0 and Loan_NumMonth $\in (0.161765, 0.205882]$ then Good	

Table 6.10: Rules obtained by DE, ADE Rule Miners and CHAID on German Credit Scoring Dataset (Continued)

	If BalanceCheque > 0.666667 and OtherInstPlans > 0.5 and CreditHistory <= 0.75 and NumCreditsThisBank <= 0 and Loan_NumMonth $\in$ (0.205882, 0.294118] then Good If BalanceCheque > 0.666667 and OtherInstPlans > 0.5 and CreditHistory <= 0.75 and NumCreditsThisBank <= 0 and Loan_NumMonth $\in$ (0.294118, 0.338235] then Bad If BalanceCheque > 0.666667 and OtherInstPlans > 0.5 and CreditHistory <= 0.75 and NumCreditsThisBank <= 0 and Loan_NumMonth > 0.338235 then Good If BalanceCheque > 0.666667 and OtherInstPlans > 0.5 and CreditHistory <= 0.75 and NumCreditsThisBank > 0 and house_ownsVsFree <= 0 then Bad If BalanceCheque > 0.666667 and OtherInstPlans > 0.5 and CreditHistory <= 0.75 and NumCreditsThisBank > 0 and house_ownsVsFree > 0 then Good	

Table 6.11: Rules obtained by DE, ADE Rule Miners and CHAID on Financial Statement Fraud Dataset

Model		Rules	AUC
DE	Rule Miner	IF Net profit/Primary business income $\in$ [0.29, 0.99] and Total assets/Capital and reserves $\in$ [0.02, 0.885] then Fraud IF Total debt/Total equity $\in$ [0.14, 0.99] and Undistributed profit/Net profit $\in$ [0.11, 0.719] then Non-Fraud	0.875

Table 6.11: Rules obtained by DE, ADE Rule Miners and CHAID on Financial Statement Fraud Dataset (Continued)

ADE Miner	Rule	IF Primary business income/Fixed assets $\in [0.0002, 0.8341]$ and Undistributed profit/Net profit $\in [0.1099, 0.6639]$ then Fraud IF Gross profit/Primary business profit $\in [0.0236, 0.6832]$ and Undistributed profit/Net profit $\in [0.1262, 0.6946]$ then Non-Fraud	0.875
DE Miner (with reduced features from [172])	Rule	IF Gross profit/Primary business profit $\in [0.0236, 0.6832]$ and Undistributed profit/Net profit $\in [0.1262, 0.6946]$ then Fraud IF Current assets/Total assets $\in [0.0273, 0.7137]$ and Fixed assets/Total assets $\in [0.1577, 0.9508]$ then Non-Fraud	0.925
ADE Miner (with reduced features from [172])	Rule	IF Fixed assets/Total assets $\in [0.0397, 0.9970]$ and Gross profit/Primary business profit $\in [0.01435, 0.3439]$ then Fraud IF Net profit/Primary business income $\in [0.0036, 0.9975]$ and Gross profit/Primary business profit $\in [0.0324, 0.5114]$ then Non-Fraud	0.925

Table 6.11: Rules obtained by DE, ADE Rule Miners and CHAID on Financial Statement Fraud Dataset (Continued)

CHAID [147]	If Net profit ≤ 0.972292 and Account receivable /Accounts receivable of last year ≤ 0.479157 then Fraud If Net profit ≤ 0.972292 and Account receivable /Accounts receivable of last year > 0.479157 then Fraud If Net profit $\in (0.972292, 0.985193]$ and Primary business income/Total assets $\in (0.229062, 0.523098]$ then Non-Fraud If Net profit $\in (0.972292, 0.985193]$ and Primary business income/Total assets > 0.523098 then Fraud If Net profit > 0.985193 and Primary business income/Last year's primary business income ≤ 0.0258066 then Fraud If Net profit > 0.985193 and Primary business income/Last year's primary business income $\in (0.0258066, 0.0524028]$ then Non-Fraud If Net profit $\in (0.972292, 0.985193]$ and Primary business income/Total assets ≤ 0.229062 and Gross profit/Total assets > 0.938435 then Fraud If Net profit > 0.985193 and Primary business income/Last year's primary business income > 0.0524028 and Net profit/Total assets ≤ 0.969912 then Non-Fraud If Net profit > 0.985193 and Primary business income/Last year's primary business income > 0.0524028 and Net profit/Total assets > 0.969912 then Non-Fraud	

Table 6.11: Rules obtained by DE, ADE Rule Miners and CHAID on Financial Statement Fraud Dataset (Continued)

	If Net profit $\in$ (0.972292, 0.985193] and Primary business income/Total assets $\leq$ 0.229062 and Gross profit/Total assets $\leq$ 0.938435 and Current assets/Current liabilities $\leq$ 0.284323 then Fraud If Net profit $\in$ (0.972292, 0.985193] and Primary business income/Total assets $\leq$ 0.229062 and Gross profit/Total assets $\leq$ 0.938435 and Current assets/Current liabilities $>$ 0.284323 then Non-Fraud	0.725

6.9 Conclusions and Future Directions

Novel DE and an ADE based rule miners are proposed separately. The effectiveness of the proposed classifiers are demonstrated on detecting financial statement fraud and credit scoring. These proposed methods produce rules as opposed to the black box techniques in the machine learning, i.e. neural networks, SVM. Also, they have produced much higher AUC in contrast to the standard decision trees with respect to having lesser rule length. Additionally, ADE and DE miners turned out to be statistically equal (i.e. null hypothesis is not rejected) in four out of six comparisons, as demonstrated through a t-test. However, ADE miner is reasonably less complex as compared to the DE miner in terms of fine-tuning the parameters. Also, with the factors discussed in the results section makes ADE miner a superior as well as stable a choice. Further, the proposed rule miners yielded a couple of rules with very high AUC compared to a baseline CHAID which yielded a much higher number with reduced AUC. Intervals for the attributes present in the antecedent of the rules yield more information and human comprehensibility compared to other extant rule-based classifiers. In future, will experiment with better objective functions and employ even better evolutionary computing-based optimization techniques.

Chapter 7

High Utility Itemset and Association Rule Mining using Binary Differential Evolution: An application to Customer Segmentation

7.1 Introduction

Frequent itemset mining (FIM) finds items which go together based on some measures of interest which are mostly frequency-based, i.e., for example, support, confidence, etc. Sometimes these frequent itemsets may not be of value, as they are not adding to the return of investment of the company or firm. To address this issue, a new branch has emerged in recent times, i.e. High Utility Itemset Mining (HUIM) [52]. In HUIM, a new methodology of considering the utility has been introduced, instead of using a frequentist-based approach.

One important problem with the mining itemsets is that search space to be explored is exponential in number. When the items to be searched are large in number, searching for the itemsets becomes even more harder and this turns out to be an NP-hard problem. Suppose there are ‘M’ items in the transactional database then the search space becomes $2^M - 1$. To address this issue, most of the FIM algorithms use

the property of downward closure. The downward closure property, which is a process of restricting the search space and then searching in the restricted search space. The above property tends to reduce the search space of the frequent itemset mining methods. But, in the case of high utility mining, this property of downward closure cannot be applied as for the reason described in the example of section 7.5.2.

There has been a recent development of using Evolutionary Computing (EC) based HUIM. The use of EC-based HUIM algorithms can address the issue of the combinatorial explosion of the search space when the items become large in the transactional database. Genetic Algorithm (GA) [110], Binary Particle Swarm Optimization (BPSO) [151] are utilized to solve the problem of HUIM [145, 191]. But none applied, state-of-the-art, Binary Differential Algorithm (BDE) or an ABDE based EC techniques for solving the HUIM problem. Also, the optimum number of itemsets using EC techniques was not very high. On the other hand, differential evolution (DE) has long been successfully applied to solving the problems of high dimensions. Further DE is robust and quick compared to GA [73, 74]. Adaptive Differential Evolution (ADE) [304] is a variant of DE where the tuning of DE parameters, i.e. ‘CR’ and ‘F’ is not needed.

EC techniques can also be applied for solving High Utility Association Rule Mining (HUARM) problems as there is combinatorial explosion of the search space. HUARM using EC techniques has not been reported. None utilized, state of the art, Binary Differential Algorithm (BDE) or an Adaptive BDE based EC techniques for solving HUARM problems. Therefore, this is the first work, where HUARM is solved using two EC techniques, i.e. BDE and ABDE and compared with one another. In solving the HUARM task, a novel objective function is devised. Also, instead of utilizing the downward closure [14, 145], we employed the random discretization process by employing a sigmoid function.

According to Kumar and Reinartz [176]: “Customer Relationship Management (CRM) is the practice of analyzing and utilizing marketing databases and leveraging communication technologies to determine corporate practices and methods that will maximize the lifetime value of each individual customer to the firm”. A definitive objective is to optimize the present and future value of customers for the firm. CRM employs analytical techniques for recognizing new customers, sustaining associations with them, holding them by fulfilling their needs and lastly ensuring that productive and loyal customers don’t attrite to rival, competing firms.

Customer monetary value is in terms of the money streams related with the customer by his/her entire association with the firm. This can in one way be evaluated by the recency, frequency and monetary (RFM) analysis of his/her relationship with the firm [112]. By finding the customers with high monetary value, customer campaigning can be performed by presenting the products to the customers by the way of advertisements, discounts and regular mails [163].

We applied the HUIM and HUARM to, among other benchmark datasets, an *OnlineRetail* dataset available on UCI repository [56] with a real-world application of Customer Segmentation based on value, i.e. utility.

7.2 Motivation

The motivation of the present work is as follows:

1. The earlier studies could not show conclusively the power and versatility of BDE and ABDE in terms of the maximum number of itemsets mined when compared with the two-phase HUIM. Therefore, we thought of exploring in this direction.
2. We observed that BDE and ABDE-based HUIM and HUARM algorithms have never been applied to one of the important tasks of CRM viz. Customer Segmentation based on monetary value of the customer, i.e. utility of the customer.
3. From the review of related work, we observed that EC techniques were never employed for mining high utility association rule mining. Therefore, we propose a novel EC based HUARM. Also, there are no works reporting top high utility association rules using either with EC-based techniques or heuristic-based techniques.
4. The reasons for choosing DE are [73, 74]: 1) It can work on non-linear, non-differentiable, and multi-modular search spaces. 2) It is genuinely robust and quick. 3) Though intended to take care of real-valued problem spaces, it can also be utilized for discrete problem spaces. In view of the above advantages, we found that DE is the best choice for the EC-based HUIM or HUARM.

7.3 Contribution

The contributions of the present work are as follows:

1. We developed powerful BDE and ABDE based HUIM and HUARM algorithms without taking the help of the downward closure property.
2. The proposed HUIM and HUARM algorithms fared well in terms of execution time, memory and convergence.
3. Explored a possible application of the HUIM and HUARM for the analytical CRM, i.e. customer segmentation based on monetary value of the customers.
4. Taking cue from Sarath and Ravi [263], novel objective function for the EC-based HUARM is developed.

7.4 Related Work

Evolutionary algorithms [24] are population-based stochastic search strategies, where a population (i.e. solution vectors) are generated iteratively utilizing heuristics until the point where optimum is accomplished. These methods contrast in how the variables are encoded and the heuristics they utilize when updating the variables. This section reviews works reported in three areas: (i) EC applied to data mining (ii) HUIM and (iii) EC applied to HUIM and HUARM.

7.4.1 Evolutionary Computing applied to Data Mining

Various DM techniques require optimization at different stages, for example, in feature selection or in tuning a classifier/regression model or in deciding the ideal parameter that would bring about precise outcomes for the technique [231]. Also, evolutionary techniques can be used independently for solving different data mining tasks as reported in [168]. Given this inherent flexibility of EC techniques, one can conclude that EC techniques alone can be employed and without depending on traditional techniques to solve different data mining tasks which could involve non-linear, multimodal objective functions and non-convex search spaces.

7.4.2 High Utility based Itemset Mining

HUIM is a basic business issue and a developing point in late decades, as an expansion of frequent itemset mining (FIM) [323, 324]. However more factors, for example, profit and quantity are considered in [52]. Many HUIM algorithms were created to mine the HUIs. At first, a two-stage (TWU) HUIM [195] was developed and structured with the transaction-weighted downward closure (TWDC) property to prune the unpromising HUIs beforehand, to find the HUIs. Also, a high-utility-pattern (UP)-tree [189] was developed for mining HUIs. In ref. [178], a projection-based HUIM with the index-based component was developed to mine HUIs. Fournier-Viger et al. [100], introduced a HUIM, in particular, an FHM, to rapidly mine HUIs depending on the assembled Estimated Utility Co-occurrence Structure (EUCS) and the list-based methodology. Other related HUIMs [339, 340] are additionally developed as an extension.

While existing studies of utility mining have principally centered around the issue of discovering high-utility itemsets from transaction databases assume that every item involves just a single slot. As it may, in some genuine situations, the quantity of slots consumed by various items commonly differs. Consequently, in article [54], the item may physically take any of the fixed number of slots.

7.4.3 Evolutionary Computing based High Utility Itemset Mining and High Utility Association Rule Mining

Mostly, EC based approaches have been applied to mine frequent itemsets as the EC-based techniques can search around non-linear, multimodal and non-continuous search spaces. EC techniques are powerful meta-heuristic search techniques which can solve complex learning problems. Works utilizing EC, in high utility itemset mining are presented in this subsection. But, there are no works reported for mining high utility association rules using EC techniques though two of the heuristic-based approaches are reported in [206, 221, 261]. Also, there are no works reporting top high utility association rules using either with EC-based techniques or heuristic-based techniques.

In [145], ranked mutation operator of the Genetic Algorithm (GA) is employed to solve the HUIM task (i.e. HUPE-GARM). In [145], also developed GA-Tree approach to solving the HUIM task (i.e. HUPE-GARM⁺). Lin et al. in [191] employed Binary PSO (BPSO) to solve the HUIM (i.e. HUPE-BPSO) by employing high-transaction-weighted

utility 1-itemsets (1-HTWUIs). In [190], PSO-Tree based approach is employed to solve the HUIM task (i.e. HUPE-BPSO⁺). In [19, 317], an ant colony optimization (ACO) based algorithm is developed to solve the HUIM task. In [278], an artificial bee colony optimization algorithm is employed to solve the HUIM task. In [279], which is the latest work, where EC techniques namely Bat algorithm, PSO and GA are employed to solve HUIM task. A multi-objective evolutionary approach based frequent and high utility miner is developed in [332]. All the above HUIM miners employed basic EC techniques like GA, PSO, ACO, etc., and none used the state of the art EC techniques like Differential Evolution [74] which is the most powerful and a versatile algorithm

7.5 High Utility Itemset Mining (HUIM) and High Utility Association Rule Mining (HUARM)

The transactional database ' TB ' has ' N ' transactions, i.e. $\{T_1, T_2, T_3, \dots, T_N\}$ and each transaction ' T_i ' contains a subset of items or all items from the set containing ' M ' items, i.e. $\{I_1, I_2, I_3, \dots, I_M\}$. Each item ' I_j ' of transaction ' T_i ' can be picked in a particular quantity ' $Q_{i,j}$ '. Each item ' I_j ' has a corresponding unit profit ' P_j ' (which is positive) associated with it. Let ' K ' be an itemset, i.e. a proper subset of items ' I ' in a transaction ' T_i ' of transactional database ' TB '.

There are two reasons why HUIM is interesting than frequent itemset mining. The first reason is that the HUIM gives us more profit than frequent itemset mining. The second reason is that, in the HUIM, the anti-monotone property or apriori property does not hold, as satisfied by frequent itemset mining. That is a superset may have high, low or equal utility than the subset of an itemset. Therefore, the downward closure property as satisfied by frequent itemset mining does not hold for HUIM, and this can be shown in the example of section 7.5.2. Definitions and measures are given in section 7.5.1.

Association rule mining assumes a vital job in numerous decision-making systems, and association rules help to decide the connections among items in the database. According to [221]: "Conventional association rule mining, in light of the support-confidence, gives us the rules that are important to users. In this methodology, all items are given similar importance by considering the presence of items in a transaction, and the utility of items isn't considered". Therefore, mining high utility association

Table 7.1: Transactional Database with Quantities

Trans.	Items (Quantity)
T_1	$a(2), b(5), c(3)$
T_2	$b(5), c(1), d(3)$
T_3	$d(2), e(4)$
T_4	$a(4), b(1), c(2), d(2), e(1)$
T_5	$c(3), d(3), e(3)$

rules is a fascinating theme of study, which helps numerous businesses to examine their businesses, their sale trends as well as discovering client patterns, to make possible business decisions on the sale and to add more profits.

According to [206, 221]: “An itemset having utility no lower than a minimum utility limit (min-util) is known as a high utility itemset (HUI) and an association rule is having utility-confidence no lower than a minimum utility-confidence limit (min-uconf) is known as a high utility association rule”. This framework in high utility association rule mining is called utility-confidence framework. These definitions and measures for this utility-confidence framework are described in section 7.5.1 and also in the example given in section 7.5.3.

Numerous examinations are finished in HUIM, in which, the utility for the items in the database is considered. These definitions and measures for this utility-confidence framework are described in section 7.5.1 and also in the example given in section 7.5.2.

Consider an example transactional database presented in Table 7.1. T_1 of Table 7.2 has three items which are ‘a’, ‘b’ and ‘c’ with quantities 2, 5, 3 respectively and similarly for other transactions. The unit profits of ‘a’, ‘b’, ‘c’, ‘d’ and ‘e’ are 4, 5, 1, 3, 2.

7.5.1 Definitions of HUIM and HUARM

The definitions for computing the utility of an itemset are taken from [206, 221] and reproduced below:

Def. 1: *Utility of item ‘ I_j ’ with a unit profit ‘ P_j ’ picked in a particular quantity ‘ $Q_{i,j}$ ’ in transaction ‘ T_i ’ is $u(I_j, T_i) = P_j \times Q_{i,j}$.*

E.g. 1: From Table 7.1, U_{11} , i.e. item a with unit profit 4 and quantity 2 is equal to 8, U_{12} i.e. item b with unit profit 5 and quantity 5 is equal to 25, U_{13} i.e. item c with unit profit 1 and quantity 3 is equal to 3, U_{14} i.e. item d with unit profit 3 and

quantity 0 is equal to 0 and U_{15} , i.e. item e with unit profit 2 and quantity 0 is equal to 0.

Def. 2: The utility of an itemset ‘ X ’ in transaction ‘ T_i ’ is denoted as $u(X, T_i)$, and defined as $u(X, T_i) = \sum_{I_j \subseteq X \wedge X \subseteq T_i} u(I_j, T_i)$

E.g. 2: From Table 7.1, utility of itemset b, c in transaction T_1 is $u(bc, T_1) = u(I_2, T_1) + u(I_3, T_1) = 25 + 3 = 28$

Def. 3: The utility of an itemset ‘ X ’ in a database ‘ TB ’ is denoted as $u(X)$, and defined as

$$u(X) = \sum_{X \subseteq T_i \wedge T_i \subseteq TB} u(X, T_i)$$

E.g. 3: From Table 7.1, utility of itemset b, c in transaction database TB is $u(bc) = u(I_2, T_1) + u(I_3, T_1) + u(I_2, T_2) + u(I_3, T_2) + u(I_2, T_4) + u(I_3, T_4) = 25 + 3 + 25 + 1 + 5 + 2 = 61$

Def. 4: The transaction utility of a transaction ‘ TB ’ is denoted as $tu(T_i)$ and defined as

$$tu(T_i) = \sum_{X \subseteq T_i} u(X, T_i)$$

E.g. 4: From Table 7.1, utility of all items in transaction ‘ T_1 ’ is $tu(T_1) = u(I_1, T_1) + u(I_2, T_1) + u(I_3, T_1) + u(I_4, T_1) + u(I_5, T_1) = 8 + 25 + 3 + 0 + 0 = 36$

Def. 5: The total utility of a database ‘ TB ’ is denoted as ‘ TU ’ and defined as:

$$TU = \sum_{T_i \subseteq TB} tu(T_i)$$

E.g. 5: From Table 7.1, utility of all items in ‘ TB ’ is $TU = tu(T_1) + tu(T_2) + tu(T_3) + tu(T_4) + tu(T_5) = 36 + 35 + 14 + 31 + 18 = 134$

Def. 6: High utility itemset of a transactional database is $HUI = \{X \mid u(X) \geq TU \times \alpha\}$ where ‘ α ’ are the specific thresholds for minimum utility.

E.g. 6: From Table 7.1, itemset b, c is a HUI as $u(bc) = 61 \geq 134 \times 0.25$ here total utility of ‘ TB ’ i.e. $TU = 134$ and threshold ‘ α ’ = 25%.

Def. 7: According to [221]: “The local utility value of an item I_i in an itemset X , denoted by $lvu(I_i, X)$, is the sum of utility values of the items I_i in all the transactions containing X ”

$$lvu(I_i, X) = \sum_{X \subseteq T_i \wedge T_i \subseteq TB} u(I_i, T_i)$$

E.g. 7: From Table 7.1, $luv(c, bc) = U_{13} + U_{23} + U_{43} = 3+2+1 = 6$

Def. 8: According to [221]: “The local utility value of an itemset X in another itemset Y such that $X \subseteq Y$, denoted by $luv(X, Y)$, is the sum of local utility measure values of each item $x_i \in X$ in itemset Y , which is given by $luv(X, Y) = \sum_{I_i \in X \subseteq Y} luv(I_i, Y)$ ”.

E.g. 8: From Table 7.1, $luv(cd, bcd) = U_{23} + U_{24} + U_{43} + U_{44} = 1+ 9 + 2 + 6 = 18$

Def. 9: According to [221]: “A high utility association rule ‘ R ’ is an implication between two high utility itemsets. $A, B \subseteq I$ of the form $A \rightarrow B$, utility confidence of rule R , denoted as $uconf(R)$, is defined as $uconf(R) = \frac{luv(X, XY)}{u(X)}$. Percentage of $uconf(R) = uconf(R) \times 100$.”

E.g. 9: From Table 7.1, $uconf$ of $c \rightarrow b = \frac{luv(c,bc)}{u(c)} = \frac{U_{13} + U_{23} + U_{43}}{U_{13} + U_{23} + U_{43} + U_{53}} = \frac{6}{9} = 0.6666 \Rightarrow 66.66\%$ of $uconf$.

Def. 10: According to [221]: “Let $R1: A1 \rightarrow B1$ and $R2: A2 \rightarrow B2$ be two valid association rules in a utility-confidence framework. $R2$ is redundant with regard to $R1$ if $A2 \cup B2 \subseteq A1 \cup B1$, $R1.Utility \geq R2.Utility$, $support(R1) = support(R2)$ and $A1 \subseteq A2, B2 \subseteq B1$ where $Ri.Utility$ is the utility of the rule Ri , $i = 1, 2$, and the support of a rule $R: A \rightarrow B$ is $support(A \cup B)$ ”.

Def. 11: According to [221]: “The support of an itemset K is an indication of how frequently that K appears in database ‘ TB ’. The support value of K in ‘ TB ’ is defined as the proportion of itemsets in a database containing K , denoted as $support(K)$ ”.

E.g. 10: From Table 7.1, $support$ of $c \rightarrow b = support(b,c) = \frac{3}{5} = 0.6 \Rightarrow 60\%$ of $support$.

7.5.2 HUIs Mined for the Example

Table 7.2 presents the HUIM results of the example ‘ TB ’ shown in Table 7.1. In Table 7.2 we can see that itemset i.e. ‘ d, e ’ though frequent it is not the HUI. Also, itemsets ‘ a, b, c ’ and ‘ a, b ’ violate the anti-monotone property, i.e. ‘ a, b, c ’ is having high utility than ‘ a, b ’.

7.5.3 HUARMs mined for the Example

The HUARM results are presented below for the example ‘ TB ’ shown in Table 7.1 and these rules have crossed the minimum threshold for association rule mining. Rules

Table 7.2: Itemsets with a threshold of 25 on Utility

High Util- ity Itemset	Support	Utility
<i>b, c</i>	60	61
<i>a, b, c</i>	40	59
<i>a, b</i>	40	54
<i>b, c, d</i>	40	48
<i>b, d</i>	40	45
<i>d, e</i>	60	37
<i>a, b, c, d, e</i>	20	31
<i>c, d</i>	60	30
<i>a, b, d, e</i>	20	29
<i>a, b, c, d</i>	20	29
<i>a, b, d</i>	20	27
<i>a, c, d, e</i>	20	26

that are non-redundant are presented below. Association Rules with a threshold on the product of 25% of Total Utility and 97.5% of Utility Confidence

1. $b \rightarrow c$ Support: 60, Confidence: 100, Utility: 61,
UConfidence: 100, Product: 6100
2. $a \rightarrow b, c$ Support: 40, Confidence: 100, Utility: 59,
UConfidence: 100, Product: 5900 2
3. $c \rightarrow b$ Support: 60, Confidence: 75, Utility: 61,
UConfidence: 66.66, Product: 4066.66
4. $e \rightarrow d$ Support: 60, Confidence: 100, Utility: 37,
UConfidence: 100, Product: 3700
5. $c \rightarrow a, b$ Support: 40, Confidence: 50, Utility: 59,
UConfidence: 55.55, Product: 3277.77

7.6 Binary Differential Evolution (BDE) and Adaptive Binary Differential Evolution (ABDE)

7.6.1 Differential Evolution

Storn and Price developed DE [293] in 1997 for continuous optimization problems. DE incorporates four critical segments, which are fundamentally initialization, mutation, recombination and selection. At each iteration, new solutions are made with the mix of solutions of the present set of solutions which are called mutations. The solutions that at this stage are joined with a target solution in a procedure called crossover. Finally, considering fitness, the solutions are chosen. Let a , b , c be the three solutions vectors picked randomly from the solution vectors and the accompanying equation operators are utilized for mutation and crossover. The code snippet for the mutation and crossover is given below:

If $(\text{rand}(0, 1) < \text{CR} \parallel i = R)$ then $u_i = a_i + F \times (b_i - c_i)$

If $(\text{rand}(0, 1) > \text{CR} \ \& \ i \neq R)$ then $u_i = x_{\text{current}}$

DE/rand/1/bin standard variant of DE is employed. Differential weight, ‘F’, is between 0 and 2. Crossover Rate ‘CR’, is between 0 and 1.

The reasons for choosing DE are as follows [73, 74]: 1) DE can work around non-linear, non-differentiable, and multi-modular search spaces. 2) It is genuinely robust and quick. 3) Though intended to take care of real-valued problem spaces, it can also be utilized for discrete problem spaces. The drawback of Differential Evolution is that of the encoding which is difficult to be characterized in non-real-valued problem spaces.

7.6.2 Adaptive Differential Evolution

ADE considered changing DE with two new ideas: 1) Roulette Wheel Selection [25] is employed on the 12 parameter sets of Crossover Rate (CR) and Differential Weight (F) based on the success rate of the selection and 2) Random Localization [144]. The adjustment of ADE parameters occurs through Roulette Wheel Selection. In [304], the Roulette Wheel Selection is utilized to choose the better set of parameters dependent on the success rate of the choice of each set of parameters. There are twelve sets of parameters. Six utilize binomial crossover, and another six utilize exponential crossover.

All the twelve sets of parameters utilize random localization in the mutation procedure. In the adaptation procedure, the twelve sets of CR and F are recognized by 12 unique probabilities q_h , $h = 1, 2, \dots, H$ (here H is 12). The success rate changes these probabilities in the past iterations. In the event that $f(y) > f(x_i)$ then the h -th setting is selected.

In [304], a random localization approach is presented for DE. Rather than choosing the current vector arbitrarily, select the best among the three chosen vectors, and this saves around 35% of the objective function evaluations [304]. The mutant vector 'u' is generated from three arbitrarily chosen vectors a , b , c . Of the three, the best vector is chosen as 'a'.

$$u_i = a_i + F \times (b_i - c_i)$$

The new vector is processed from the mutant vector 'u' and the original vector by performing either binomial or exponential crossover. The purpose behind picking the adaptive version of DE is the dynamic mechanism of determination of best control parameters of the DE, which thus decreases the control parameter tuning time.

7.6.3 Binary Differential Evolution and Adaptive Binary Differential Evolution

The DE was initially developed to take care of the problems in continuous space. Therefore in [91], to apply DE for binary spaces, Engelbrecht and Pampara developed a binary version of DE, i.e. Binary DE (BDE) by passing the continuous values of DE to a sigmoid function and then comparing the result of a sigmoid function with a random number generated between 0 to 1. The solution vectors generated by DE in each of the iterations are discretized in the manner as follows. Discretize solution vector 'x' with a sigmoid function.

$$w_i = \begin{cases} 1 & \text{if}(\text{rand}(0, 1) \leq \text{Sigmoid}(x_i)) \\ 0 & \text{if}(\text{rand}(0, 1) > \text{Sigmoid}(x_i)) \end{cases}$$

Where 'w' is a binary vector.

The procedure mentioned above is also followed for the discretization of the continuous ADE to a binary version, i.e. ABDE. By this binary discretization step BDE and ABDE become combinatorial in nature.

7.7 Proposed High Utility Itemset Miners

BDE and ABDE based high utility itemset miners contain five important steps, which are mainly initialization, mutation, recombination, discretization, and selection. New solutions are generated from the current set of solutions using ‘mutation’ and ‘crossover’. Then ‘selection’ is performed to select the fittest. BDE and ABDE have two parameters which are CR and F. The following subsections briefly describe the working of BDE and ABDE based high utility itemset miner, which is also illustrated in Figure 7.2.

In this work, we pursued the Michigan approach for itemset encoding. If be ‘M’ the number of items in the dataset. For every item, there are two conceivable values for each item, i.e. either 1 or 0. The value 1 in the implies that the item is included in the itemset. On the off chance the value of 0, the item is absent in the itemset and illustrated in Figure 7.1.

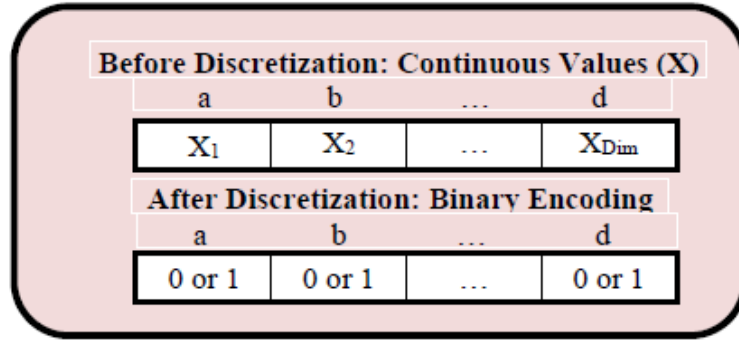


Figure 7.1: Solution Encoding Scheme for HUIM

7.7.1 BDE and ABDE based HUIM

7.7.1.1 Initialization

Initialise NP, the number solution vectors of dimensionality size, Dim for HUIM (where Dim is the number of Items). Check whether the solution vectors are within bounds i.e. between -10 to 10. Discretise the continuous solution vectors x_i to binary solution vectors y_i using the sigmoid function (i.e. X and Y in algorithms 1 and 2). Compute the fitness values of the NP number of solution vectors as in subsections 7.7.1.5.

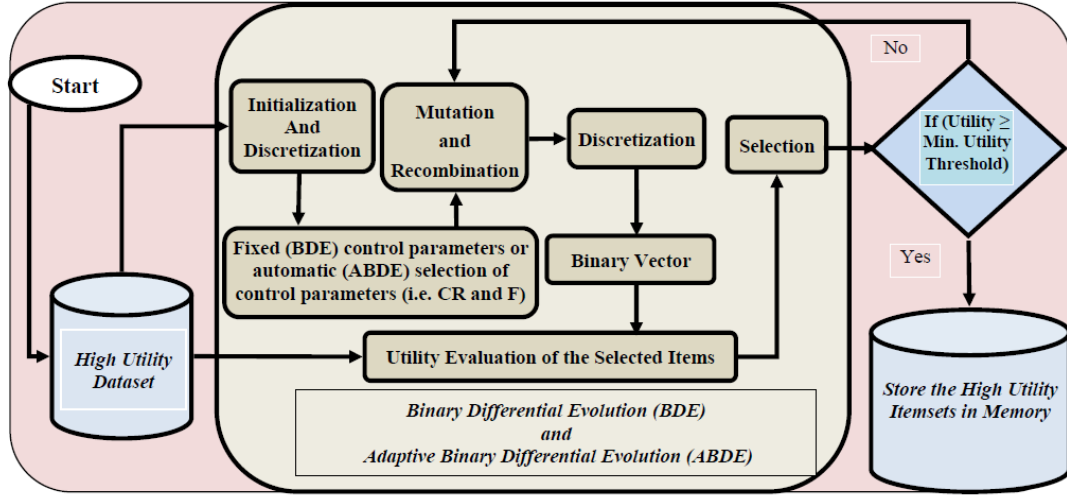


Figure 7.2: BDE and ABDE-based HUIM Algorithms Process Flow

7.7.1.2 Mutation

Mutation for BDE:

Let x_1 , x_2 , x_3 , and $x^{current}$ be the three unique solution vectors chosen randomly from the population or solution vectors. Perform the mutation using the following equation

$$u'_i = x_{1,i} + F * (x_{2,i} - x_{3,i}) \quad (1a)$$

The above equation is a standard variant of DE i.e. of DE/rand/1/bin. Differential weight, 'F', is between 0 and 2.

Mutation for ABDE:

Let x_1 , x_2 , x_3 and $x^{current}$ be the three unique solution vectors chosen randomly from the population or solution vectors. The mutant vector 'u' is generated from three arbitrarily chosen vectors x_1 , x_2 , x_3 . Of the three, the best vector in terms of fitness value is chosen as ' x_1 '. Perform the mutation using the following equation

$$u'_i = x_{1,i} + F * (x_{2,i} - x_{3,i}) \quad (1b)$$

The above equation is a variant of DE i.e. of DE/randl, where 'randl' stands random localization [144]. Differential weight, 'F', is either 0.5 or 0.8.

7.7.1.3 Recombination

Recombination in BDE:

Using the mutation vector and crossover rate parameter CR recombination is performed as follows:

$$v_i = \begin{cases} u_i & \text{if } (\text{rand}(0, 1) \leq CR \mid i = I_{rand}) \\ x_i^{current} & \text{if } (\text{rand}(0, 1) > CR \text{ and } i \neq I_{rand}) \end{cases} \quad (2)$$

CR is between 0 and 1. I_{rand} is a random dimension between 1 to Dim and I_{rand} is used so that v and $x^{current}$ are different.

Recombination in ABDE:

1. In the binomial crossover (DE/././bin expresses the exponential crossover), the changed elements are scattered arbitrarily from 1 to D. The binomial crossover in ABDE uses three values of CR ($CR = \{0, 0.5, 1\}$) for the two values of F (i.e. $F = 0.5$ and $F = 0.8$). Therefore, a total of 6 combinations.
2. For the exponential crossover (DE/././exp expresses the exponential crossover), the beginning position of crossover is picked arbitrarily from 1 to D, and L consecutive elements (included in a roundabout manner) are taken from the mutant vector v . Likelihood of replacing the k th component in the elements from 1, 2, . . . , L with $L \leq D$, diminishes exponentially with expanding k .
3. The three estimations of CR for exponential crossover are derived from the roots of a particular polynomial [305], which intern uses three values of the likelihood, i.e. p_m which is set up equidistantly within the bounds of $(1/D, 1)$. The exponential crossover in ABDE uses three estimated values of CR along with the two values of F (i.e. $F = 0.5$ and $F = 0.8$). Therefore, a total of 6 combinations.
4. These 12 altogether different parameter settings give a fair chance for adjusting exploration and exploitation in the diverse phases of the optimization process [304].

7.7.1.4 Discretization

DE is developed for real input parameter-based optimization problems. Therefore, the discretization of the decision variables to binary values is necessary. The resulting

binary version of DE is called BDE or ABDE [91] and illustrated in Figure 7.1.

Discretize v with a sigmoid function.

$$w_i = \begin{cases} 1 & \text{if}(\text{rand}(k, 1) \leq \text{Sigmoid}(v_i)) \\ 0 & \text{if}(\text{rand}(k, 1) > \text{Sigmoid}(v_i)) \end{cases} \quad (3)$$

Where w is a binary vector or the itemset and k is the bias which lies between 0 and 1 (open interval)

7.7.1.5 Fitness Evaluation

For BDE or ABDE based HUIM:

For the binary vector or itemset w , compute the fitness function (F_1), i.e. utility value U_{K-TB} , of the itemset which is to be maximized.

$$F_1 = \text{maximize}\{u(X)\} \quad (4)$$

These definitions and computations required for calculation of ' $u(X)$ ' is presented in the sections 7.5.1 and 7.5.2.

7.7.1.6 Selection

For BDE or ABDE based HUIM:

If $\text{Fitness}(w) \leq \text{Fitness}(y^{\text{current}})$ replace current vectors of y and x in the population (i.e. both binary and continuous) with solution vectors w and v (i.e. both binary and continuous). Else do not replace the current vectors.

If $\text{Fitness}(y^{\text{current}}) \geq [\text{Minimum Utility Threshold, i.e. (Total Utility of TB, i.e. } U_{TB} \times \alpha)]$ then store this itemset or solution in the memory. Where ' α ' are the specific threshold percentages for the minimum utility of a TB. Repeat steps in subsections 7.7.1.2 to 7.7.1.6 until a maximum number of iterations is met.

HUIs crossing Threshold, i.e. Total Utility of TB, $U_{TB} \times \alpha$ are sorted in decreasing order of $U_{TB} \times \alpha$. ' α ' are specific threshold percentages Tables 7.4. Memory list is a data structure to hold the final itemsets. Please refer Algorithms 7 and 8 for the detailed steps of implementation of the BDE and ABDE based HUI miners.

Algorithm 7: Binary Differential Evolution (BDE) based HUIM

Input: X (i.e. Solution vectors), Y (i.e. a vector containing fitness values for the corresponding solution vectors), NP (i.e. size of the population or solution vectors), CR, F

$I = 1$
 $J = 1$
BDE based HUIM(X, Y)
{
 while $I \leq \text{MaxIterations}_{BDE}$ **do**
 while $J \leq NP$ **do**
 Choose solution vectors x_1, x_2, x_3 randomly
 as mutant vectors from $X_{ContinuousI}$
 Perform Mutation
 Perform Binomial Crossover
 Perform Binary Discretization of the
 new solution vector x_{newI} using Sigmoid Function
 $\hat{x}_I$ is the new binary solution vector
 Apply Fitness Function (F_1)
 $\hat{y}_I = \text{ComputeFitness-}F_1(\hat{x}_I)$
 if $\hat{y}_I$ better than the fitness of the current solution vector in X_I **then**
 Replace current vector by $\hat{x}_I$ in $X_{BinaryI}$
 Replace current vector by x_{newI} in $X_{ContinuousI}$
 Also replace the corresponding fitness value with $\hat{y}_I$
 end
 if $\hat{y}_I$ greater than minimum itemset threshold **then**
 Add the itemset to the Memory List M_I .
 end
 $J = J + 1$
 end
 $I = I + 1$
 end
 return (M_I)
}

Algorithm 8: Adaptive Binary Differential Evolution (ABDE) based HUIM

Input: X (i.e. Solution vectors), Y (i.e. a vector containing fitness values for the corresponding solution vectors), NP (i.e. size of the population or solution vectors)

$I = 1$
 $J = 1$
ABDE based HUIM(X, Y)
{
 while $I \leq \text{MaxIterations}_{ABDE}$ **do**
 while $J \leq NP$ **do**
 Perform Roulette Wheel Selection to choose between the 12 parameter sets (of CR and F) based on the success rate of each set of parameters
 Choose solution vectors x_1, x_2, x_3 randomly as mutant vectors from $X_{ContinuousI}$
 Select the best among three as the current solution vector (Random Localization)
 Perform Mutation
 Perform Binomial or Exponential Crossover
 Perform Binary Disretization of the new solution vector x_{newI} using Sigmoid Function
 $\hat{x}_I$ is the new binary solution vector
 Apply Fitness Function (F_1)
 $\hat{y}_I = \text{ComputeFitness-}F_1(\hat{x}_I)$
 if $\hat{y}_I$ better than the fitness of the current solution vector in X_I **then**
 Replace current vector by $\hat{x}_I$ in $X_{BinaryI}$
 Replace current vector by x_{newI} in $X_{ContinuousI}$
 Also replace the corresponding fitness value with $\hat{y}_I$
 end
 if $\hat{y}_I$ greater than minimum itemset threshold **then**
 Add the itemset to the memory list M_I .
 end
 $J = J + 1$
 end
 $I = I + 1$
 end
 return (M_I)
}

7.8 Proposed High Utility Association Rule Miners

BDE and ABDE based high utility association rule miners contain five important steps, which are mainly initialization, mutation, recombination, discretization, and selection. New solutions are generated from the current set of solutions using ‘mutation’ and ‘crossover’. Then ‘selection’ is performed to select the fittest. BDE and ABDE have two parameters which are CR and F. The following subsections briefly describe the working of BDE and ABDE based high utility association rule miner, which is also illustrated in Figure 7.4.

In this work, we pursued the Michigan approach for rule encoding. Let ‘M’ be the number of items in the dataset. There are two binary parts for each item in the association rule encoding, unlike itemset coding which has only one binary part for each item. The value 1 in the initial part implies that the item is included in the rule. The value of 0 means the item is not included in the rule. The second part of the rule indicates whether the item ought to be incorporated into the antecedent or consequent part of the rule. On the offset it means if 1, the item is part of the consequent of the rule. If the two binary parts of the I_j are 10, then the j^{th} item is included in the rule and is incorporated into the antecedent part of the rule. On the off chance that the binary values of I_j are 11, the j^{th} item is included and part of the consequent part of the rule. If the binary parts of the I_j are 00 or 01, then the j^{th} item is excluded from the rule. In Figure 7.3, green color represents antecedent and blue color represents consequent.

Before Discretization: Continuous							
a	b		c		d		
5.5	2.3	-1.2	3.4	7.5	8.3	6.2	1.4
After Discretization: Binary Encoding							
a	b		c		d		
0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1

Figure 7.3: Before and after binary discretization of the high utility encoded association rule

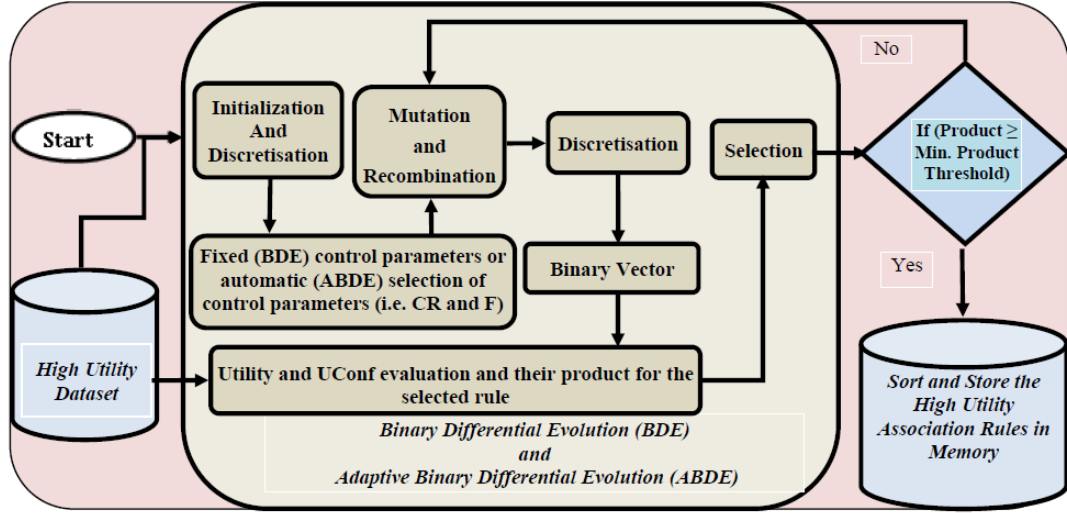


Figure 7.4: BDE and ABDE-based HUARM Algorithms Process Flow

7.8.1 BDE and ABDE based HUARM

7.8.1.1 Initialization

Initialise NP, the number solution vectors of dimensionality size, $2 \times \text{Dim}$ for HUARM (where Dim is the number items). Check whether the solution vectors are within bounds i.e. between -10 to 10. Discretise continuous solution vectors x_i to binary solution vectors y_i using the sigmoid function (i.e. X and Y in algorithms 1 and 2). Compute the fitness values of the NP number of solution vectors as in subsections 7.8.1.5.

7.8.1.2 Mutation

Mutation for BDE:

Let x_1, x_2, x_3 , and x^{current} be the three unique solution vectors chosen randomly from the population or solution vectors. Perform the mutation using the following equation

$$u_i' = x_{1,i} + F * (x_{2,i} - x_{3,i}) \quad (1a)$$

The above equation is a standard variant of DE i.e. of DE/rand/1/bin. Differential weight, 'F', is between 0 and 2.

Mutation for ABDE:

Let x_1 , x_2 , x_3 and $x^{current}$ be the three unique solution vectors chosen randomly from the population or solution vectors. The mutant vector 'u' is generated from three arbitrarily chosen vectors x_1 , x_2 , x_3 . Of the three, the best vector in terms of fitness value is chosen as ' x_1 '. Perform the mutation using the following equation

$$u'_i = x_{1,i} + F * (x_{2,i} - x_{3,i}) \quad (1b)$$

The above equation is a variant of DE i.e. of DE/rand/l, where 'rand/l' stands random localization [144]. Differential weight, 'F', is either 0.5 or 0.8.

7.8.1.3 Recombination

Recombination in BDE:

Using the mutation vector and crossover rate parameter CR recombination is performed as follows:

$$v_i = \begin{cases} u_i & \text{if } (\text{rand}(0, 1) \leq CR \mid i = I_{rand}) \\ x_i^{current} & \text{if } (\text{rand}(0, 1) > CR \text{ and } i \neq I_{rand}) \end{cases} \quad (2)$$

CR is between 0 and 1. I_{rand} is a random dimension between 1 to Dim and I_{rand} is used so that v and $x^{current}$ are different.

Recombination in ABDE:

1. In the binomial crossover (DE/././bin expresses the exponential crossover), the changed elements are scattered arbitrarily from 1 to D. The binomial crossover in ABDE uses three values of CR ($CR = \{0, 0.5, 1\}$) for the two values of F (i.e. $F = 0.5$ and $F = 0.8$). Therefore, a total of 6 combinations.
2. For the exponential crossover (DE/././exp expresses the exponential crossover), the beginning position of crossover is picked arbitrarily from 1 to D, and L consecutive elements (included in a roundabout manner) are taken from the mutant vector v. Likelihood of replacing the kth component in the elements from 1, 2, . . . , L with $L \leq D$, diminishes exponentially with expanding k.
3. The three estimations of CR for exponential crossover are derived from the roots of a particular polynomial [305], which intern uses three values of the likelihood, i.e. p_m which is set up equidistantly within the bounds of $(1/D, 1)$. The exponential

crossover in ABDE uses three estimated values of CR along with the two values of F (i.e. $F = 0.5$ and $F = 0.8$). Therefore, a total of 6 combinations.

4. These 12 altogether different parameter settings give a fair chance for adjusting exploration and exploitation in the diverse phases of the optimization process [304].

7.8.1.4 Discretization

DE is developed for real input parameter-based optimization problems. Therefore, the discretization of the decision variables to binary values is necessary. The resulting binary version of DE is called BDE or ABDE [91] and illustrated in Fig. 7.3.

Discretize v with a sigmoid function.

$$w_i = \begin{cases} 1 & \text{if}(\text{rand}(k, 1) \leq \text{Sigmoid}(v_i)) \\ 0 & \text{if}(\text{rand}(k, 1) > \text{Sigmoid}(v_i)) \end{cases} \quad (3)$$

Where w is a binary vector or the rule and k is the bias which lies between 0 and 1 (open interval)

7.8.1.5 Fitness Evaluation

For BDE or ABDE based HUARM:

For the association rule encoded vector w , compute the fitness function (F_2) i.e. product of utility value, U_{K-TB} , and Utility Confidence, $UConf$, which is to be maximized.

$$F_2 = \text{maximize}\{u(X) \times UConf\} \quad (4)$$

These definitions and computations required for calculation of ' $u(X)$ ' and ' $Uconf$ ' is presented in the sections 7.5.1 and 7.5.2

$Uconf$ is a measure presented in [221] and the maximum value of this measure tells the confidence of a high utility rule. There might be multiple rules with high ' $Uconf$ ' measure. Therefore we choose to multiply it with utility value of the rule to discriminate high valued association rules.

7.8.1.6 Selection

For BDE or ABDE based HUARM:

If $\text{Fitness}(w) \leq \text{Fitness}(y^{\text{current}})$ replace current vectors of y and x in the population (i.e. both binary and continuous) with solution vectors w and v (i.e. both binary and continuous). Else do not replace the current vectors.

If $\text{Fitness}(y^{\text{current}}) \geq [\text{Minimum Utility Threshold, i.e. } (U_{TB} \times \alpha)] \times [\text{Minimum UConf, i.e. } (\text{Maximum Uconf} \times \beta)]$ then store this rule or solution in the memory. Where ' α ' are the specific threshold percentages for the minimum utility of a TB. Where ' β ' are the specific threshold percentages for the minimum utility confidence of a TB. Repeat from steps in subsections 7.8.1.2 to 7.8.1.6 until a maximum number of iterations is met.

High utility association rules crossing $(U_{TB} \times \alpha) \times (\text{Maximum Uconf} \times \beta)$ are sorted in decreasing order of the above product. ' α ' and ' β ' are specific threshold percentages Tables 7.4 and 7.5. Memory list is data structure to hold the final rules. Sort the memory list according to objective function and present the top 5 rules. Please refer Algorithms 9 and 10 for the detailed steps of implementation of BDE and ABDE based HUARMs.

Algorithm 9: Binary Differential Evolution (BDE) based HUARM

Input: X (i.e. Solution vectors), Y (i.e. a vector containing fitness values for the corresponding solution vectors), NP (i.e. size of the population or solution vectors), CR, F

$I = 1$

$J = 1$

BDE based HUARM(X, Y)

{

while $I \leq \text{MaxIterations}_{BDE}$ **do**

while $J \leq NP$ **do**

Choose solution vectors x_1, x_2, x_3 randomly

as mutant vectors from $X_{ContinuousR}$

Perform Mutation

Perform Binomial Crossover

Perform Binary Discretization of the

new solution vector x_{newR} using Sigmoid Function

$\hat{x}_I$ is the new binary solution vector

Apply Fitness Function (F_2)

$\hat{y}_R = \text{ComputeFitness-}F_2(\hat{x}_R)$

if $\hat{y}_R$ better than the fitness of the current solution vector in X_R

then

Replace current vector by $\hat{x}_I$ in $X_{BinaryR}$

Replace current vector by x_{newI} in $X_{ContinuousR}$

Also replace the corresponding fitness value with $\hat{y}_R$

end

if $\hat{y}_R$ greater than minimum association rule threshold **then**

Add the association rule to the Memory List M_R .

end

$J = J + 1$

end

$I = I + 1$

end

Sort the association rules in descending order of the objective function.

return (M_R)

}

Algorithm 10: Adaptive Binary Differential Evolution (ABDE) based HUARM

Input: X (i.e. Solution vectors), Y (i.e. a vector containing fitness values for the corresponding solution vectors), NP (i.e. size of the population or solution vectors)

$I = 1$

$J = 1$

ABDE based HUARM(X, Y)

{

while $I \leq \text{MaxIterations}_{ABDE}$ **do**

while $J \leq NP$ **do**

Perform Roulette Wheel Selection to choose between the 12 parameter sets (of CR and F) based on the success rate of each set of parameters

Choose solution vectors x_1, x_2, x_3 randomly as mutant vectors from $X_{ContinuousR}$

Select the best among three

as the current solution vector (Random Localization)

Perform Mutation

Perform Binomial or Exponential Crossover

Perform Binary Discretization of the

new solution vector x_{newR} using Sigmoid Function

$\hat{x}_R$ is the new binary solution vector

Apply Fitness Function (F_2)

$\hat{y}_R = \text{ComputeFitness-}F_2(\hat{x}_R)$

if $\hat{y}_R$ better than the fitness of the current solution vector in X_R

then

Replace current vector by $\hat{x}_R$ in $X_{BinaryR}$

Replace current vector by x_{newR} in $X_{ContinuousR}$

Also replace the corresponding fitness value with $\hat{y}_R$

end

if $\hat{y}_R$ greater than minimum association rule threshold **then**

Add the association rule to the Memory List M_R .

end

$J = J + 1$

end

$I = I + 1$

end

Sort the association rules in descending order of the objective function.

return (M_R)

}

7.9 Results and Discussion

The present work is performed on Intel Core i7 processors with 32 GB RAM and implemented in R. The proposed BDE and ABDE based HUIM is compared with the work in [190, 191]. Proposed HUIM algorithms are only compared with standard extant evolutionary computing based HUIM algorithms. The proposed HUIM algorithms are run for 10,000 iterations and with a population size of 20. Its convergence is also compared with two-phase HUIM [195]. BDE based HU-ARM is contrasted with ABDE based HUARM as there no evolutionary computing techniques-based HUARM algorithms. The proposed HUARM algorithms are also run for 10,000 iterations and with a population size of 20. Two metrics are compared, i.e. 1) Number of HUIs mined at various thresholds and 2) convergence at a particular threshold. Six high utility datasets and one real-world UCI *OnlineRetail* dataset [56] (i.e. UCI dataset is preprocessed to the application of customer segmentation based on value) are analyzed in the current study for the comparison. The difference between the six high utility datasets and preprocessed UCI *OnlineRetail* dataset is that the in preprocessed UCI repository dataset, each row represents a unique customer performing multiple transactions for the particular items. The seven high utility datasets are presented in Table 7.3. For the *Accidents* dataset, the top 10% of the transactions are used for the HUIM. The *OnlineRetail* dataset is preprocessed by removing the rows where the customer ID is not present and where the quantity of the items is negative. The reason for choosing these datasets is the high dimensionality associated with them and not the number of transactions, as we want to show the power of the chosen EC techniques in mining HUIs. Another reason is that we want to compare with the state-of-the-art EC techniques and show that our proposed HUIM algorithm is powerful in mining most of the HUIs crossing the threshold. The *Accidents* dataset with individual items below 25% support, *Foodmart* dataset with individual items below 0.35% support and *Retail* dataset with individual items below 0.4% support are dropped. Therefore, *Accidents* dataset has 40 items, *Foodmart* dataset has 318 items and *Retail* dataset has 317 items left after above preprocessing phase. These are presented in Table 7.3 in parenthesis.

The algorithm is run for 25 runs, and the results of the median-run in terms of the number of itemsets or rules mined are reported to nullify the effect of random seed in the optimization process (However, the best-run results surpass all the compared extant

Table 7.3: Datasets

Datasets	No. of Transactions	No. of Items
<i>Chess</i>	3196	75
<i>Mushroom</i>	8124	119
<i>Connect</i>	67,557	129
<i>Accidents_10%</i>	34,018	468 (40)
<i>Foodmart</i>	4141	1559 (318)
<i>Retail</i>	88,162	16,470 (317)
<i>OnlineRetail</i>	4339 (Customers)	3877 (Products)

methods). For the BDE, a generalist set of optimal parameters is arrived at by grid search for all seven datasets. Different thresholds are set for seven high utility datasets (see Tables 7.4 and 7.5). The itemsets crossing the threshold are stored in a flat text file in the memory. Thresholds are needed to check the speed of convergence of the proposed algorithm. It also helps in dealing with highly sparse transactional datasets where each of the transactions is sparse in the number of items and also sparse in the average transaction length.

Table 7.4: Thresholds on Utility

Datasets	Various Threshold Values of ' α ' set on Total Utility of TB
<i>Chess</i>	25%, 25.5%, 26%, 26.5%, 27%
<i>Mushroom</i>	14%, 14.25%, 14.5%, 14.75%, 15%
<i>Connect</i>	28.9%, 29.1%, 29.3%, 29.5%, 29.7%
<i>Accidents_10%</i>	13.1%, 13.4%, 13.7%, 14%, 14.3%
<i>Foodmart</i>	0.04%, 0.045%, 0.05%, 0.055%, 0.06%
<i>Retail</i>	0.3%, 0.4%, 0.5%, 0.6%, 0.7%
<i>OnlineRetail</i>	0.05%, 0.055%, 0.06%, 0.065%, 0.07%

7.9.1 Results of HUIM

7.9.1.1 Number of Itemsets Mined at various Thresholds

Table 7.6 presents that the proposed HUIM-BDE and HUIM-ABDE could mine an additional number of itemsets than extant methods, i.e. HUI-BPSO⁺ [190], HUI-BPSO [191], HUPE-GARM⁺ [145] and HUPE-GARM [145]. Also, the results show that the number of HUIs mined by HUIM-BDE did not vary drastically at various thresholds

Table 7.5: Thresholds on Utility Confidence (UConf)

Datasets	Various Threshold Values of ‘ β ’ set on Maximum UConf
Chess	97.5%, 98%, 98.5%, 99%, 99.5%
Mushroom	97.5%, 98%, 98.5%, 99%, 99.5%
Connect	97.5%, 98%, 98.5%, 99%, 99.5%
Accidents_10%	82.5%, 83%, 83.5%, 84%, 84.5%
Foodmart	60%, 62.5%, 65%, 67.5%, 70.0%
Retail	60%, 62.5%, 65%, 67.5%, 70.0%
OnlineRetail	60%, 62.5%, 65%, 67.5%, 70.0%

and HUIM-BDE could mine more itemsets than the remaining five algorithms in three datasets, i.e. *Chess*, *Mushroom* and *Connect*. Also, there is not much variance among the HUIs mined among *Accidents*, *Foodmart* and *Retail* datasets. Also, these are the results of median run of number of itemsets to quantify statistically and the best run results in all the datasets far supersede the extant methods [278, 279].

7.9.1.2 Convergence of proposed HUIMs with extant methods

Figures 7.5 to 7.10 illustrate the convergence of the proposed HUIM-BDE and HUIM-ABDE at a fixed threshold with extant methods, i.e. two-phase HUIM (i.e T-HUIM) [195], HUI-BPSO, and HUPE-GARM. As you can see from the *Chess* and *Connect* dataset convergence plots that the highest number of itemsets are mined by the proposed HUI miners and also the proposed HUI miners are able to mine an optimal number of itemsets the case in the *Mushroom*, *Accidents* and *Retail* datasets. Also, faster convergence is achieved on *Chess*, *Mushroom*, *Connect* and *Accidents* datasets than the extant methods. But, to contrast various algorithms we kept fixed number of iterations. Also, these are the results of median run of number of itemsets to quantify statistically and the best run results in all the datasets far supersede the extant methods. The reason why BDE is performing exceedingly well over other HUIM techniques is that it does better exploration and exploitation at the selected control parameter setting.

Dataset	Thres-holds	HUIM-BDE	HUIM-ABDE	HUI-BPSO+	HUI-BPSO	HUPE-GARM+	HUPE-GARM
<i>Chess</i>	25%	1444	922	83	75	11	12
	25.5%	1093	776	41	36	9	8
	26%	840	617	16	16	3	5
	26.5%	597	493	5	5	1	0
<i>Connect</i>	27%	408	387	1	1	0	0
	28.9%	547	112	73	69	10	11
	29.1%	454	78	40	35	7	9
	29.3%	386	48	10	10	1	2
	29.5%	298	39	8	8	1	1
	29.7%	230	28	4	4	0	0
<i>Mush-room</i>	14%	400	36	33	24	8	9
	14.25%	285	21	21	15	4	5
	14.5%	195	10	10	8	1	2
	14.75%	132	4	6	3	0	0
	15%	88	1	1	1	0	0
<i>Accidents (10%)</i>	13.1%	110	72	108	45	12	8
	13.4%	72	49	72	32	11	3
	13.7%	44	35	44	21	8	1
	14%	31	23	31	10	7	0
	14.3%	19	17	19	6	6	0
	0.04%	10	9	27	0	0	0
<i>Foodmart</i>	0.045%	6	4	9	0	0	0
	0.05%	2	2	2	0	0	0
	0.055%	2	2	0	0	0	0
	0.06%	0	0	0	0	0	0
	0.3%	42	26	36	0	0	0
<i>Retail</i>	0.4%	29	16	20	0	0	0
	0.5%	26	15	16	0	0	0
	0.6%	21	13	11	0	0	0
	0.7%	14	11	3	0	0	0

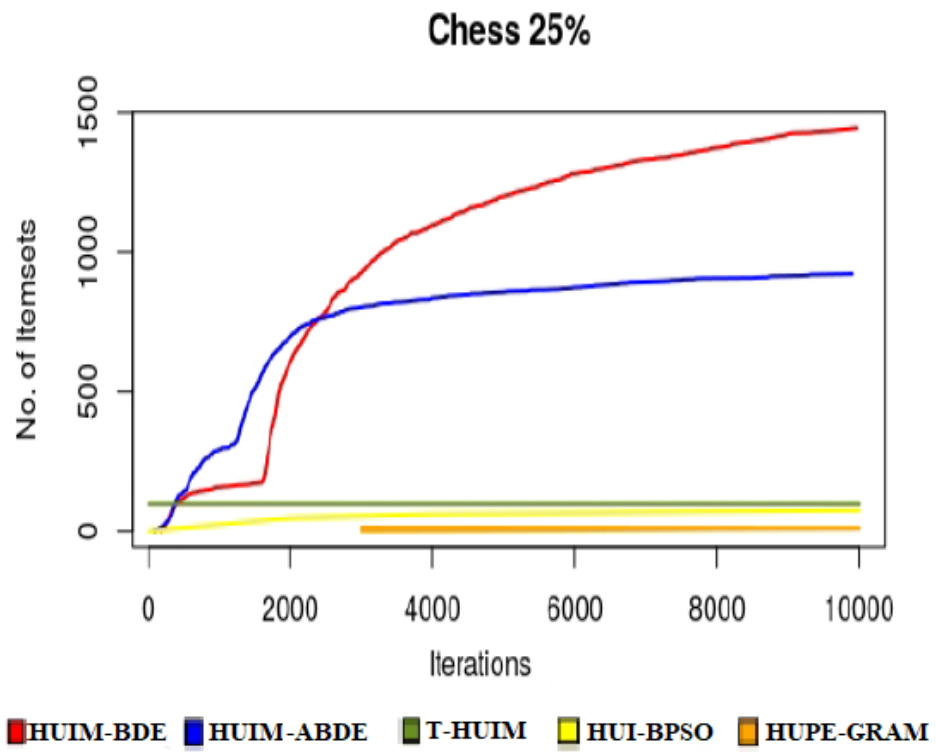


Figure 7.5: Convergence of *Chess* Dataset (HUIM)

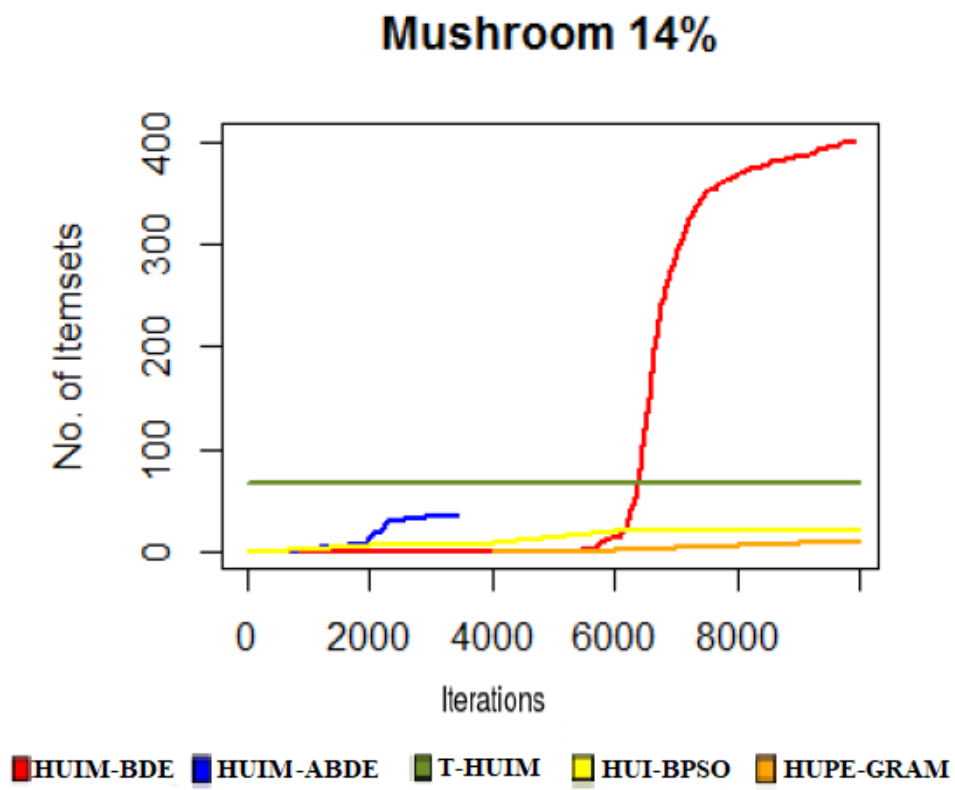


Figure 7.6: Convergence of *Mushroom* Dataset (HUIM)

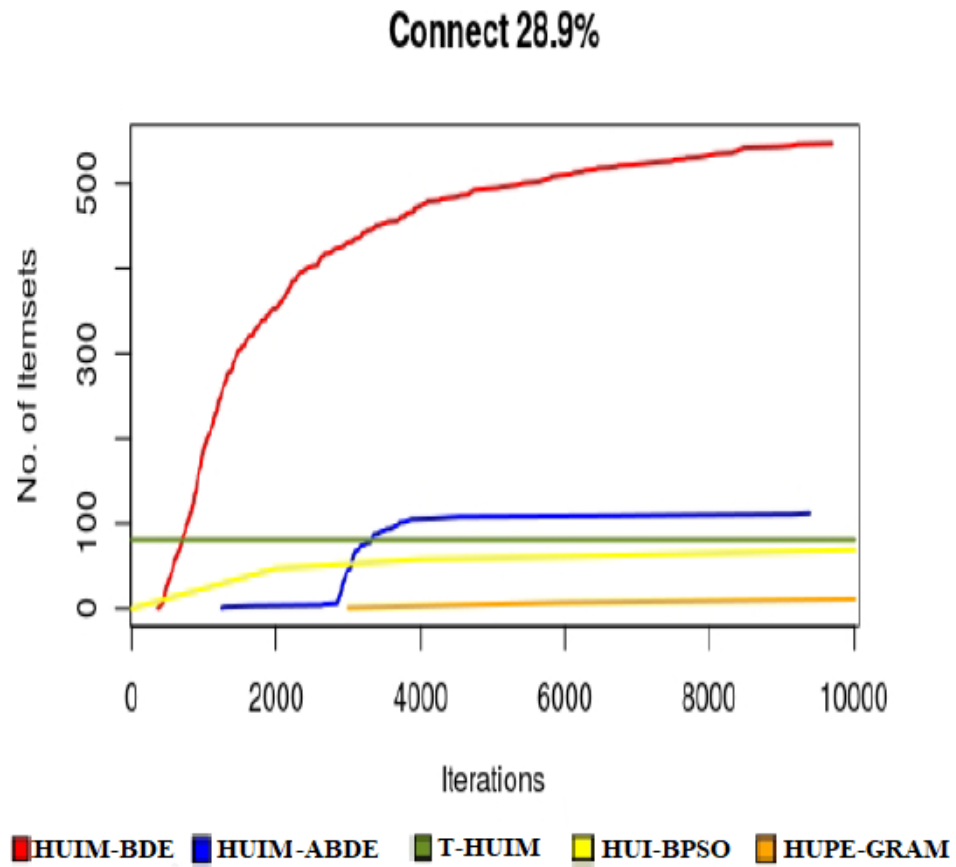


Figure 7.7: Convergence of *Connect* Dataset (HUIM)

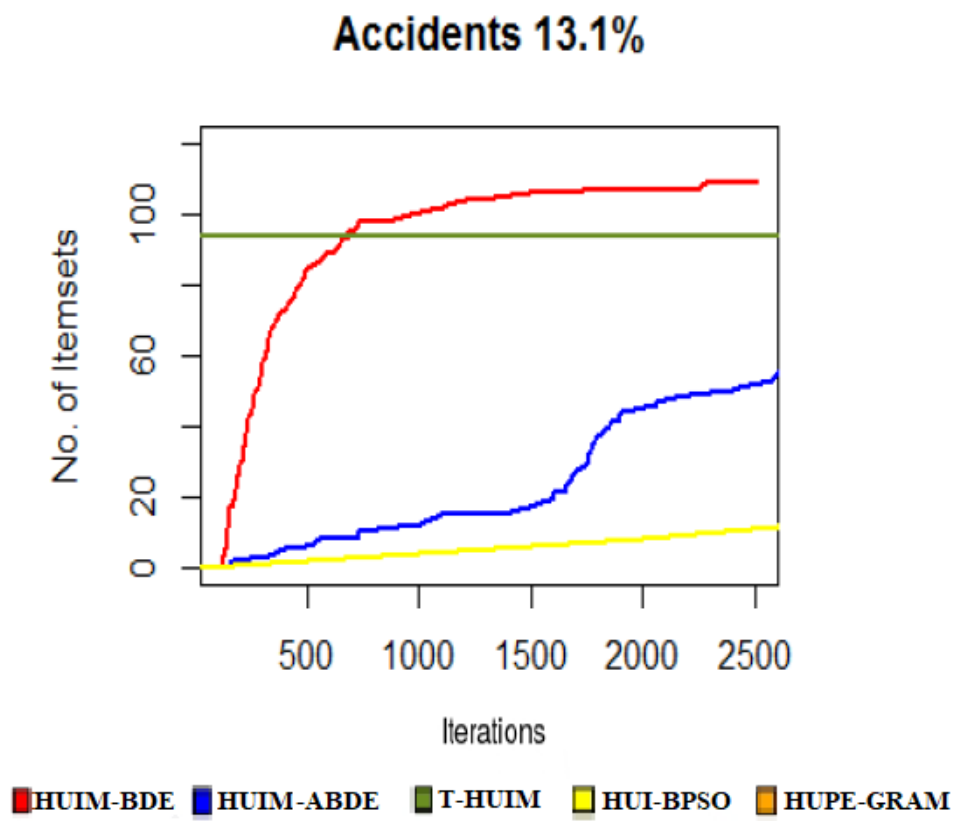


Figure 7.8: Convergence of *Accidents* Dataset (HUIM)

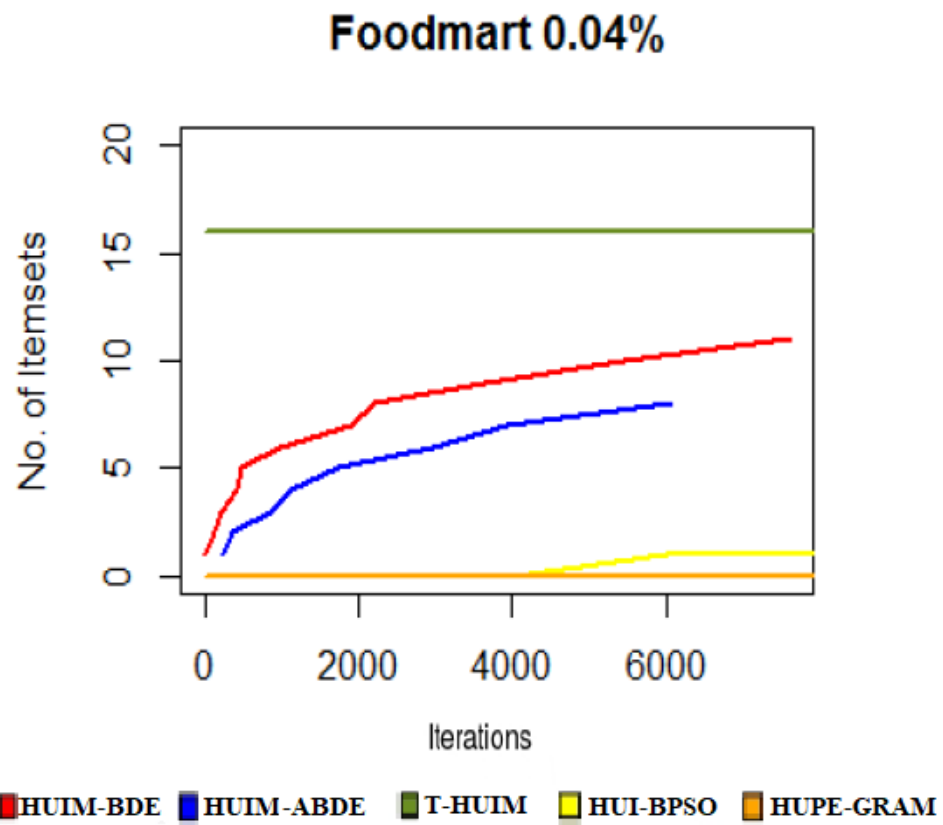


Figure 7.9: Convergence of *Foodmart* Dataset (HUIM)

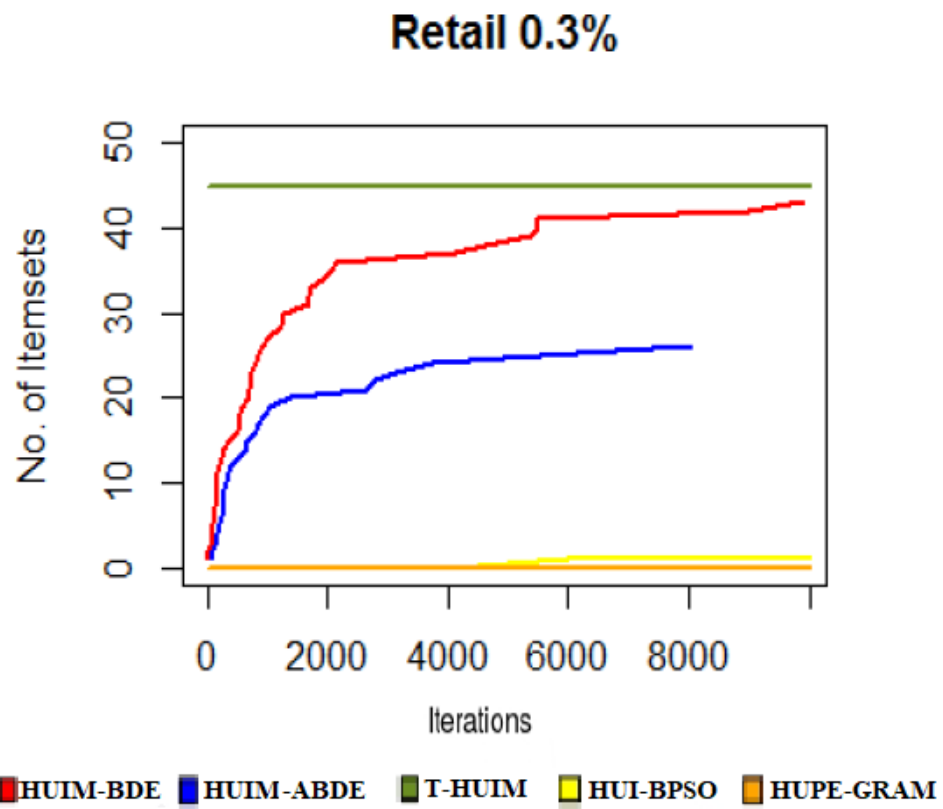


Figure 7.10: Convergence of *Retail* Dataset (HUIM)

7.9.1.3 HUIM-based Customer Segmentation based on Value

The top 5 HUIs from both BDE and ABDE are presented below. As we can see from the findings of the proposed HUIM algorithms that RED_STAR_CARD_HOLDER and ROLL_WRAP_50.S_CHRISTMAS items gave the highest utility of 101122.25 in currency and we can also see that if ROLL_WRAP_50.S _CHRISTMAS is brought then there high chance by the customers likely that they also buy RED_STAR_CARD_HOLDER. Based on this information we can find the customer segments who are of high value to the retail business, and we can target market the same products next time to these customer segments. Figures 7.11 and 7.12 illustrate the itemsets mined and rate of convergence.

Top 5 HUIs mined of the *OnlineRetail* Dataset (i.e. best-run HUIM-BDE)

1. RED_STAR_CARD_HOLDER, ROLL_WRAP_50.S_CHRISTMAS
Utility: 101122.25
2. RED_STAR_CARD_HOLDER, SET_OF_3_COLOURED_FLYING_DUCKS
Utility: 94721.98
3. GREEN_PEONY_CUSHION_COVER, RED_STAR_CARD_HOLDER
Utility: 89766.99
4. PAPERWEIGHT_VINTAGE_COLLAGE, RED_STAR_CARD_HOLDER
Utility: 88671.33
5. PINK_POLKADOT_GARDEN_PARASOL, RED_STAR_CARD_HOLDER
Utility: 86096.89

Top 5 HUIs mined of the *OnlineRetail* Dataset (i.e. best-run HUIM-ABDE)

1. RED_STAR_CARD_HOLDER, ROLL_WRAP_50.S_CHRISTMAS
Utility: 101122.25
2. SET_OF_3_COLOURED_FLYING_DUCKS, RED_STAR_CARD_HOLDER
Utility: 94721.98
3. GREEN_PEONY_CUSHION_COVER, RED_STAR_CARD_HOLDER
Utility: 89766.99

4. JUMBO_BAG_DOLLY_GIRL_DESIGN, RED_STAR_CARD HOLDER

Utility: 62910.47

5. DOLLY_GIRL_CHILDRENS_BOWL, RED_STAR_CARD HOLDER

Utility: 60276.94

Customer IDs of the customers segmented based on products in the best itemset

ROLL_WRAP_50.S_CHRISTMAS, RED_STAR_CARD HOLDER

12347, 12355, 12356, 12359, 12362, 12365, 12395, 12409, 12415, 12417, 12431, 12456,
12502, 12523, 12539, 12540, 12567, 12572, 12578, 12597, 12621, 12640, 12688, 12700,
12721, 12731, 12744, 12748, 12749, 12769, 12795, 12836, 12852, 12868, 12885, 12886,
12921, 12949, 12952, 13004, 13013, 13015, 13050, 13055, 13059, 13081, 13089, 13098,
13134, 13141, 13144, 13209, 13230, 13263, 13267, 13317, 13319, 13324, 13344, 13350,
13365, 13370, 13408, 13468, 13476, 13488, 13496, 13505, 13517, 13591, 13623, 13627,
13731, 13735, 13737, 13759, 13767, 13838, 13842, 13853, 13854, 13869, 13880, 13899,
13932, 13951, 13969, 13983, 13994, 14016, 14021, 14030, 14031, 14051, 14088, 14096,
14105, 14146, 14156, 14194, 14214, 14226, 14239, 14246, 14288, 14289, 14292, 14298,
14299, 14306, 14309, 14323, 14338, 14344, 14359, 14360, 14362, 14383, 14387, 14431,
14439, 14456, 14487, 14535, 14553, 14573, 14587, 14606, 14619, 14621, 14626, 14639,
14640, 14644, 14646, 14649, 14740, 14769, 14770, 14803, 14842, 14849, 14853, 14873,
14895, 14903, 14911, 14952, 14961, 14978, 15004, 15009, 15018, 15031, 15039, 15044,
15065, 15078, 15149, 15150, 15159, 15189, 15194, 15228, 15265, 15281, 15290, 15296,
15304, 15329, 15343, 15345, 15373, 15376, 15382, 15394, 15410, 15416, 15444, 15480,
15493, 15498, 15502, 15508, 15513, 15527, 15547, 15563, 15570, 15601, 15615, 15622,
15646, 15658, 15669, 15674, 15695, 15696, 15701, 15730, 15777, 15796, 15804, 15808,
15812, 15813, 15827, 15852, 15867, 15885, 15906, 15916, 15917, 15944, 15974, 15993,
16012, 16023, 16033, 16045, 16081, 16107, 16128, 16133, 16135, 16168, 16180, 16187,
16201, 16211, 16214, 16240, 16278, 16316, 16366, 16384, 16389, 16497, 16509, 16533,
16552, 16556, 16558, 16560, 16573, 16609, 16618, 16652, 16653, 16670, 16672, 16684,
16709, 16710, 16723, 16764, 16771, 16782, 16791, 16808, 16837, 16839, 16855, 16905,
16906, 16923, 16928, 17040, 17043, 17061, 17064, 17086, 17107, 17115, 17125, 17139,
17147, 17188, 17223, 17315, 17337, 17364, 17365, 17396, 17406, 17410, 17428, 17444,
17453, 17504, 17510, 17648, 17672, 17679, 17706, 17728, 17738, 17789, 17819, 17848,

17865, 17883, 17886, 17905, 17920, 17932, 17949, 18044, 18055, 18081, 18092, 18159, 18192, 18196, 18202, 18223, 18242, 18245, 18257, 18260, 18282

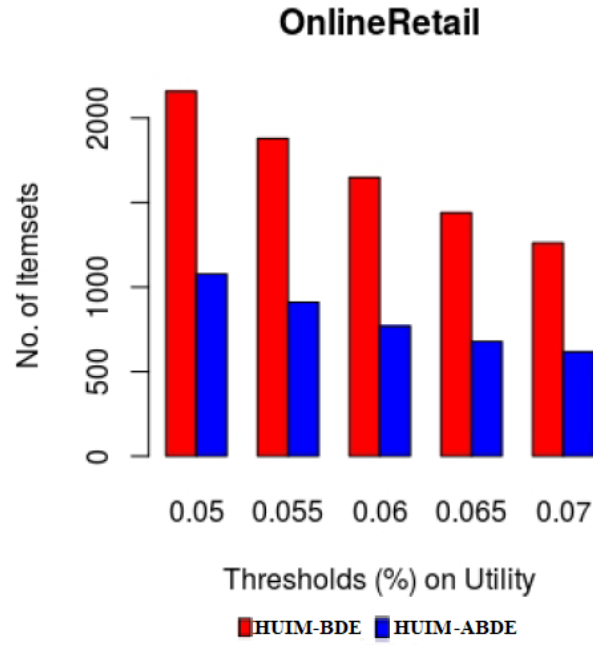


Figure 7.11: No. of Itemsets Mined (*OnlineRetail* Dataset)

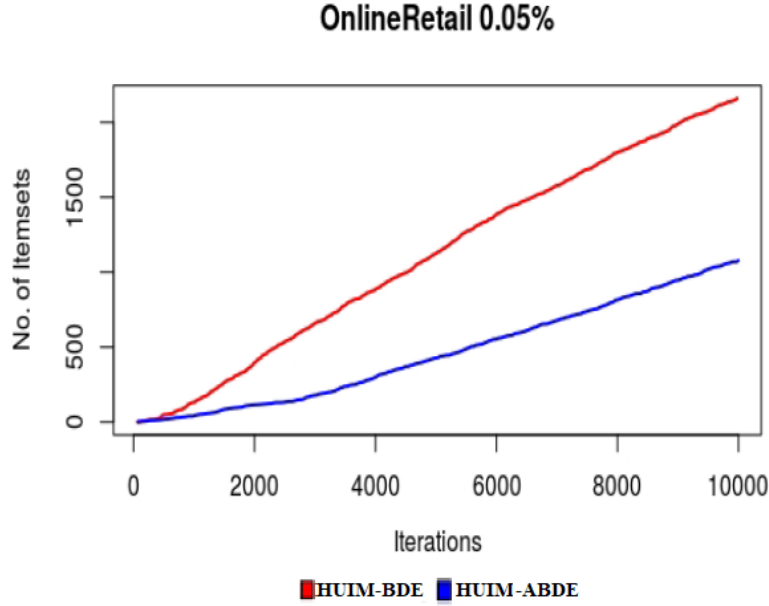


Figure 7.12: Convergence of *OnlineRetail* Dataset (HUIM)

7.9.1.4 Optima Obtained and Statistics by proposed HUIM algorithms

The statistics of the optimum obtained for the proposed models (i.e. BDE and ABDE) of HUIM algorithms for 25 runs are presented in Table 7.7. The optimum obtained after 10000 iterations (i.e. each run) for 25 runs of both the EC techniques (i.e. BDE and ABDE) employed for HUIM is reported with best, median, worst, mean, standard deviation. In the HUIM, the objective function employed is an only utility which is to be maximized. As we can see in Table 7.7, the best optima did not change in case of HUIM mining for both the EC techniques employed and this suggests that both the optimization techniques can converge to the optima in one of the 25 runs. But, comparing other statistics of 25 runs like median, worst, mean and standard deviation, BDE based HUIM is better in convergence to the optimum.

The t-Test is performed on objective function values of 25 runs. BDE performs better than ABDE. Therefore, a pairwise t-test is performed at 1% significance and 48 degrees of freedom with BDE versus ABDE. In the event that the p-value is under 0.01, at this point the null hypothesis is rejected, thereby conveying that the pair of optimizers are not equal. Else, if the p-value is more than 0.01, then the null hypothesis

isn't rejected, and the pair of optimizers are equal. t-Test results are presented in Tables 7.7, with the significance value of 'T', i.e. as true, when the p-value is underneath 0.01 and 'F', i.e. as false, when the p-value is higher than 0.01. Out of the seven p-values from Tables 7.7, it is evident that in five out of seven p-values are indicating that both the optimizers are statistically equal (i.e. null hypothesis is not rejected), and in two cases BDE is found to be statistically not equal to ABDE (i.e. null hypothesis is rejected).

Table 7.7: Statistics of the optimum (utility) values obtained over 25 Runs

Algorithm	Statistics	<i>Accidents</i>	<i>Chess</i>	<i>Connect</i>	<i>Mushroom</i>	<i>Foodmart</i>	<i>Retail</i>	<i>OnlineRetail</i>
HUIM-BDE	Best	3062052	685719	16963313	576483	7188	481021	101122.25
	Median	3062052	685719	16395158	576483	7188	481021	89766.99
	Worst	3062052	596158	14822175	509598	5726	481021	35144.67
	Mean	3062052	672320.96	16140602.72	568792.16	7107.44	481021	83175.86
	Std. Dev.	0.0	24330.92	738926.63	16557.39	297.73	0.0	18897.38
HUIM-ABDE	Best	3062052	685719	16963313	576483	7188	481021	101122.25
	Median	3062052	681000	15521013	528310	7188	481021	67050.84
	Worst	2756735	550284	14744425	480053	5826	481021	35435.71
	Mean	3025379.4	649140.84	15842125.68	525833.68	7034.88	481021	72742.08
	Std. Dev.	71274.72	46827.67	786167.95	28372.57	377.96	0.0	18974.86
t-Value (BDE vs ABDE)		2.572	4.763	1.383	4.411	0.754	0.0	1.948
p-Value (BDE vs ABDE)		0.013	0.0	0.173	0.0	0.454	1.0	0.057
(Significance of T/F)		(F)	(T)	(F)	(T)	(F)	(F)	(F)

7.9.1.5 Memory and Time Consumed by the Proposed HUIM Algorithms

Tables 7.8 and 7.9 give the memory consumed by BDE and ABDE. Considering memory and time, ABDE is performing better than BDE. Memory consumed by ABDE will be less than BDE, as the number of HUIs mined are less in number for ABDE than BDE. But the time consumed by ABDE is less than BDE for the fixed number of iterations, i.e. 10000. Please refer [339] to compare the memory efficiency of the proposed HUI miners with the best HUI miners.

Table 7.8: Memory Consumed (in Mega Bytes) by BDE and ABDE

Dataset	BDE	ABDE	HUI-Miner [339]	FHM [339]
<i>Accidents</i>	93.4	89.4	1,656	1,480
<i>Chess</i>	42.66	39.59	405	305
<i>Connect</i>	146.84	130.2	2,565	3,141
<i>Mushroom</i>	38.77	35.92	194	224
<i>Foodmart</i>	179.6	79.6	808	211
<i>Retail</i>	378.7	373.9	NA	NA
<i>OnlineRetail</i>	518.47	379.92	NA	NA

Table 7.9: Time Consumed (in Seconds) by BDE and ABDE

Dataset	BDE	ABDE
<i>Accidents</i>	952.8	609.23
<i>Chess</i>	1538.22	1347.89
<i>Connect</i>	7284.73	7244.04
<i>Mushroom</i>	1628.03	1035.44
<i>Foodmart</i>	2973.12	1530.9
<i>Retail</i>	3243.48	2027.4
<i>OnlineRetail</i>	41126.43	19206.09

Apart from the timings of individual experiment as given in Table 7.9, we can also see from Figures 7.13 to 7.19 that the time (in seconds) of the proposed HUI miners crossed the utility threshold. As you can see that the proposed HUI miners are indeed able to mine the itemsets very fast as compared to the timing results of [190, 191] and exact methods namely UP-Growth, HUI-Miner in [339] which close to 10000 secs.

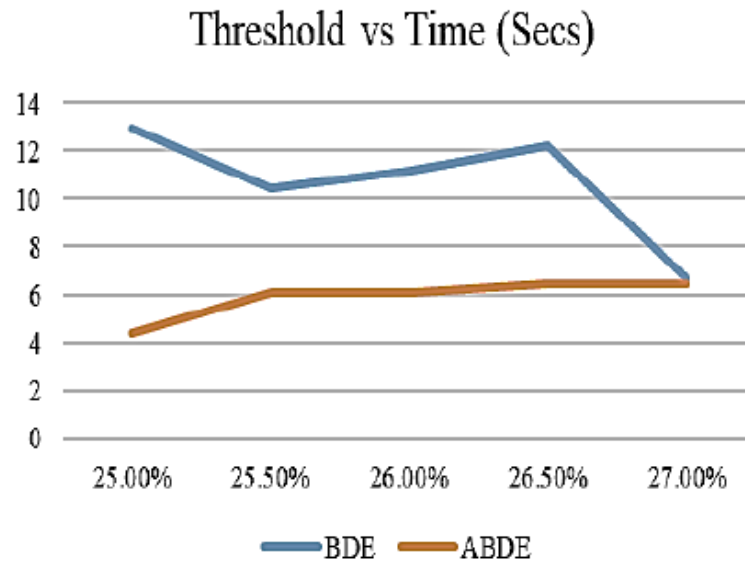


Figure 7.13: Threshold vs Time (in Secs) graph of *Chess* Dataset

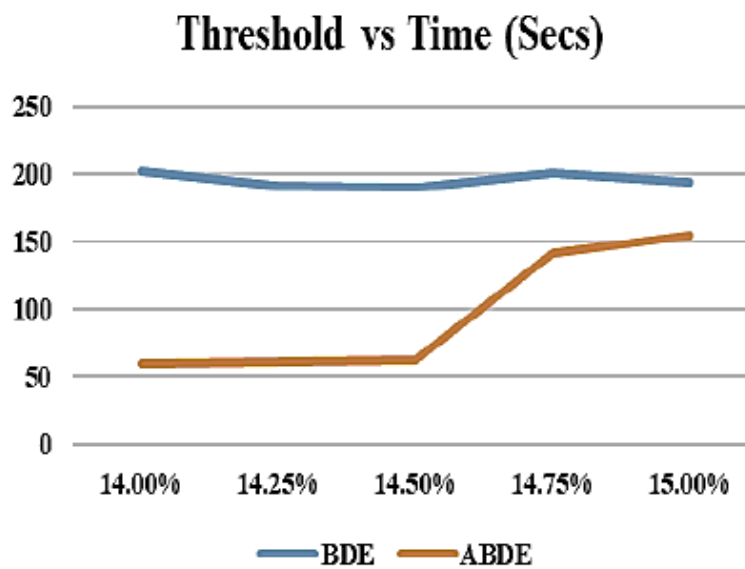


Figure 7.14: Threshold vs Time (in Secs) graph of *Mushroom* Dataset

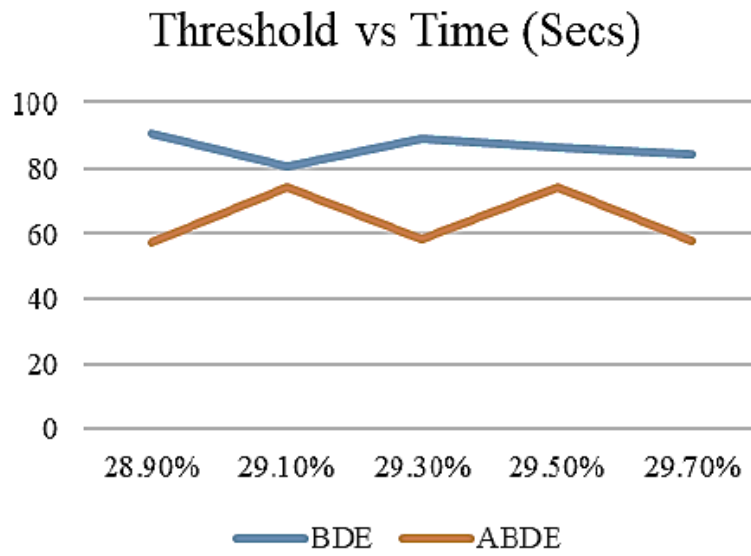


Figure 7.15: Threshold vs Time (in Secs) graph of *Connect* Dataset

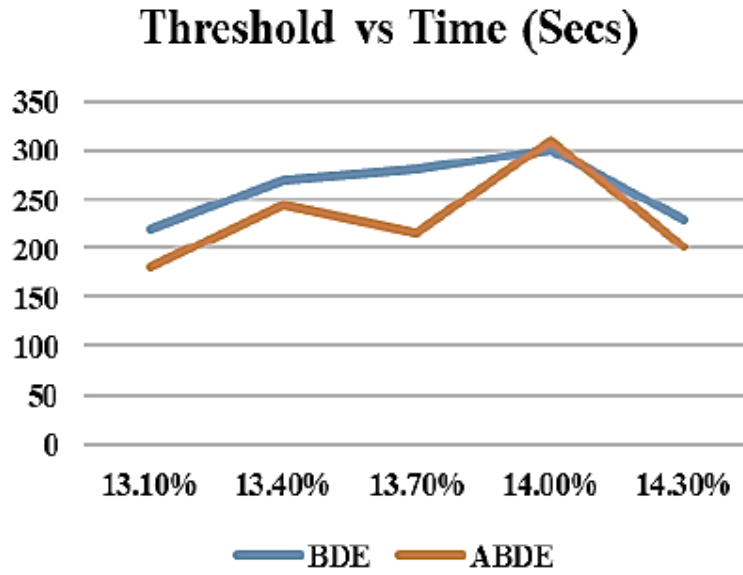


Figure 7.16: Threshold vs Time (in Secs) graph of *Accidents* Dataset

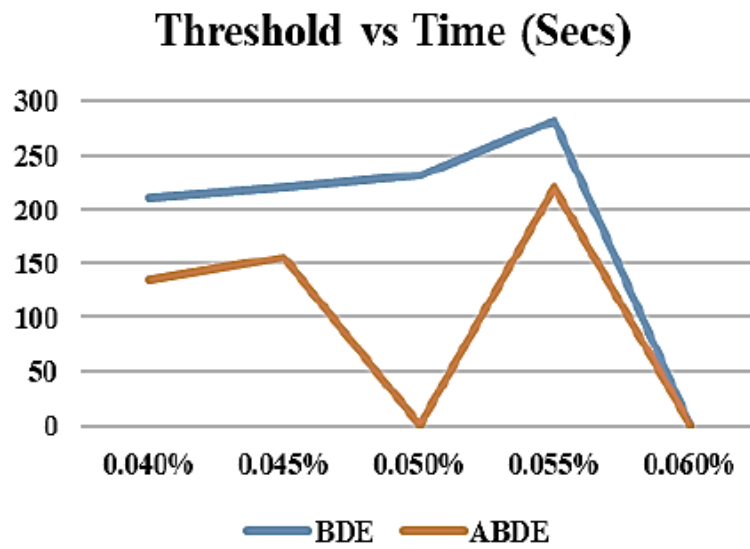


Figure 7.17: Threshold vs Time (in Secs) graph of *Foodmart* Dataset

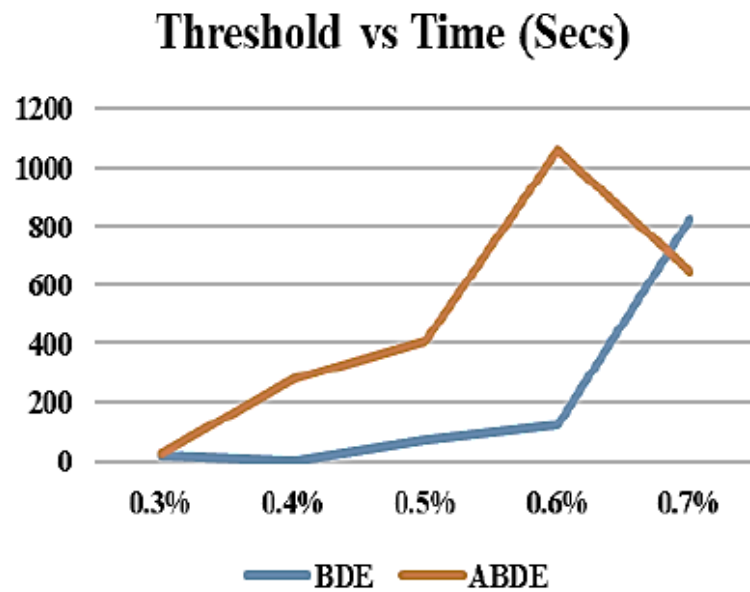


Figure 7.18: Threshold vs Time (in Secs) graph of *Retail* Dataset

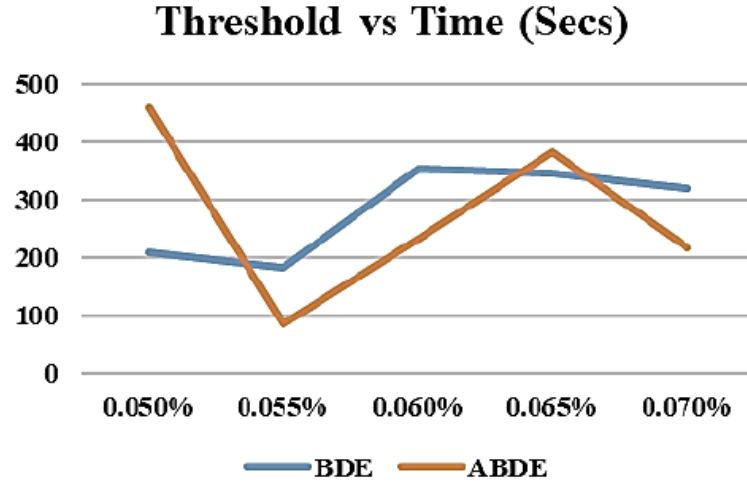


Figure 7.19: Threshold vs Time (in Secs) graph of *OnlineRetail* Dataset

7.9.2 Results of HUARM

7.9.2.1 Number of Association Rules Mined at various Thresholds

Figures 7.20 to 7.25 and Table 7.10 show that the proposed BDE-HUARM could mine an additional number of rules than ABDE-HUARM. Also, the results show that the number of high utility association rules mined did not vary drastically at various thresholds.

Table 7.10: Number of association rules mined at various thresholds

Dataset	Thresholds (Utility*Uconf)					
Chess	Thresholds	25*97.5%	25.5*98%	26*98.5%	26.5*99%	27*99.5%
	BDE	2117	1777	1507	1248	1025
	ABDE	1608	1414	1176	967	701
Mushroom	Thresholds	14*97.5%	14.25*98%	14.5*98.5%	14.75*99%	15*99.5%
	BDE	218	194	161	135	111
	ABDE	88	70	56	44	36
Connect	Thresholds	28.9*97.5%	29.1*98%	29.3*98.5%	29.5*99%	29.7*99.5%
	BDE	2300	2053	1810	1605	1415
	ABDE	1342	1199	1056	929	820
Accidents	Thresholds	13.1*82.5%	13.4*83%	13.7*83.5%	14*84%	14.3*84.5%
	BDE	254	212	168	132	105
	ABDE	21	11	7	2	0
Foodmart	Thresholds	0.04*60%	0.045*62.5%	0.05*65%	0.055*67.5%	0.06*70%
	BDE	2	1	0	0	0
	ABDE	3	1	0	0	0
Retail	Thresholds	0.3*60%	0.4*62.5%	0.5*65%	0.6*67.5%	0.7*70%
	BDE	37	22	17	14	11
	ABDE	35	22	16	12	10
OnlineRetail	Thresholds	0.05*60%	0.055*62.5%	0.06*65%	0.065*67.5%	0.07*70%
	BDE	686	602	528	466	297
	ABDE	272	242	211	186	137

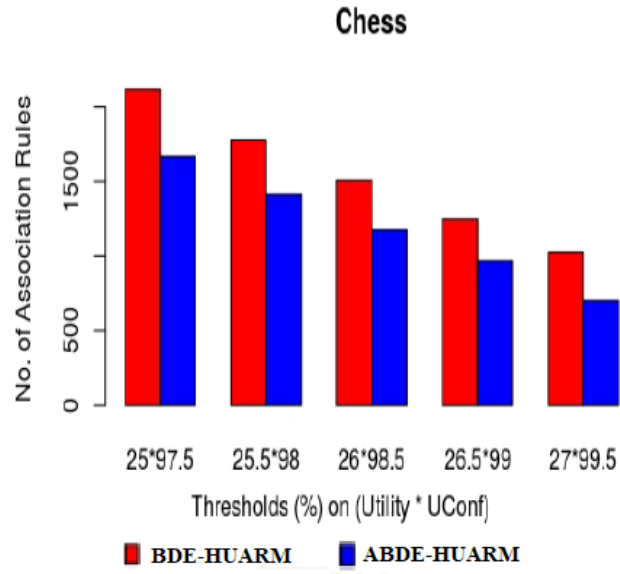


Figure 7.20: No. of Association Rules Mined *Chess* Dataset

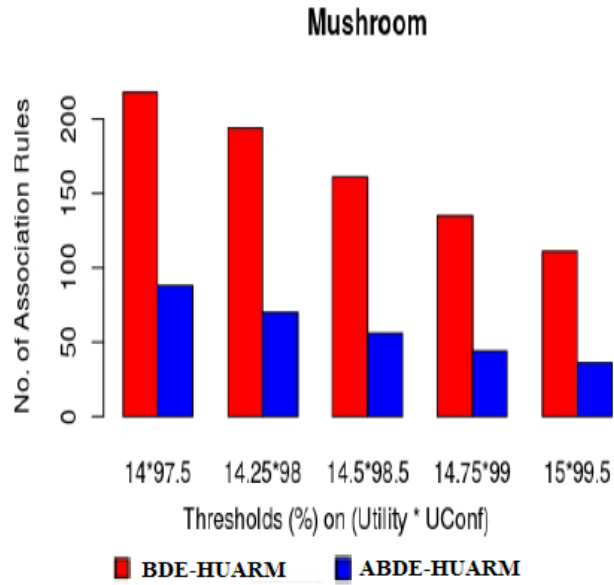


Figure 7.21: No. of Association Rules Mined *Mushroom* Dataset

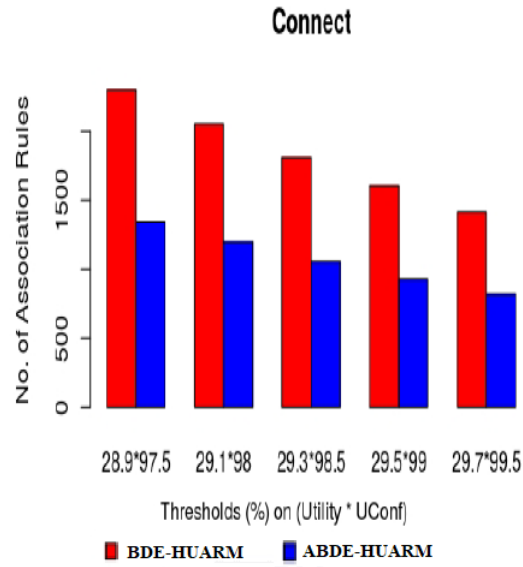


Figure 7.22: No. of Association Rules Mined *Connect* Dataset

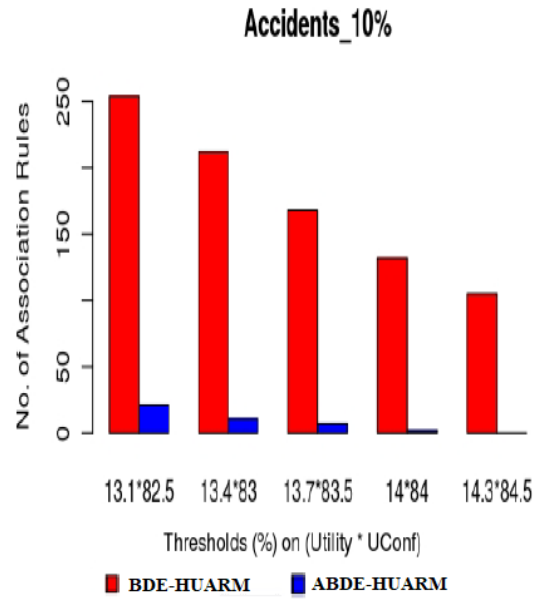


Figure 7.23: No. of Association Rules Mined *Accidents* Dataset

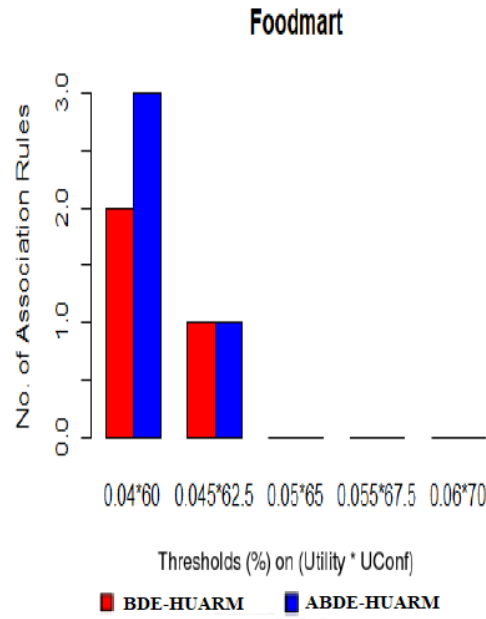


Figure 7.24: No. of Association Rules Mined *Foodmart* Dataset

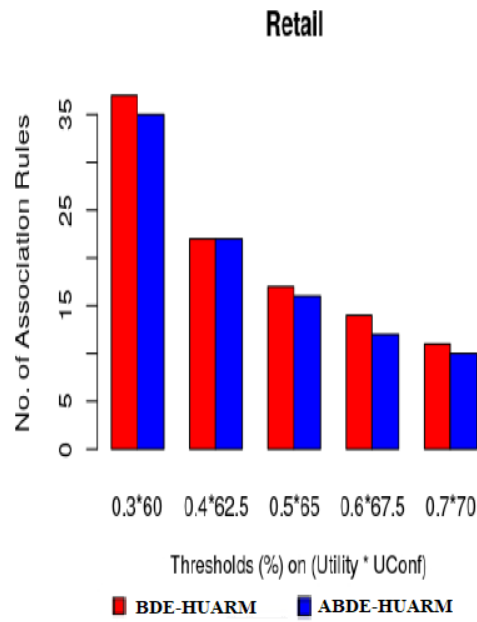


Figure 7.25: No. of Association Rules Mined *Retail* Dataset

7.9.2.2 Convergence of proposed HUARM algorithms

Figures 7.26 to 7.31 illustrate the convergence of the proposed BDE-HUARM at a fixed threshold with ABDE-HUARM. It can be seen from the convergence plots that faster convergence is achieved by BDE-HUARM than ABDE-HUARM on all six datasets.

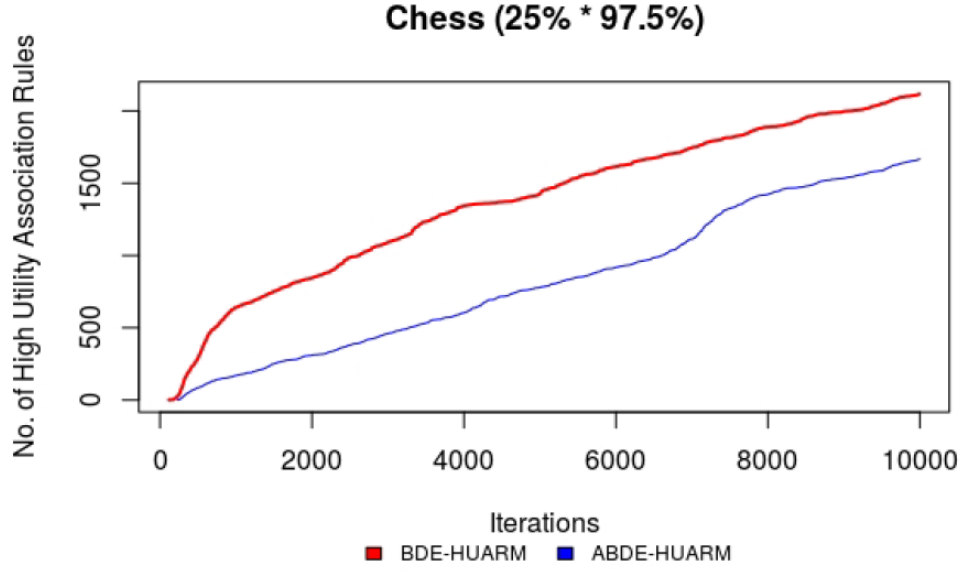


Figure 7.26: Convergence of *Chess* Dataset (HUARM)

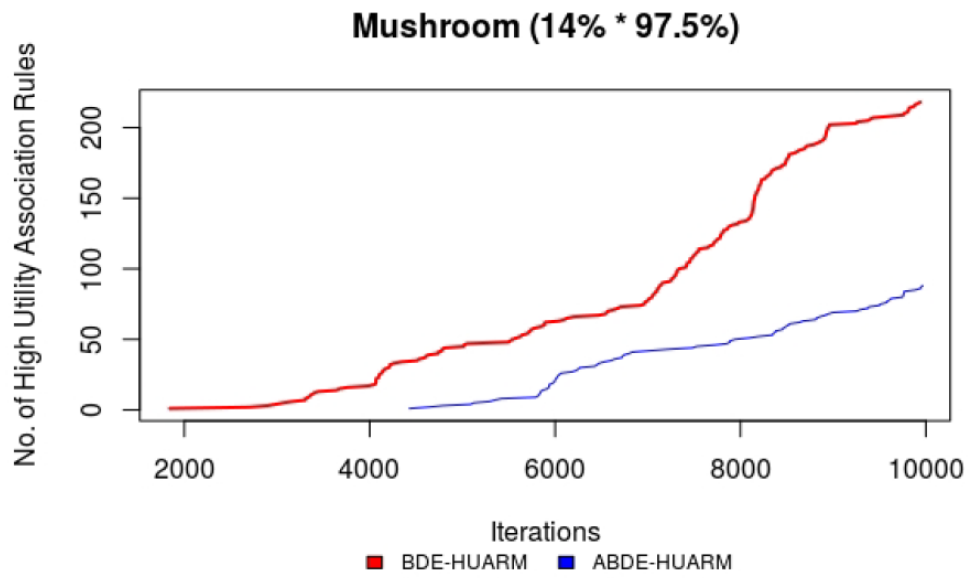


Figure 7.27: Convergence of *Mushroom* Dataset (HUARM)

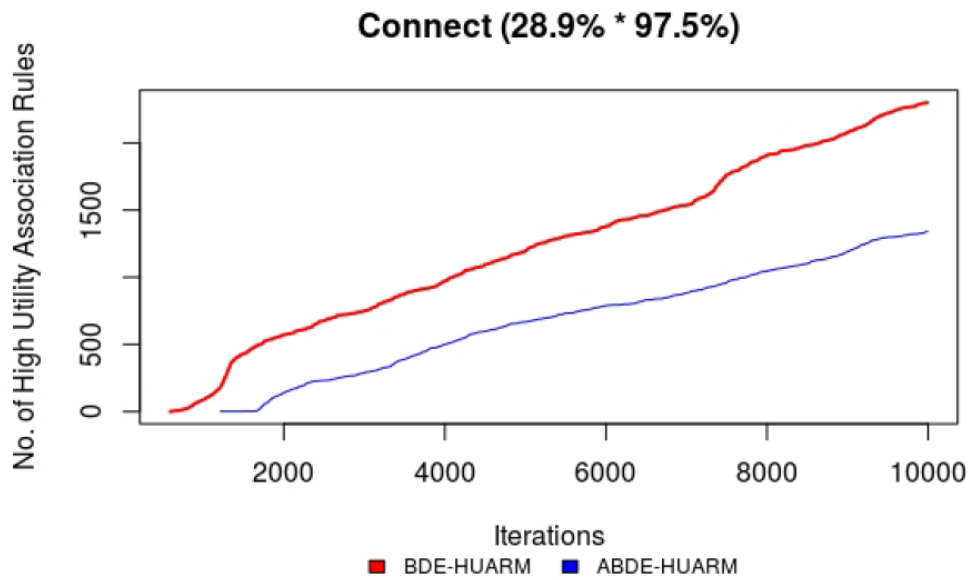


Figure 7.28: Convergence of *Connect* Dataset (HUARM)

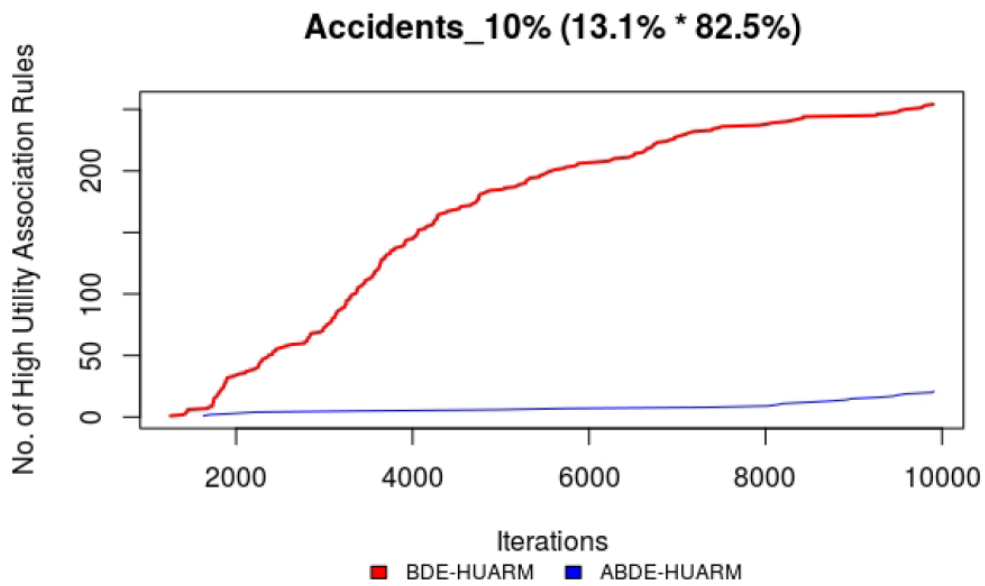


Figure 7.29: Convergence of *Accidents* Dataset (HUARM)

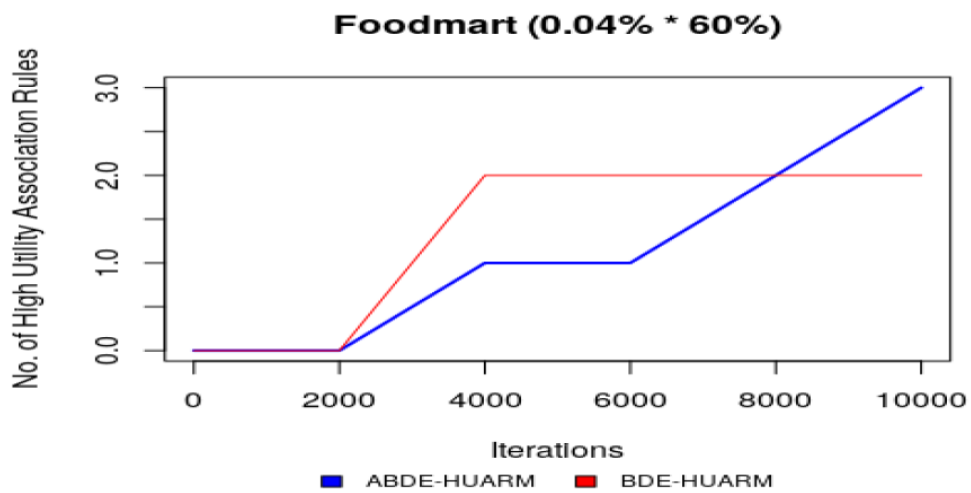


Figure 7.30: Convergence of *Foodmart* Dataset (HUARM)

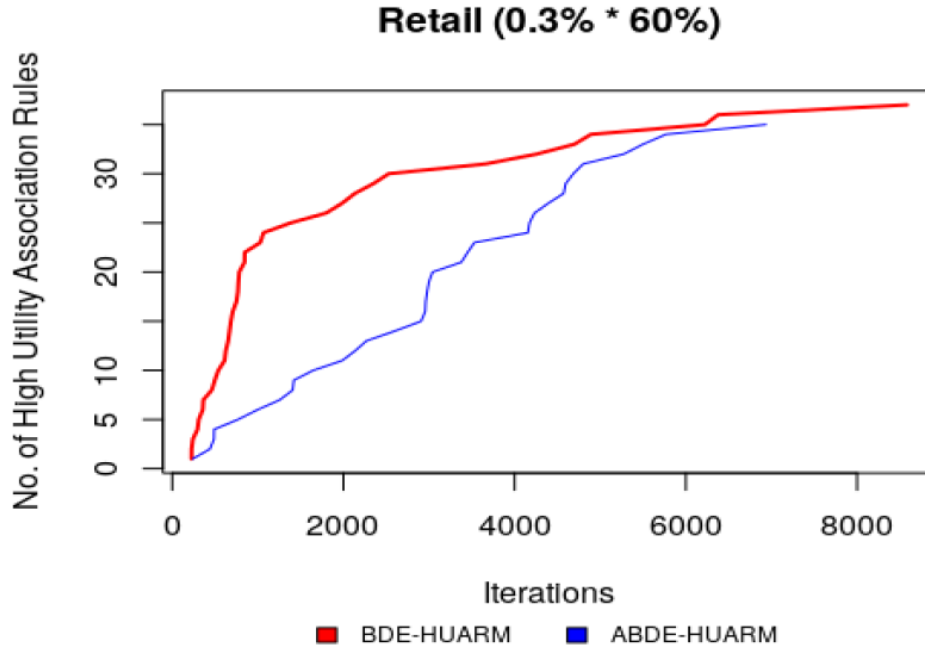


Figure 7.31: Convergence of *Retail* Dataset (HUARM)

7.9.2.3 HUARM-based Customer Segmentation based on Value

The top 5 high utility association rules for both BDE and ABDE are presented below. As we can see from the findings of the proposed HUARM algorithms that RED_STAR_CARD HOLDER and ROLL_WRAP_50.S_CHRISTMAS items gave the highest utility of 101122.25 in currency and we can also see that if ROLL_WRAP_50.S_CHRISTMAS is brought then there high chance by the customers likely that they also buy RED_STAR_CARD HOLDER. Based on this information we can find the customer segments who are of high value to the retail business, and we can target market the same products next time to these customer segments.

Top 5 High Utility Association Rules mined of the OnlineRetail Dataset (i.e. best-run BDE-HUARM)

1. ROLL_WRAP_50.S_CHRISTMAS $\rightarrow$ RED_STAR_CARD HOLDER
Utility: 101122.25, UConfidence: 88.99, Product: 8999358.07
2. PINK_POLKADOT_GARDEN_PARASOL $\rightarrow$ RED_STAR_CARD HOLDER,

ROLL_WRAP_50.S_CHRISTMAS

Utility: 103913.74, UConfidence: 86.58, Product: 8997214.34

3. RED_WOOLLY_HOTTIE_WHITE_HEART. → RED_STAR_CARD_HOLDER

Utility: 90815.22, UConfidence: 94.38, Product: 8571837.35

4. ROLL_WRAP_50.S_CHRISTMAS → PINK_POLKADOT_GARDEN_PARASOL,
RED_STAR_CARD_HOLDER

Utility: 103913.74, UConfidence: 79.45, Product: 8256983.37

5. PINK_POLKADOT_GARDEN_PARASOL → RED_STAR_CARD_HOLDER

Utility: 86096.89, UConfidence: 90.51, Product: 7792906.17

**Top 5 High Utility Association Rules mined of the OnlineRetail Dataset
(i.e. best-run ABDE-HUARM)**

1. ROLL_WRAP_50.S_CHRISTMAS → RED_STAR_CARD_HOLDER

Utility: 101122.25, UConfidence: 88.99, Product: 8999358.07

2. SET_OF_3_COLOURED_FLYING_DUCKS → RED_STAR_CARD_HOLDER

Utility: 94721.98, UConfidence: 74.69, Product: 7074877.04

3. RED_VINTAGE_SPOT_BEAKER → RED_STAR_CARD_HOLDER

Utility: 72231.19, UConfidence: 93.36, Product: 6743888.19

4. RED_STAR_CARD_HOLDER → ROLL_WRAP_50.S_CHRISTMAS

Utility: 101122.25, UConfidence: 55.39, Product: 5601935.48

5. RED_STAR_CARD_HOLDER → ASS_COL_SMALL_SAND_GECKO_P.WEIGHT

Utility: 80976.78, UConfidence: 38.99, Product: 3157432.80

Customer IDs of the customers segmented based on products in the best association rule i.e.

ROLL_WRAP_50.S_CHRISTMAS → RED_STAR_CARD_HOLDER

12347, 12355, 12356, 12359, 12362, 12365, 12395, 12409, 12415, 12417, 12431, 12456,
12502, 12523, 12539, 12540, 12567, 12572, 12578, 12597, 12621, 12640, 12688, 12700,
12721, 12731, 12744, 12748, 12749, 12769, 12795, 12836, 12852, 12868, 12885, 12886,
12921, 12949, 12952, 13004, 13013, 13015, 13050, 13055, 13059, 13081, 13089, 13098,

13134, 13141, 13144, 13209, 13230, 13263, 13267, 13317, 13319, 13324, 13344, 13350, 13365, 13370, 13408, 13468, 13476, 13488, 13496, 13505, 13517, 13591, 13623, 13627, 13731, 13735, 13737, 13759, 13767, 13838, 13842, 13853, 13854, 13869, 13880, 13899, 13932, 13951, 13969, 13983, 13994, 14016, 14021, 14030, 14031, 14051, 14088, 14096, 14105, 14146, 14156, 14194, 14214, 14226, 14239, 14246, 14288, 14289, 14292, 14298, 14299, 14306, 14309, 14323, 14338, 14344, 14359, 14360, 14362, 14383, 14387, 14431, 14439, 14456, 14487, 14535, 14553, 14573, 14587, 14606, 14619, 14621, 14626, 14639, 14640, 14644, 14646, 14649, 14740, 14769, 14770, 14803, 14842, 14849, 14853, 14873, 14895, 14903, 14911, 14952, 14961, 14978, 15004, 15009, 15018, 15031, 15039, 15044, 15065, 15078, 15149, 15150, 15159, 15189, 15194, 15228, 15265, 15281, 15290, 15296, 15304, 15329, 15343, 15345, 15373, 15376, 15382, 15394, 15410, 15416, 15444, 15480, 15493, 15498, 15502, 15508, 15513, 15527, 15547, 15563, 15570, 15601, 15615, 15622, 15646, 15658, 15669, 15674, 15695, 15696, 15701, 15730, 15777, 15796, 15804, 15808, 15812, 15813, 15827, 15852, 15867, 15885, 15906, 15916, 15917, 15944, 15974, 15993, 16012, 16023, 16033, 16045, 16081, 16107, 16128, 16133, 16135, 16168, 16180, 16187, 16201, 16211, 16214, 16240, 16278, 16316, 16366, 16384, 16389, 16497, 16509, 16533, 16552, 16556, 16558, 16560, 16573, 16609, 16618, 16652, 16653, 16670, 16672, 16684, 16709, 16710, 16723, 16764, 16771, 16782, 16791, 16808, 16837, 16839, 16855, 16905, 16906, 16923, 16928, 17040, 17043, 17061, 17064, 17086, 17107, 17115, 17125, 17139, 17147, 17188, 17223, 17315, 17337, 17364, 17365, 17396, 17406, 17410, 17428, 17444, 17453, 17504, 17510, 17648, 17672, 17679, 17706, 17728, 17738, 17789, 17819, 17848, 17865, 17883, 17886, 17905, 17920, 17932, 17949, 18044, 18055, 18081, 18092, 18159, 18192, 18196, 18202, 18223, 18242, 18245, 18257, 18260, 18282

The number of association rules mined at various thresholds for the OnlineRetail dataset for the proposed HUARM algorithms is illustrated in Figure 7.32. The convergence of the BDE and ABDE based HUARM is illustrated in Figure 7.33.

7.9.2.4 Optima Obtained and Statistics of proposed HUARM algorithms

The statistics of the optimum obtained for the proposed models (i.e. BDE and ABDE) of HUARM algorithms for 25 runs is presented in Table 7.11. The optimum obtained after 10000 iterations (i.e. each run) for 25 runs of both the EC techniques (i.e. BDE and ABDE) employed for high utility association rule mining is reported with best, median, worst, mean, standard deviation. In the case of high utility association rule

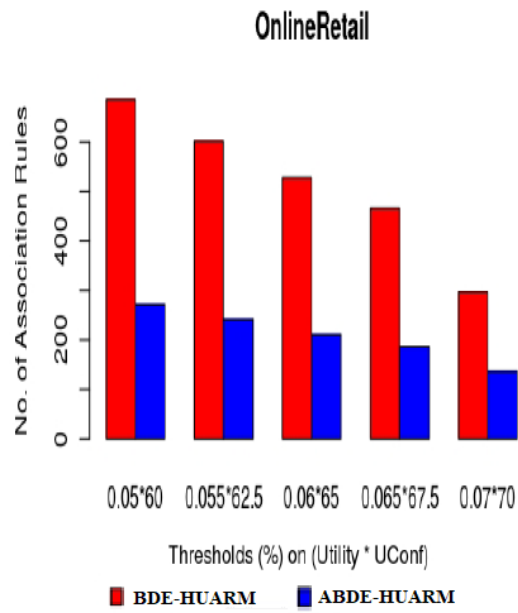


Figure 7.32: No. of Association Rules Mined (*OnlineRetail* Dataset)

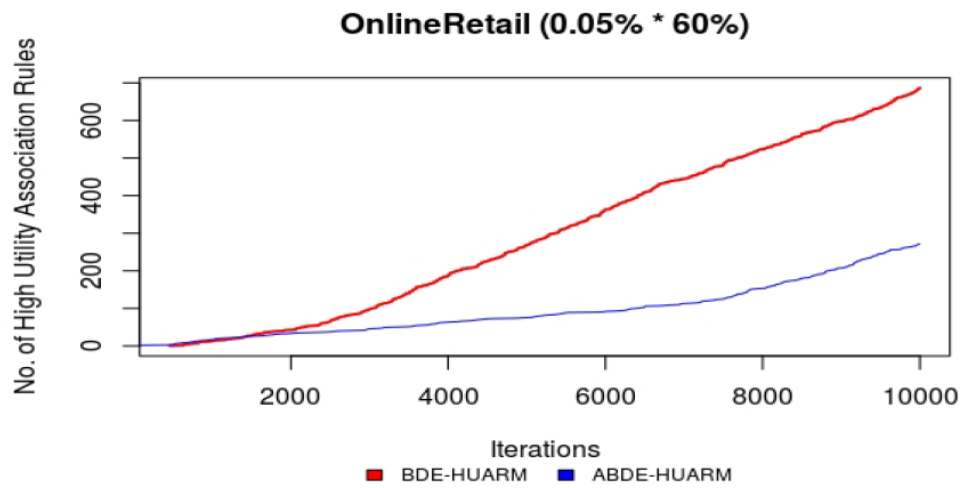


Figure 7.33: Convergence of *OnlineRetail* Dataset (HUARM)

mining, the objective function is the product of both utility and uconfidence measures, which is to be maximized. But, comparing other statistics of 25 runs like median, worst, mean and standard deviation, BDE based HUARM is better in convergence to the optimum.

The t-Test is performed on objective function values of 25 runs. BDE performs better than ABDE. Therefore, a pairwise t-test is performed at 1% significance and 48 degrees of freedom with BDE versus ABDE. In the event that the p-value is under 0.01, at this point the null hypothesis is rejected, thereby conveying that the pair of optimizers are not equal. Else, if the p-value is more than 0.01, then the null hypothesis isn't rejected, and the pair of optimizers are equal. t-Test results are presented in Tables 7.11, with the significance value of 'T', i.e. as true, when the p-value is underneath 0.01 and 'F', i.e. as false, when the p-value is higher than 0.01. Out of the seven p-values from Tables 7.11, it is evident that in six out of seven p-values are indicating that both the optimizers are statistically equal (i.e. null hypothesis is not rejected), and in only one cases BDE is found to be statistically not equal to ABDE (i.e. null hypothesis is rejected).

Table 7.11: Statistics of the optimum (Utility * Uconf) obtained for 25 Runs

Algorithm	Statistics	Accidents	Chess	Connect	Mushroom	Foodmart	Retail	OnlineRetail
BDE-HUARM	Best	306074543	68502016.9	1691816100	57648300	835800	33310574.9	9231202.16
	Median	305173953	68365547.8	1688952238	57398359.1	443000	33310574.9	6380328.13
	Worst	261710995	67704721.8	1688909630	56592449.8	295400	33310574.9	1979530.16
	Mean	297853752	68221044.6	1689065109	57365798.3	447584.9	33310574.9	6280939.81
ABDE-HUARM	Std. Dev.	13740786.9	198819.13	573240.77	233633.83	124563.9	0.0	2541600.87
	Best	293394250	68502016.9	1688964638	57412750.3	680600	33310574.9	8999358.07
	Median	264392826	68095269.9	1688935603	56575304.1	394600	33310574.9	3214705.48
	Worst	234848424	67982050.3	1601365061	51923500	322800	14521023.8	1607948.18
	Man	261615366	68199540.9	1674189035	56287588.3	416276	32558992.9	4250709.26
	Std. Dev.	19032640.7	170025.83	26107991.45	1361323.64	88948.3	3757910.2	2548955.75
t-Value (BDE vs ABDE)		4.881	0.259	1.801	2.468	0.646	0.632	1.783
p-Value (BDE vs ABDE) (Significance of T/F)		0.000 (T)	0.796 (F)	0.077 (F)	0.017 (F)	0.520 (F)	0.530 (F)	0.080 (F)

7.9.2.5 Memory and Time Consumed by the Proposed HUARM Algorithms

Tables 7.12 and 7.13 give the memory consumed by BDE and ABDE. Considering memory and time, ABDE is performing better than BDE. Memory consumed by ABDE will be less than BDE, as the number of high utility association rules mined are less in number for ABDE than BDE. But the time consumed by ABDE is less than BDE for the fixed number of iterations, i.e. 10000. These models were executed in single threaded R tool. Therefore, if executed on multi-threaded environment they will be 10x to 15x time faster. This is also another reason why we did not compare the execution times of the proposed methods with two heuristic methods in [206, 261].

Table 7.12: Memory Consumed (in Mega Bytes) by BDE-HUARM and ABDE-HUARM

Dataset	BDE	ABDE
Accidents	163.15	151.5
Chess	42.66	39.59
Connect	146.84	130.2
Mushroom	38.77	35.92
Foodmart	39.74	98.45
Retail	256.33	405.78
OnlineRetail	518.47	379.92

Table 7.13: Time Consumed (in Seconds) by BDE-HUARM and ABDE-HUARM

Dataset	BDE	ABDE
Accidents	5085.4	2289.53
Chess	1538.22	1347.89
Connect	7284.73	7244.04
Mushroom	1628.03	1035.44
Foodmart	3527.8	3436.6
Retail	4470.4	4107.5
OnlineRetail	41126.43	19206.09

7.9.3 Control Parameters of the BDE and ABDE

There are two parameters for BDE and ABDE. One is Crossover Rate (CR), and another is Differential Weight (F). CR and F are tuned using a grid search for BDE. But, for ABDE these two are adaptive. Another set of parameters for BDE and ABDE

are iterations and population size which are fixed at 10000 and 20 for HUIM to compare with extant methods. For HUARM also iterations and population size which are fixed at 10000 and 20. Parameter values taken by the BDE are between 0.35-0.75 for CR and between 0.50-1.35 for F.

7.10 Conclusions and Future Directions

We developed high utility itemset and association miners with Binary Differential Evolution and Adaptive Binary Differential Evolution which is efficient in terms of the number of itemsets or rules generated at various thresholds and convergence rate. This work has also shown the power and versatility of the BDE. Additionally, ABDE and BDE-based HUIM miners turned out to be statistically equal (i.e. null hypothesis is not rejected) in five out of seven comparisons, as demonstrated through a t-test. ABDE and BDE-based HUARM miners turned out to be statistically equal (i.e. null hypothesis is not rejected) in six out of seven comparisons, as demonstrated through a t-test. But, considering the time consumed in tuning control parameters of BDE as well as mining high utility itemsets and rules, ABDE is the winner. For ABDE, there is no need to tune the control parameters manually. We explored a possible application of the HUIM and HUARM for the business needs, i.e. customer segmentation based on monetary value of the customers. One possible advantage of this study applied to the retail sector is to allocate and manage the retail shelf-space efficiently as a result of which high ROI for the businesses can be realized. Another advantage of this study applied to the retail sector is to analyze the transactional databases of the businesses, to identify which possible items offer high revenue if sold together. In future, better EC techniques in both single and multi-objective frameworks can also be employed to solve this problem.

Chapter 8

Subspace based Anomaly Detection via Harmony Search and Differential Evolution: An application to Analytical CRM and Security

8.1 Introduction

Anomalies are the samples, items or events that are not distributed according to the expected pattern or do not conform to other samples or patterns present in a dataset. These anomalies may also indicate a significant threat such as cyber-crimes. Anomalies can be divided into the samples caused due to error or the samples with irregular distribution. In some cases, data can be non-normal or may appear as a combination of distributions, which make the anomaly detection very difficult.

As the dimensionality increases, this makes the data samples look sparser and also appear equidistant. This medium or high dimensionality makes anomaly detection hard for conventional anomaly detection methods. To improve the efficiency of anomaly detection, some authors propose subspace-based anomaly detection methods which detect the anomalies by making use of sparsity coefficient, which was proposed and analyzed in [13]. Therefore, it is selected as the objective function for the optimization

algorithms and the samples lying in the sparse subspaces are termed as anomalies.

Some used a Genetic Algorithm [110], to search for the sparse subspaces in the exponential number of subspaces [13]. In [74], optimization methods that are more stable and effective than GA can be found. This study proposes population-based EC methods for anomaly detection. These are suitable for detecting anomalies in security, insurance and banking data. This study also introduced multiplying *precision* by *recall*, which is a new measure, as the selection constraint, to evaluate the detected anomalies as part of the selection process of MDE.

8.2 Related Work

In [70], GA is employed to divide the data into subsets of multi-case anomalies. This approach relies on mean squared error for multi-case anomaly detection. In [303], GA is utilized not only for anomaly detection, but it is also utilized in linear regression model for feature detection. This feature detection is mostly carried out by finding different clusters of anomalies and non-anomalies. In another study [215], particle swarm optimization technique, commonly known as PSO was suggested for detecting anomalies. This technique is compared with the commonly used Local Outlier factor (LOF) technique. Instead of using the manual separation measures, PSO rather finds the separation measures dynamically. In [212], PSO was used to identify the smallest number of non-reducts on the rough sets. Another study [197] proposed a hybrid method to find the anomalies, which combined the PSO technique with self-organizing maps (SOM) technique. This hybrid method improved the efficiency and gave better results than the PSO or SOM model alone. In a later study [121], another anomaly detection method was proposed based on local search of categorical data; and the objective function utilized is entropy. Also, the results of this study were compared to those of K-Nearest Neighbor and LOF. In [180], a non-parametric kernel density method is utilized to detect outliers. This was compared with Local Correlation Integral (LOCI) and LOF.

Ref. [341] compares different numerical algorithms used for anomaly detection and evaluates all these techniques thoroughly. In [341], different variables such as identifying relevant attributes, hubness and combinatorial explosion, are described as problems

when dealing with high-dimension anomalies. In [53] evaluates various detection techniques but extends the study to find the applications of these different techniques for various applications.

8.2.1 Keystroke-based User Authentication

User Authentication is one of the core aspects to be considered for maintaining security. For authentication, data and information of each user must be stored. Normally, the user authentication process is performed in three methods. In the first method, a user is authenticated based on “something he/she knows”, for example PIN or password-based authentication. In the second method users are authenticated based on “something he/she has”, such as smart cards. The third approach is based on “something he/she is”, which includes biometric verification. Authentication processes based on passwords does not consider user-centric data; due to which it can be easily compromised. However, fingerprint-based authentication method or keystroke dynamic authentication models consider user specific information and thus they are more reliable and secure. As in recent years, there has been an increasing trend in security, where the machine learning methods based on keystroke dynamics are preferred over conventional methods. So, the keystroke-based authentication is continuously gaining attention of the security and artificial intelligence (AI) researchers. Gaines et al. [247] did a seminal work based on keystroke dynamics. This study gained considerable attention and from this point onwards, the research in this field increased significantly. The deployment and implementation expenses of these keystroke dynamics methods are low which make them more attractive. [247] also shows that there are some differences in the typing trends, among different experts. Therefore, it is quite challenging to repeat a person’s keystroke pattern with significant accuracy; this means that the keystroke pattern of each person is identical to their specific signature [247]. Keystroke dynamics uses argons such as, user verification, user recognition, user identification, or user authentication and user authorization. Authentication is the process of recognizing the user as the identity he/she claims. In this system, the authentication process is aimed at enabling just one user, who is the actual holder of an account to be recognized through the passwords/smart cards/biometric, and thereby classifying others who are not authenticated as imposters. This study marks the cases of the authentic user as the genuine category and the cases thar are not authentic, as imposter category. The

latter are in significantly larger numbers. For this reason, it is recommended to use one class classification in order to learn the keystroke pattern of the authentic user only. *But, not in all situations labels will be available to do one-class classification. Therefore, clustering-based anomaly detection techniques are mostly preferred over one-class classification techniques.*

Different statistical methods, including standard deviation, mean, and median are utilized for differentiate the timing vector by way of a statistical t-test [213]. A research published in 1980 [247], was one of the initial studies which focused on the keystroke time. As explained by various experts [152, 207], it is possible to employ keystroke timing vectors in sensing into which category, genuine or imposter, the observation falls. Some researchers in [207] try to simulate the keyboard-based proximity detection method for the keystroke recognition with the MLP (multi-layer perceptron) neural connections and the k-nearest neighbor. Balagani et al. [26] employed a distance-oriented approach based on a specific distance measure named as Bhattacharya distance for the authentication and verification of fixed text keystroke. In contrast, Joyce and Gupta [143] developed a technique based on keystroke latencies. Test signatures are compared with these keystroke latencies to detect user identity and profile. [106] gives a detailed evaluation of the existing keystroke datasets. In this research, the public dataset that Killourhy and Maxion [152] developed, was utilized. During the research, it was observed that clustering-based anomaly detection is a fertile area and no significant work is reported in this area.

[152], presents 11 unique techniques of anomaly detection, which are used to determine and reject imposters. Moreover, three fundamental distance-based anomaly detection approaches – Manhattan, Euclidian, and Mahalanobis, were also proposed in this study. [125] presents a comparison of different anomaly types in data. In this study, three major types of anomalies in data are discussed. It also divides the detection methods into three major groups i.e. those which are used when normality and abnormality both are present; those which are used when no prior knowledge of data is available and those which are used when there are only few samples of abnormalities are present. [204] designed DeepSecure, a neural network architecture, which produced an Equal Error Rate (EER) of 0.03. [204] considers DeepSecure to be the best of all models for keystroke user authentication.

8.2.2 Clustering-based Anomaly detection for CRM

The banking and insurance data utilized in the study are of credit card default, banking customer churn, and automobile insurance fraud. In [296, 326], utilized these datasets in classification setting. However, these three datasets are never employed for clustering-based anomaly detection framework; and the assumption is made that minority class samples are anomalies for these datasets. In [296, 326], analysis of the classifiers reveals that the results are marginal (i.e. apart from the under sampling) because their attributes are extremely non-discriminative and obtaining similar outcomes as in [296, 326] or a slight enhancement indicates the efficacy of the proposed subspace anomaly detection methods.

8.3 Optimization Problem Formulation

The subspace detection method used in this chapter employs the sparsity coefficient which is a variation of Z-score [271] for anomaly detection.

$$Z - score = \frac{x - \mu}{\sigma}$$

The objective function is formulated by approximating the normal distribution by binomial distribution with the help of Bernoulli trials. Here the assumption would be made that the data is non-normal. Therefore, normal distribution estimated through binomial distribution based on Bernoulli trials [238].

$$y = \frac{X - np}{\sqrt{np(1-p)}}$$

Here, random variable following binomial distribution is denoted by ‘X’ while sample size, chances of success, mean and standard deviation are denoted by ‘n’, ‘p’, and ‘np’ and ‘ $\sqrt{np(1-p)}$ ’ respectively. Anomalies are assumed to be non-normal and in accordance with the fundamentals of Chebyshev’s inequality, any sample that lies outside the limit of $[\mu-3\sigma, \mu+3\sigma]$ would be considered an anomaly. When data is non-normal, 89% of the data lies in the $\mu-3\sigma$ and $\mu+3\sigma$ limit as per by Chebyshev’s inequality. In this context,

$$S(D) = \frac{N(D) - (N * f^d)}{\sqrt{N * f^d * (1 - f^d)}}$$

N is the total number of available samples, ‘M’ represents the no. of data dimensions or features, ‘f’ is that fraction of total observations/data which is present in subspace. Here, N(D) represents the number of samples present in d-dimensional subspace. N(D) is normally null when the subspace is empty. S(D) is the sparsity coefficient, which should be minimized, for the M-dimensional dataset, by generating d-dimensional subspaces of the data specified using the range ‘ $\emptyset$ ’

$$\text{Minimize } S(D)$$

$$M > 0$$

$$0 \leq x \leq \emptyset$$

$$d = 2, 3, \dots, d_{fixed}$$

When the subspace is sparse N(D) tends to 0, Therefore:

$$S(0) = -\sqrt{\frac{N}{\emptyset^d - 1}}$$

So, a sample is considered an anomaly when it lies outside of the limit of $\mu - k\sigma$ and $\mu + k\sigma$, where $k=3$ equals the amount of standard deviations, ‘ σ ’ from mean ‘ μ ’. Assuming $S(0)=k$, the following equation is derived for fixing the upper limit of ‘d’.

$$d = \left\lceil \log_{\emptyset} \left[\left(\frac{N}{k^2} \right) + 1 \right] \right\rceil$$

First, it is worth noting that each dimension denotes a dataset feature or attribute. In the case when $x_i = 0$, the respective data feature or attribute is not considered when calculating S(D). Before optimization, the sample indices are noted, and then the elements are sorted in the ascending order of the value of the attributes. These organized and sorted elements are then divided into equal parts of ‘ $\emptyset$ ’. N(D), is intersection of the indices in the d-dimensional subspace.

Initially, ‘k’ is fixed according to need. A trial-and-error approach is used to select ‘ $\emptyset$ ’. Afterwards, ‘d’ is replaced in the equation, in such a way that it should not be less

than 2. Moreover, it is replaced in such a way that placing the k in $S(D)$ (for $N(D)=1$) makes $S(D)$ more than ' k '. Generally, the optimization processes create a new solution vector which satisfies the constraints in the problem. This solution vector is then employed in computing the objective function. In case when the objective function performs better than the rest of the solutions, it should be placed into the population. This process should be repeated until the termination criterion is achieved.

8.4 Proposed approach for anomaly detection in keystroke data

Our proposed anomaly detection is employed for user authentication. User authentication is performed using the subspace-based anomaly detection methods, i.e. using HS, IMHS, DE, MDE IMHS+MDE, and distance-based anomaly detection methods. The process flow is illustrated in Figure 8.1. When the comparison is performed across different evolutionary computing algorithms, we need to set the maximum number of iterations to a fixed value, to check the speed of convergence to optimum.

8.4.1 Euclidean-based Method

In Euclidean-based anomaly detection algorithm each test timing vectors are compared with the baseline vector generated from ' n ' dimensional training timing vectors using the Euclidean distance. Any sample falling outside the $[\mu-3\sigma, \mu+3\sigma]$ in n -dimensional space is treated as an anomaly with the Euclidean distance between the test vector and the mean vector. Euclidean distance is computed by:

$$d(p, q) = d(q, p) = \sqrt{\sum_{i=1}^n (q_i - p_i)^2}$$

8.4.2 Manhattan-based Method

In Manhattan-based anomaly detection algorithm each test timing vectors are compared with the baseline vector generated from ' n ' dimensional training timing vectors using the Manhattan distance. Any sample falling outside the $[\mu-3\sigma, \mu+3\sigma]$ in n -dimensional space is treated as an anomaly with the Manhattan distance between the

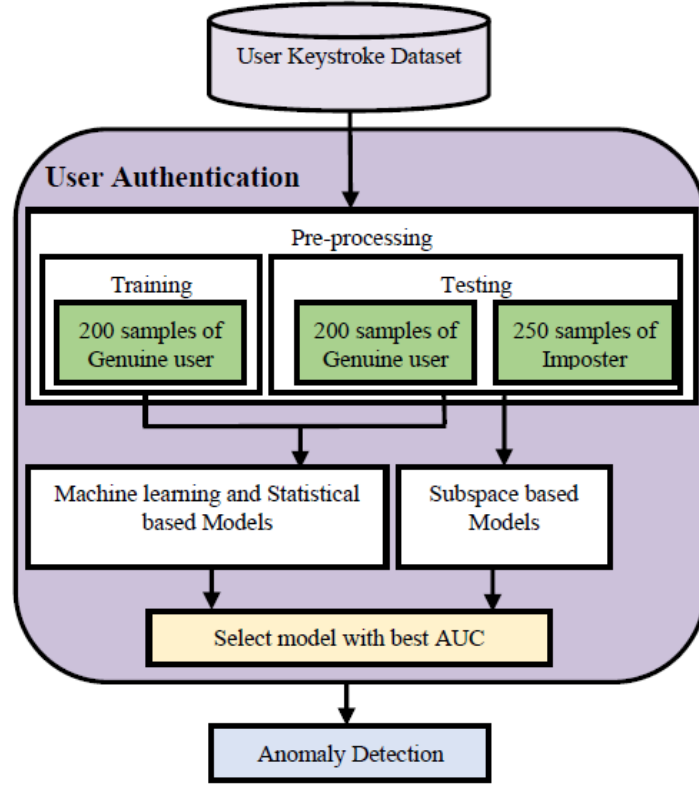


Figure 8.1: Proposed approach for detecting anomalies in keystroke data

test vector and the mean vector. Manhattan distance is computed by:

$$d(x, y) = \sum_{i=1}^N |x_i - y_i|$$

8.4.3 Mahalanobis-based Method

The Mahalanobis-based method is the extension of the Euclidean anomaly detection algorithm. It computes the co-variance matrix along with the baseline vector on the training timing vectors. While testing, the Mahalanobis distance is computed between the baseline vector and the test timing vector. The Mahalanobis distance is computed by:

$$d(x, y) = (x - y)^T S^{-1} (x - y)$$

Any sample falling outside the $[\mu-3\sigma, \mu+3\sigma]$ in n-dimensional space is treated as an anomaly.

8.4.4 Harmony Search

Harmony Search (HS) is run for maximum number of iterations. HS has two explicit parameters, i.e. harmony memory considering rate - 'hmcr' and pitch adjusting rate - 'par'. These parameters are fixed between 0 and 1. (Please see Figure 8.2)

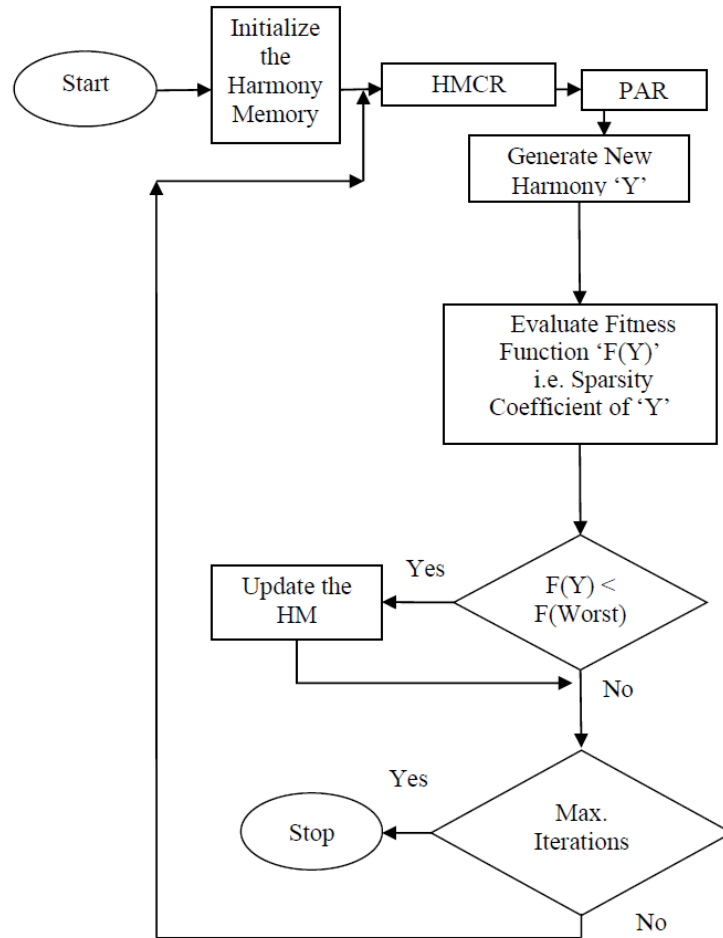


Figure 8.2: Flowchart for HS-based Subspace Anomaly Detection

8.4.5 Improved Modified Harmony Search

Improved Modified Harmony Search (IMHS) is run for maximum number of iterations. IMHS has one explicit parameters, i.e. 'par' and is fixed between 0 and 1. 'hmcr' parameter is linearly incremented from 0.7 to 0.9. (Please see Figure 8.3)

Steps

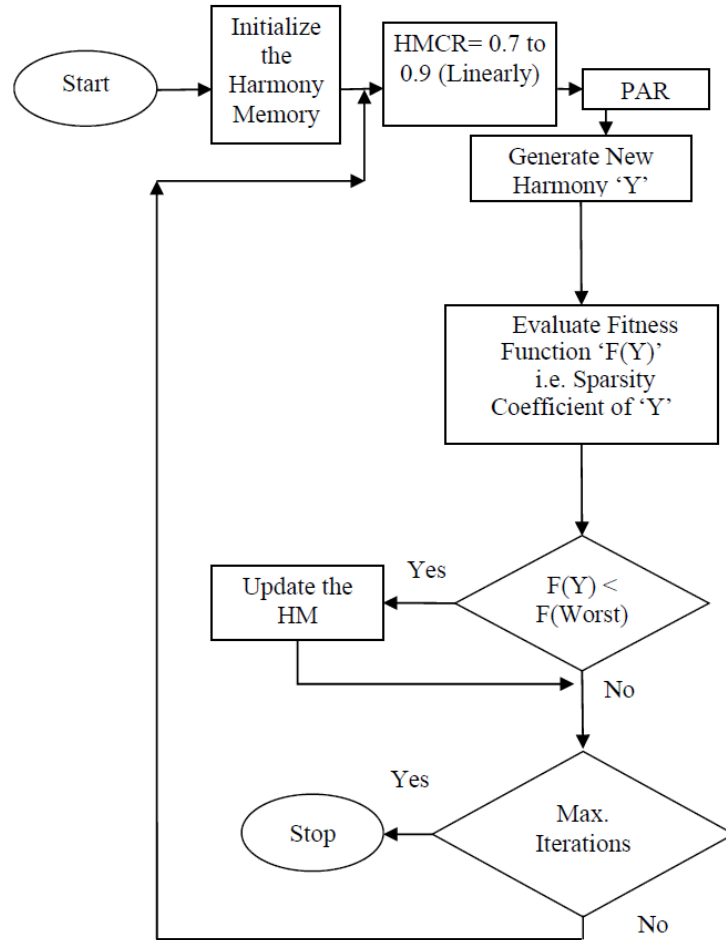


Figure 8.3: Flowchart for IMHS-based Subspace Anomaly Detection

8.4.6 Differential Evolution

DE is used to search the sparse subspaces with sparsity coefficient as the objective function. Crossover rate and mutation in DE do the job of exploration and exploitation of sparse subspaces. Therefore, tuning of CR is an important task. DE replaces the current solution in the selection. ‘CR’ is the DE parameter called rate of crossover, which is the range of 0 and 1. Also, ‘F’ is another DE parameter called differential weight, which is in the range of 0 and 2. However, here we fixed F to be ‘1’ because it is an integer optimization problem. DE is run for maximum number of iterations. (Please see Figure 8.4)

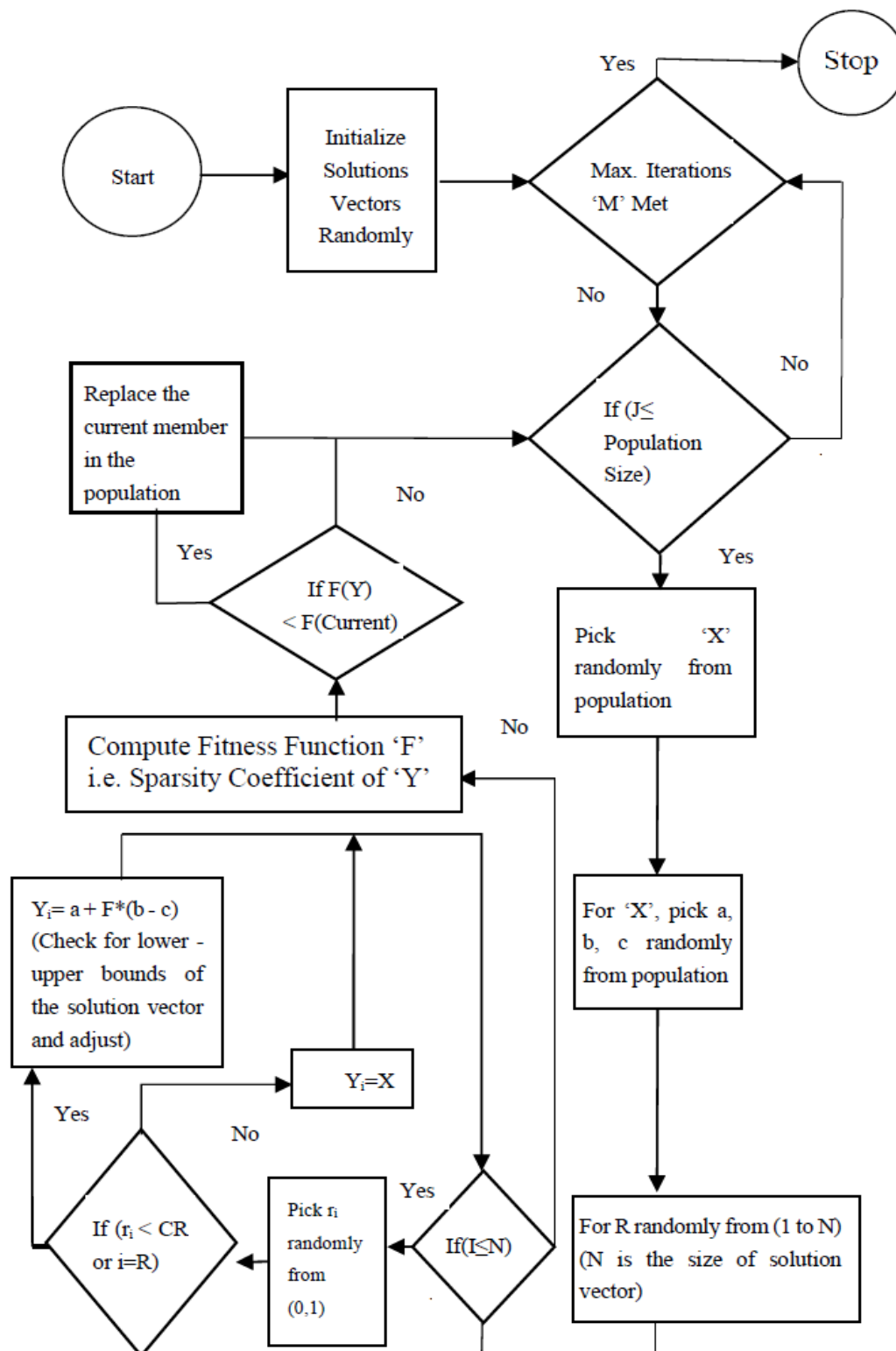


Figure 8.4: Flowchart for DE-based Subspace Anomaly Detection

8.4.7 Modified Differential Evolution

Modified Differential Evolution (MDE) is used to search the sparse subspaces with sparsity coefficient as the objective function. Crossover rate and mutation in MDE do the job of exploration and exploitation of sparse subspaces. Therefore, tuning of CR is an important task. The MDE has a simple improvement in selection operation. MDE replaces the worst solution in the selection, instead the current solution as in DE. ‘CR’ is the DE parameter called rate of crossover, which is the range of 0 and 1. Also, ‘F’ is another DE parameter called differential weight, which is in the range of 0 and 2. However, here we fixed F to be ‘1’ because it is an integer optimization problem. MDE is run for maximum number of iterations. (Please see Figure 8.5)

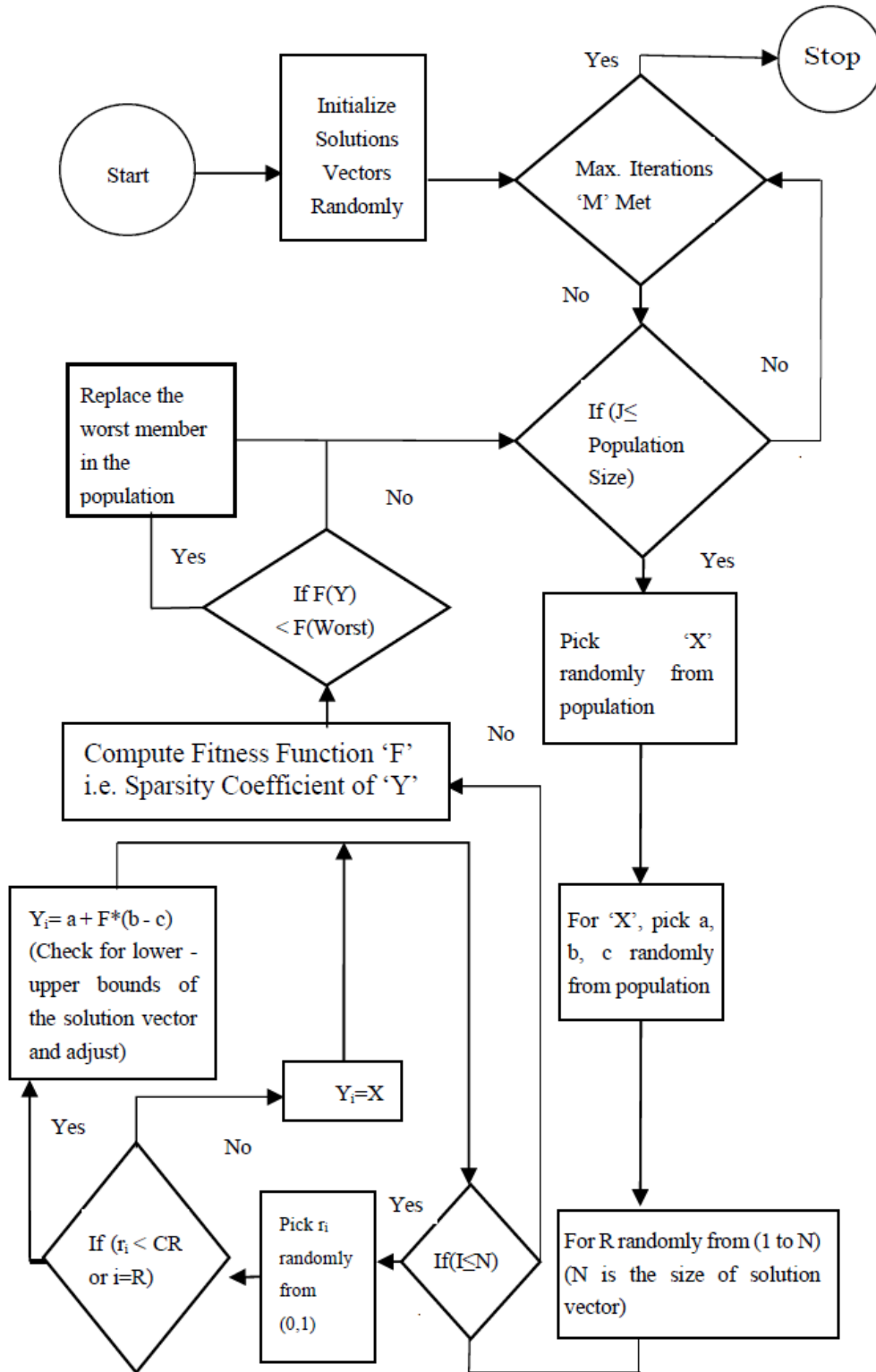


Figure 8.5: Flowchart for MDE-based Subspace Anomaly Detection

8.4.8 Improved Modified Harmony Search + Modified Differential Evolution

IMHS-based subspace anomaly detection is run for half number of iterations and MDE-based subspace anomaly detection is run for half number of iterations. Then the intersection of the results of both IMHS and MDE anomaly detection algorithms is taken. This is called IMHS+MDE-based subspace anomaly detection. IMHS+MDE has three explicit parameters, i.e. ‘par’, ‘CR’ and ‘F’. These parameters are fixed between 0 and 1. ‘hmcr’ parameter is linearly incremented from 0.7 to 0.9. ‘F’ is fixed at 1 as the problem is modeled as an integer optimization problem.

8.5 Proposed approach for anomaly detection in banking and insurance data

The anomaly detection methodology proposed in this section is based on the MDE-based subspace detection. The assumption made in this work is that the anomalies are minority class samples. See Figure 8.6 for the process flow. Taking cue from the results in Section 8.7.1, MDE is chosen as the optimization technique as it is simple and also yielded the best results compared to other subspace methods. A slight improvement in terms of selection constraint is proposed to improve MDE-based subspace anomaly detection.

8.5.1 Modified Differential Evolution with selection constraint

MDE is used in this chapter for searching the sparse subspaces which have the sparsity coefficient as the objection function [13]. In this technique, the mutation and crossover rate (CR) are used to explore and exploit the sparse subspaces. The Modified Differential Evolution technique replaces the worst member of the population with the new solution, whereas DE replaces the current solution. Therefore, MDE has better selection when compared with simple Differential Evolution. ‘CR’ ranges from 0-1. Another DE parameter, is differential weight, denoted by ‘F’ which ranges from 0-2. This is an integer optimization problem. Therefore, ‘F’ is fixed at 1. Stopping criterion for the DE is when the standard deviation in the objective function values in the population falls below a fixed threshold level. We employed an external performance measure as

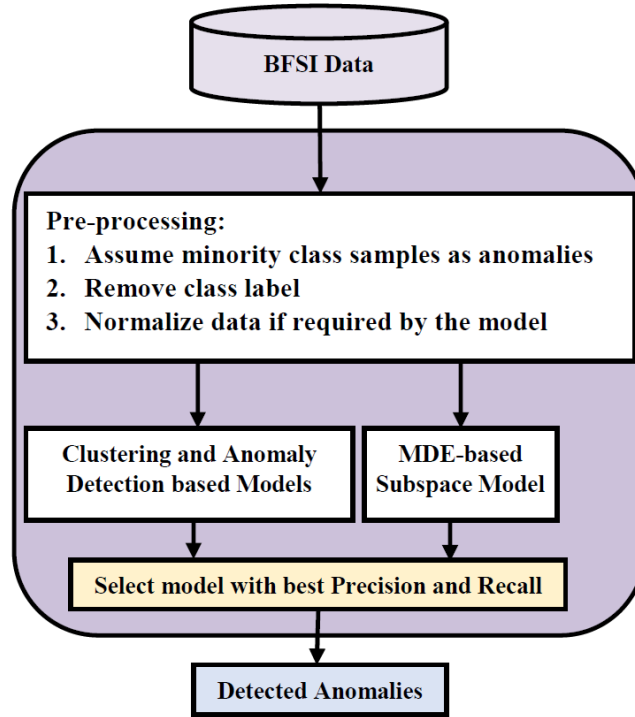


Figure 8.6: Proposed approach for detecting anomalies in banking and insurance data

selection constraint, namely, precision multiplied by recall at every iteration after a pre-specified iteration count of 2000 and taken a snapshot of the population and update the current value of the external performance measure. (Please see Figure 8.7)

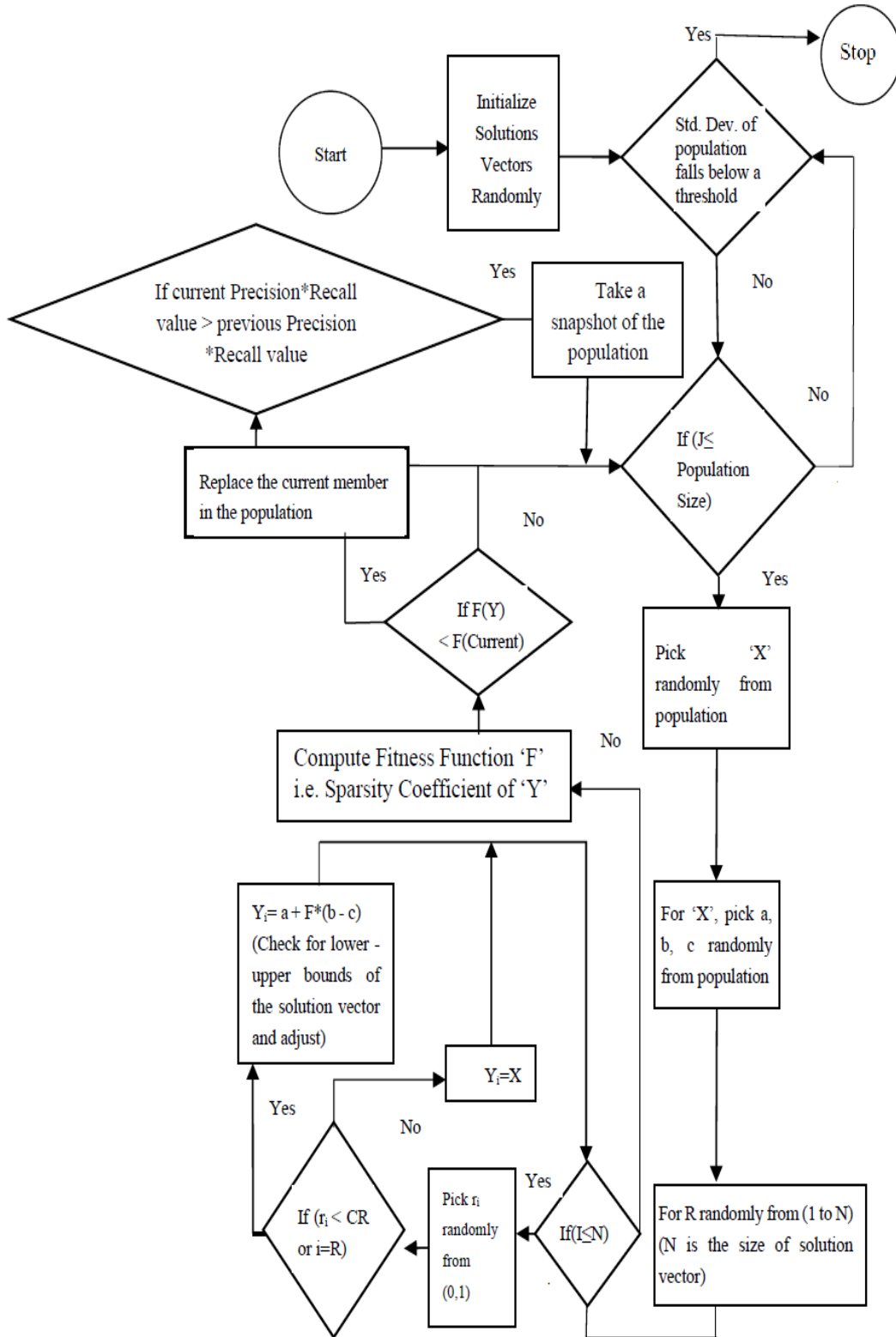


Figure 8.7: Flowchart for MDE-based Subspace Anomaly Detection with selection constraint

8.6 Description of Datasets

8.6.1 Keystroke data

This experiment employs Carnegie Mellon University’s keystroke dataset [152]. The dataset comprises of keystroke-timing data obtained from 51 users in 8 sessions. Moreover, in every session the password typing information for 50 times is gathered as keystroke data. So, a total of 400 samples of keystroke timing per user were collected. A strong password “.tie5Roanl” was chosen for this study. The dataset features are recorded from three keystroke actions for every password character. 1) Hold time reflects the period between pressing and letting go of a key. 2) Up-down time is the period between releasing the key and pressing the second key. 3) On the other hand, down-down time is the gap between pressing the first key and pressing the second key. Each password character features these three synchronized vectors within the dataset. The study made use of all three as explained in [254]. The data is structured into one authentic user and 50 various fake users. The process is repeated 51 times for all 51 users by considering 200 random samples out of 400 samples of the authentic user as genuine and 5 random samples each from the other 50 non authentic users for a total of 250 samples as imposters. Therefore, the data consists of 450 samples each with 200 genuine and 250 imposters repeated for 51 different users.

8.6.2 Churn Dataset

In this chapter, the churn dataset employed is considered from [296]. This churn dataset has the information about a Latin-American bank’s credit card churn. Moreover, it consists of two clusters called as churners and non-churners. This dataset comprises a variable of ground truth and 21 other explanatory variables. In total, it has 14,814 samples; of which, churners are 6.76% whereas non-churners are 93.24%.

8.6.3 Insurance Fraud Dataset

Like the churn dataset, this dataset is also considered from [296]. It basically consists of data pertaining to various claims of automobile insurance between 1994 and 1996. It is an already processed dataset which has a ground truth variable and 24 explanatory variables. In total, it has 14,497 samples; of which, fraud samples are 6% and the non-fraud samples are 94%.

8.6.4 Default Dataset

The default dataset utilized in [326] is considered in this chapter for default detection. It is developed on the credit card information of Taiwanese clients who defaulted on their loans and also who are non-defaulters. It comprises the ground truth variable and 23 explanatory variables. In total, it has 30,000 samples; of this, defaulters are 22.12% and non-defaulters are 77.8%.

8.7 Results and Discussion

The models were run on an Intel (R) Core (TM) i7-6700 processor with 32GB RAM on Rstudio. R is used for developing codes for anomaly detection.

8.7.1 Results of keystroke-based user authentication

For non-subspace-based methods, a mean vector is computed from the genuine train set. Then, three different score vectors are computed as follows: 1) Train scores by calculating a distance measure between the mean vector and the genuine train set 2) User is scored by calculating a distance measure between the mean vector and genuine test set 3) Imposter is scored by calculating a distance measure between the mean vector and imposter test set. For the subspace-based methods, training is not required. Therefore, 450 observations of the test set are directly fed to the Evolutionary Computing-based subspace-based algorithms.

The result of the entire population, after a fixed number of iterations, is considered the output of a run of the optimization algorithm. The experiments are run for 30 times, on each of the 51 different anomaly datasets, and the best AUC of the 30 runs is considered for reporting the results. The population size of EC techniques depends on the percentage of samples considered as anomalies according to Chebyshev's inequality. The iterations or generations of MDE, DE, IMHS and HS are set to 10000. For IMHS+MDE, a value of 5000 generations is set for IMHS and a value of 5000 iterations for MDE

This study chose AUC as the evaluation measure to judge the proposed models. The AUC is calculated by computing the arithmetic mean of specificity and sensitivity measures calculated from the experimental data, which comprise of imposter and genuine samples. In this case, the test samples which fall within $[\mu-3\sigma, \mu+3\sigma]$ are

marked by the models as genuine models; on the other hand, any samples that do not fall within $[\mu-3\sigma, \mu+3\sigma]$ are marked by the models as imposters. The study used eight anomaly detection techniques: five EC-based subspace anomaly detection techniques, and three distance-based anomaly detection techniques namely Manhattan, Euclidean, and Mahalanobis. Subspace-oriented techniques were selected because of the surprising behavior of distance-based techniques for medium or high dimensional data as said in [12].

Table 8.1: Results of Proposed Algorithms in AUC

Methods	AUC	St.Dev.	Parameters
<i>MDE-based Subspace Anomaly Detection</i>	<i>0.652</i>	<i>0.0593</i>	CR = 0.1, F = 1.0, $\emptyset = 6$
IMHS+MDE-based Subspace Anomaly Detection	0.647	0.0675	CR = 0.1, F = 1.0, PAR = 0.1, $\emptyset = 6$
DE-based Subspace Anomaly Detection	0.646	0.0581	CR = 0.15, F = 1.0, $\emptyset = 6$
HS-based Subspace Anomaly Detection	0.601	0.0677	HMCRCR = 0.7, PAR = 0.1, $\emptyset = 6$
IMHS-based Subspace Anomaly Detection	0.592	0.0703	PAR = 0.1, $\emptyset = 6$
Euclidean	0.285	0.1097	-
Mahalanobis	0.501	0.0027	-
Manhattan	0.544	0.0606	-

The mean AUC results of 51 different user datasets are considered for evaluation. Manhattan, Mahalanobis, and Euclidean produced 0.544, 0.501, and 0.285, respectively whereas the proposed methods, IMHS, DE, MDE, HS, and IMHS+MDE, with subspace anomaly detection produced 0.592, 0.642, 0.652, 0.601, and 0.647 AUCs , respectively, as depicted in Table 8.1. While the performance of the subspace-based techniques was better than that of the distance-oriented techniques, the improvement of the best performing proposed model i.e. MDE is minimal in relation to other subspace-based models. For this reason, a non-parametric Wilcoxon Signed Rank test was performed to find the statistical significance of the MDE-based technique with different approaches. The results suggest that the statistical significance with IMHS+MDE is identical; however, we opted for MDE because its complexity less in comparison with that of IMHS+MDE. The results of the same are presented in Table 8.2. If the p-value is below 0.05, it suggests that the null hypothesis is rejected; as such, the statistical significance between the models is present. However, if the p-value is above 0.05 then the null hypothesis is not rejected and there is no statistical significance between the compared models.

Table 8.2: Wilcoxon Signed Rank Test

Compared Methods	Test statistic	p-values
MDE vs IMHS+MDE	810.5	0.0948
MDE vs DE	1008	0.0012
MDE vs HS	1306	1.62e-09
MDE vs IMHS	1311	1.24e-09
MDE Vs Euclidean	1326	5.14e-10
MDE Vs Manhattan	1250	3.74e-08
MDE Vs Mahalanobis	1326	5.13e-10

Table 8.3 shows the comparison of the proposed technique with the state-of-the-art. But, lacking models of clustering-based anomaly detection for keystroke data, where mostly classification-based anomaly detection techniques are reported. Therefore, the comparison is not apt as the proposed technique is a clustering-based anomaly detection technique. But, the proposed technique outperformed k-Means, which is a clustering-based technique.

Table 8.3: Comparison with the State-of-the-art using EER

Model/Algorithm	EER (Avg.)	St.dev.
Deep Secure [204]	0.03	0.024
Deep Belief Nets (DBN) [80]	0.035	0.027
GMM-UBM [80]	0.055	0.052
Median Vector Proximity [15]	0.08	0.055
Manhattan-Mahalanobis (No Outlier) [337]	0.084	0.056
Manhattan-Mahalanobis (Outlier) [337]	0.087	0.06
Manhattan (Scaled) [152]	0.0962	0.0694
Nearest Neighbor (Mahalanobis) [152]	0.0996	0.0642
Outlier Count (z-score) [152]	0.1022	0.0767
SVM (one-class) [152]	0.1025	0.065
Mahalanobis [152]	0.1101	0.0645
Manhattan (Filter) [152]	0.136	0.0828
Neural Network (Auto-Assoc) [152]	0.1614	0.0797
Euclidean [152]	0.1706	0.0952
Fuzzy Logic [152]	0.2213	0.1051
<i>MDE-based Subspace Anomaly Detection (proposed)</i>	<i>0.348</i>	<i>0.0593</i>
k-Means [152]	0.3722	0.1391
Neural Network (Standard) [152]	0.8283	0.1483

8.7.1.1 AUC and EER

Area Under Curve (AUC) is considered as the evaluation metric for the study. It is defined as follows:

$$AUC = \frac{(Sensitivity + Specificity)}{2}$$

EER is computed when both False Acceptance Ratio (FAR) = False Rejection Ratio (FRR), must be minimized.

$$FAR = FPR = FP / (FP + TN) = (1 - Specificity)$$

$$FRR = FNR = FN / (FN + TP) = (1 - Sensitivity)$$

where FP: False positive, FN: False Negative, TN: True Negative, TP: True Positive

EER is derived from the above equation of AUC as follows:

$$AUC = 1 - [(FRR + FAR) / 2]$$

$$(FRR + FAR) / 2 = 1 - AUC$$

When FAR=FRR, **EER = 1 - AUC**

8.7.2 Results of banking and insurance data

The best result of 25 runs is chosen as the final result, as the proposed models are stochastic. Partition Around Medoids (PAM) [257] and K-Means [120], the basic clustering techniques, were used to on the datasets, and compared with the proposed models. In order to augment the hypothesis that the proposed models work more effectively as compared to clustering-based anomaly detection methods. Angle Based Outlier Detection (ABOD) [166] and LOF [41] were used. There is a strong similarity between subspace anomaly detection techniques and LOF.

In this chapter, to evaluate our models, we selected precision multiplied by recall as our performance metric for the selection constraint. Apart from calculating precision multiplied by recall, we also computed AUC, but, not as selection constraint, to judge the models even more profoundly. Here, anomalies detected as anomalies is TP, non-anomalies detected as non-anomalies is TN, anomalies detected as non-anomalies is FN and non-anomalies detected as anomalies is FP.

$$\text{Sensitivity or Recall} = \frac{TP}{TP + FN}$$

$$\text{Specificity} = \frac{TN}{TN + FP}$$

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{AUC} = \frac{\text{Sensitivity} + \text{Specificity}}{2}$$

Table 8.4, Table 8.5, and Table 8.6 showcase the results of the proposed models as opposed to some of the selected models on three banking and insurance datasets. The proposed selection constraint yielded better results with respect to the churn dataset than the other proposed model that did not have selection constraint and other standard models. It can be noted that the model that had a selection constraint performed slightly better than the proposed model and other standard models on the two remaining datasets: default and insurance datasets. The churn dataset produced a precision estimated at 20.11 while the AUC was 0.681 and the recall was measured at 50.89. The AUC with the selection constraint on churn dataset is the highest when contrasted with

the two other datasets. It was also observed that the high level of precision derived from the suggested frameworks demonstrate that the identified false positives are less as opposed to the other models. The LOF yielded a higher recall in the three datasets; despite this, the AUC and precision were less than the proposed models. The MDE-based subspace anomaly detection frameworks do not employ a distance measure when searching for anomalies. However, the LOF-oriented outlier detection method relies on the distance-based density measure which, as explained in [12], has a surprising behavior when dimensionality increases.

Table 8.4: Results of Churn Dataset

Churn Dataset			
Model	Precision	Recall	AUC
MDE based Subspace Detection with Selection Constraint	20.11	50.89	0.681
MDE based Subspace Detection	24.09	17.26	0.566
LOF [41]	6.23	77.84	0.464
K-Means [120]	1.18	1.89	0.452
PAM [257]	6.41	26.44	0.492
ABOD [166]	7.55	10.07	0.505

Table 8.5: Results of Insurance Fraud Dataset

Insurance Fraud Dataset			
Model	Precision	Recall	AUC
MDE based Subspace Detection with Selection Constraint	7.75	36.51	0.544
MDE based Subspace Detection	9.13	9.64	0.517
LOF [41]	6.09	76.59	0.507
K-Means [120]	3.44	0.003	0.498
PAM [257]	2.85	19.17	0.388
ABOD [166]	7.03	20.15	0.504

Table 8.7 presents the parameters of MDE-based subspace anomaly detection, and indicates that it has two explicit parameters i.e. ‘F’ and ‘CR’ and one implicit parameter called ‘ $\emptyset$ ’. Population size is increased till the results stabilize as each population member represents subspace with anomalies.

Table 8.6: Results of Default Dataset

Default Dataset			
Model	Precision	Recall	AUC
MDE based Subspace Detection with Selection Constraint	24.09	44	0.523
MDE based Subspace Detection	23.08	31.78	0.508
LOF [41]	21.69	78.22	0.49
K-Means [120]	22.25	85.08	0.503
PAM [257]	16.81	28.82	0.441
ABOD [166]	27.59	12.7	0.516

Table 8.7: Parameters of MDE-based Subspace Anomaly Detection

MDE Parameters	Dataset	CR	F	$\emptyset$
With Selection Constraint	Churn	0.5	1.0	23
	Insurance Fraud	0.1	1.0	13
	Default	0.1	1.0	33
Without Selection Constraint	Churn	0.4	1.0	23
	Insurance Fraud	0.05	1.0	13
	Default	0.3	1.0	33

8.8 Conclusions and Future Directions

This study proposed a bunch of subspace clustering techniques based on hybrid and standalone evolutionary techniques namely, HS, IMHS, DE, MDE, IMHS+MDE, for anomaly detection in case of keystroke user authentication, churn detection, default detection, and insurance fraud detection. One of the proposed models uses an evaluation measure in terms of precision multiplied by recall, as the constraint within selection while the other model does have any selection constraint. With respect to precision and AUC, the proposed model with selection constraint is better than the other proposed model. Further, it is possible to perform feature selection and under-sampling to get better results. Powerful optimization algorithms can also be employed, alongside with new optimization operators to obtain better results than the proposed models.

Chapter 9

Differential Evolution trained Fuzzy Cognitive Map: An application to modelling efficiency in banking

9.1 Introduction

When considering connections of different concepts, one may opt to use mapping strategies to understand the relationships of the concepts. According to Kosko [162] when introducing the Fuzzy Cognitive Map (FCM) concept in 1986, there would be a solution generated for mapping concepts to explain causal knowledge and reasoning processes best. In this case, the FCM utilizes concepts of the cognition processes that relay the interconnection of the concepts under evaluation. One may utilize the FCM in a situation that needs visualized causal knowledge in an easy to grasp format.

The particular technique of FCM with the causal knowledge or representation belongs to the family of neuro-fuzzy systems. These systems are known for their capability to represent decision making models. In particular, the FCM serves as a useful tool when predicting, forecasting, and classifying tasks to make their causal knowledge visible and easy to use in subsequent psychological scenes and imaginative applications.

The FCM [82] requires training to adapt it and tune it to the weights used in the mapping process. Some training options arise from the knowledge obtained from

Artificial Neural Networks (ANN). This study will focus on the Differential Hebbian Learning (DHL) that is useful for training the FCM. Training is necessary to find the appropriate weights between concepts as a means of ensuring the model reaches a desired state of equilibrium. The Hebbian algorithm is deterministic and therefore requires modifications when using it with the FCM. Besides, Hebbian algorithm, there are rest of other options for training, with the most known stochastic ones being the Swarm Intelligence and the Evolutional Computation (EC) approaches. The intention of using these training methods is to minimize human intervention in the FCM. Eventually, the training using EC techniques also creates adaptive FCMs without making them black box and hence making them different from ANNs.

This study focused on evolutionary algorithms, in particular, Differential Evolution (DE). These are population-based stochastic search strategies allowing the updating of population variables using heuristics. The updates take place until a point of optimum or near optimum representation. The key differences in the methods are in their options for encoding variables and the types of heuristics they use during the refreshment of variables. The metaheuristics in the evolutionary algorithms family include Genetic Algorithms [110], Differential Evolution [293], Particle Swarm Optimization [151], Ant Colony Optimization [67], and Firefly Algorithm [320].

The role played by the Banking Financial Services and Insurance (BFSI) sector in a country is critical. The sector covers operations and financial challenges experienced every day with emergent ones requiring automated and robust means of handling issues and advancing operational agendas. The techniques under EC promise to solve visualization and modeling challenges that plague operations in the BFSI given that they can provide global or near-global optimum outcomes. Decision-making performed through FCM make it viable to be employed for bank efficiency prediction.

9.2 Motivation

At least four main reasons signify the intention of carrying out the research. First, there is no known bank efficiency prediction model based on the important financial ratios and using the FCM. Secondly, the causal knowledge and the cognitive processes responsible for relationships between various financial ratios representing bank efficiency

will provide an important knowledge source for practitioners. In particular, the relationships among of Utilization of Assets, Profit Margin, Return on Equity, Expenses Ratio and Return on Assets. Thirdly, the sequential and batch mode of training in FCM needs to be compared and analyzed. Lastly, Differential Evolution trained Fuzzy Cognitive Map is never reported for banking application problems.

9.3 Contribution

In the present work, the DE-trained, and Hebbian-trained FCM models are developed separately for the banking sector's efficiency prediction. Two FCM algorithms in sequential and batch modes are developed. Lastly, two FCM models are proposed for the efficiency prediction of the banks by looking at five financial ratios.

9.4 Related work

A review on the studied subject reveals several connections of the present study with previous ones as follows. A seminal work utilized ECs for training FCMs where there was a genetic algorithm trained FCM [291]. A data-driven non-linear Hebbian learning appeared as a viable proposal for training different samples of data with FCMs [290].

Several studies [138, 139, 240] used FCM prediction models in radiation therapy systems, dengue outbreak prediction in a geospatial manner, and other conventional development applications in agriculture such as precision farming. There was also usage of FCMs in predicting cotton crop production [235]. These studies confirm the robustness of applying the FCMs for causal relationship. Papageorgiou [236] offers a survey showing the types of training with a focus on the Hebbian training and the EC-based training as well as a hybrid of both training methods in various FCM applications by giving a general understanding of the versatility and usability of the training methods. In another survey, the same author [237] suggests different types of cognitive maps and the context in which to use FCM in different areas. The analysis of stability, parameter convergence when using adaptive estimation and the evaluation of the existence of unique solutions for the concept values in FCM are additional studies with application of FCM [33, 34]. Another study [42] looked at the different behaviors of diverse activation functions including sigmoid, hyperbolic, threshold linear

function and stepped function to contribute to the FCMs. In [246], ranking of Indian banks according to their productivity based on a data envelopment analysis and with a fuzzy multi-attribute decision-making (DEA-FMADM) hybrid model is performed. A description of ‘catastrophic forgetting’ for sequential learning of neural network is presented in [259].

9.5 Proposed Methodology

This study uses continuous FCM. Specifically, it follows a modified Kosko activation function and sigmoid threshold function as noted in the next subsection. The Mean Absolute Percentage Error (MAPE) is used as error measure or fitness function, which needs to be minimized. The model also uses a pre-determined iteration or a MAPE value lower than 0.1% for its convergence criterion. Section 9.6.4 will explain the steps for training the Differential Evolution FCM. The Hebbian learning with the Oja weight update rule [290] in both training modes with an explanation is presented in section 9.6.5.

9.5.1 Activation Function

The model shows concepts using C_i and $i = 1 \dots N$. Concepts have each an activation value $A_i \in [0, 1]$, $i = 1 \dots N$ as well as signed fuzzy weights W_{ij} . They have values within the $[-1, 1]$ range. They cover the edge between C_i and C_j while $j = 1 \dots N$. Moreover, in this activation function, N is the total number of concepts. The model calculates the value of every concept A_i and a particular occurrence ‘k’ by taking the sum of the preceding value A_i in an existing occurrence ‘k-1’ together with the value of A_i of C_i in a particular occurrence ‘k-1’ also multiplied by the concept link weight W_{ij} value.

$$A_i^k = \text{Sigmoid}(A_i^{k-1} + \sum_{\substack{j \neq i \\ j = 1}}^N A_j^{k-1} * W_{ji}^{k-1})$$

$\text{Sigmoid}(x) = \frac{1}{1+e^{-\lambda x}}$, where $\lambda = 1$.

9.5.2 Mean Absolute Percentage Error

The study uses the MAPE as a means to predict the performance of the model. The following formula will be applied for MAPE.

$$MAPE = \frac{100}{n} \sum_{t=1}^n \left| \frac{Actual_t - Predicted_t}{Actual_t} \right|$$

9.5.3 Root Mean Squared Error

Moreover, the models also utilize the Root Mean Squared Error (RMSE) for prediction.

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^n (Actual_t - Predicted_t)^2}$$

9.5.4 DE trained FCM

The Algorithms 11 and 12 present the DE trained FCM in sequential and batch modes of training.

9.5.5 The FCM with Hebbian Training Mode

The model uses the Oja learning rule [290]. It modifies the standard Hebb's rule. It solves the stability problems in the model using multiplicative normalization. In particular, the Oja weight update rule (WUR) [290] used in training the FCM will be applied in both sequential and batch modes. An important realization is the weight update is a nonlinear rule. It assists in capturing the non-linearity when using a data-driven methodology. The WUR formula is as follows.

$$w_{ij}^{n+1} = w_{ij}^n + \eta * A_j^{new} * (A_j^{old} - w_{ij}^n * A_j^{new})$$

9.5.6 Sequential and Batch modes of Training

In the first case which involves sequential training mode, there is a single concept vector used one after another. However, the mode does not hold the memory of the previous trained values, and this serves as its main limitation. On the other hand, the sequential mode allows for fast training when there are limited training options. If there is a need for repeated training of the individual concept vectors, then the sequential training

Algorithm 11: Sequential DE trained FCM

```
no_itr  $\leftarrow$  fixed_value
ncv_train  $\leftarrow$  nrow(concept_vec_train)
while i  $\leq$  ncv_train do
    cv_train  $\leftarrow$  concept_vec_train[i]
    itr  $\leftarrow$  1
    while itr  $\leq$  no_itr do
        Step-1: Generate weights of FCM using DE heuristic
        Step-2: Apply Modified Kosko with Sigmoid on the cv_train for the
        weights generated by DE
        Step-3: Compute  $MAPE_{new}$  on actual and predicted productivity
        concept values of cv_train
        Step-4: If MAPE of new weight vector less than the MAPE value of
        current weight vector in the DE population then replace the current
        weight vector and MAPE value in DE with the new weight vector and
        new MAPE value
    end
end
ncv_test  $\leftarrow$  nrow(concept_vec_test)
no_pop  $\leftarrow$  nrow(DE_Population)
while pop  $\leq$  no_pop do
    while j  $\leq$  ncv_test do
        cv_test  $\leftarrow$  concept_vec_test[j]
        Apply Modified Kosko with Sigmoid on the cv_test for the weight vector
        in the DE population i.e. DE[pop]
    end
    For all the samples in concept_vec_test compute  $MAPE$  on actual and
    predicted productivity concept values
end
Pick the weight vector with less MAPE value on cv_test from the DE_Population
```

Algorithm 12: Batch DE trained FCM

```
no_itr  $\leftarrow$  fixed_value
ncv_train  $\leftarrow$  nrow(concept_vec_train)
itr  $\leftarrow$  1
while itr  $\leq$  no_itr do
    Step-1: Generate weights of FCM using DE heuristic
    while i  $\leq$  ncv_train do
        cv_train  $\leftarrow$  concept_vec_train[i]
        Step-2: Apply Modified Kosko with Sigmoid on the cv_train for the
        weights generated by DE
    end
    Step-3: For all the samples in concept_vec_train compute  $MAPE_{new}$  on
    actual and predicted productivity concept values
    Step-4: If MAPE of new weight vector less than the MAPE value of current
    weight vector in the DE population then replace the current weight vector
    and MAPE value in DE with the new weight vector and new MAPE value
end
ncv_test  $\leftarrow$  nrow(concept_vec_test)
no_pop  $\leftarrow$  nrow(DE_Population)
while pop  $\leq$  no_pop do
    while j  $\leq$  ncv_test do
        cv_test  $\leftarrow$  concept_vec_test[j]
        Apply Modified Kosko with Sigmoid on the cv_test for the weight vector
        in the DE population i.e. DE[pop]
    end
    For all the samples in concept_vec_test compute  $MAPE$  on actual and
    predicted productivity concept values
end
    Pick the weight vector with less MAPE value on cv_test from the DE_Population
```

Table 9.1: The Six Concepts

S.No.	Input Ratio name	Definition
1	Utilization of Assets (UA)	Interest income+Non-interest income/Assets
2	Profit Margin (PM)	Net profit/Interest income+Non-interest income
3	Return on Equity (ROE)	Net profit/Owned funds
4	Expenses Ratio (ER)	Operating Expenses/Spread
5	Return on Assets (ROA)	Assets/Net profit
	Output: Efficiency	

will be slow and therefore disadvantageous over the other. On the other hand, the batch training mode takes all the concept vectors at a go for the FCM training to allow for fast turnaround times when dealing with small datasets. Otherwise, it also comes with its limitation of a very long training times for large data in training. In the DE-FCM subsection of this chapter, presents the detailed steps of the respective modes of training.

9.6 Description of the Dataset

This study utilized data from the source [246] with five financial ratios and their corresponding efficiency levels as derived from the DEA-FMADM computation model. The data is from 27 different Indian banks, and contains five input and one output concept. The Table 9.1 highlights the concepts with brief descriptions, of the five financial ratios and the output concept i.e. efficiency. In this case, one sees 15 possible concept linkages in the model from the available six concepts. But, there are five input concepts and the one output concept. Therefore, the study utilized 13 concepts linkages, as noted from the domain expert and dropped the remaining.

9.7 Experimental Setup

The experiments were conducted on a computer with the following specification. It is an Intel Core i7-6700 processor with 32GB RAM and run in fastR environment. Moreover, the data was split into 60:40 percentage, where the first segment represent the training dataset and the second segment the test dataset. Moreover, this is a random split

repeated five times. In the end, the data is split into 16 concept vectors or samples. The models used the 16 concept vectors for training while the other 11 concept vectors or samples in the case of this experiment are used for testing. The experiment used five training and test sets and selected two models that are DE-FCM and Hebbian trained FCM. The FCM maps developed in the respective modes of training are selected. The DE-FCM relies on best weight matrix from a population of 25 on the test data. On the other hand, the DE-FCM was ran in the sequential and batch training modes for 30 runs as DE is the stochastic by nature. The Hebbian-FCM is run only once in both the training modes, as it is deterministic. The iterations for DE are set to 400. CR is set to 0.9 and F is set to 0.95 for the DE-FCM by parameter tuning.

9.8 Results and Discussion

The experiment were done in a 5x2 fold cross validation setup which yield the following results as tabulated below.

Table 9.2: Sequential mode of training results for FCM

5X2 FCV	Hebbian-FCM (MAPE)	DE-FCM (MAPE)	Hebbian-FCM (RMSE)	DE-FCM (RMSE)
1	30.9878	5.2727	0.23562	0.04811
2	29.2433	3.9027	0.22489	0.03765
3	16.1033	14.8488	0.14209	0.12802
4	20.51599	12.8809	0.15409	0.11803
5	33.12868	4.1504	0.24733	0.04473
Mean	25.995814	<i>8.2111</i>	0.200804	0.075308
Variance	53.558542	<i>27.387970</i>	<i>0.002396</i>	<i>0.001924</i>
t-Value	<i>4.420107317</i>		<i>4.269079683</i>	
p-value	<i>0.002226</i>		<i>0.002727</i>	

Table 9.2 shows results for the sequential trained FCM indicating the outcome of statistical analysis showing t-value and level of significance measured by p-value. The results of the t-test confirm that there was a statistical significance ($p < 0.1$) for sequential DE-FCM. Under sequential training of FCM, the DE-FCM outperformed Hebbian FCM. The third table, (Table 9.3) shows no significant difference between Hebbian and DE training of FCM under the batch mode of training. The statistical test

Table 9.3: Batch mode of training results for FCM

5X2 FCV	Hebbian-FCM (MAPE)	DE-FCM (MAPE)	Hebbian-FCM (RMSE)	DE-FCM (RMSE)
1	6.446347	5.525913	0.05554	0.05229
2	5.18635	5.231981	0.04297	0.05401
3	15.3066	15.213922	0.13614	0.12734
4	18.95404	15.201556	0.14485	0.12717
5	7.921394	4.337817	0.06182	0.04286
Mean	10.7629562	<i>9.1022378</i>	0.088264	0.080734
Variance	36.386283	<i>31.255764</i>	0.002328	0.001821
t-Value	<i>0.4515155</i>		<i>0.2613553</i>	
p-value	<i>0.66361</i>		<i>0.80042</i>	

also corroborate the findings of the respective modes of training, showing a significant difference in the case of sequential mode and no significance in the batch training mode.

The learning process utilizes MAPE as the error measure. RMSE was computed only on the test data as additional factor of judgment after the training with MAPE. The tie noted in the batch mode required further analysis. In case of batch mode of training, the training times are compared for coming to conclusion. The DE trained FCM was faster by 19% than the Hebbian trained FCM. The computing power, used multi-threaded processing with a high-end core-i7 processor. Therefore in the batch mode, the DE.FCM should be showing significance in the low end and single threaded configurations where the computational power is notably less than in the current setup. The results also presented the FCM visualization for the bank's efficiency data as illustrated below.

Both Hebbian, and DE trained FCMs are used on the experimental dataset. In this case, five financial ratios for efficiency prediction of 27 banks were utilized. Both sequential and batch modes of training are employed to create the visual and cognitive map through FCM. The dataset was small and would lead to inaccuracies when using the 5x2 fold cross-validation. Nonetheless, the intended purpose of using the FCM was to offer predictions. Thus, the 5x2 FCV appears as an authentic validation option.

Figure 9.1 shows the generated FCM graph, but before coming up with the graph, there was a need to train the FCM with Hebbian and DE using sequential and batch modes. Then utilize the built model in training for the testing. Otherwise, the problems

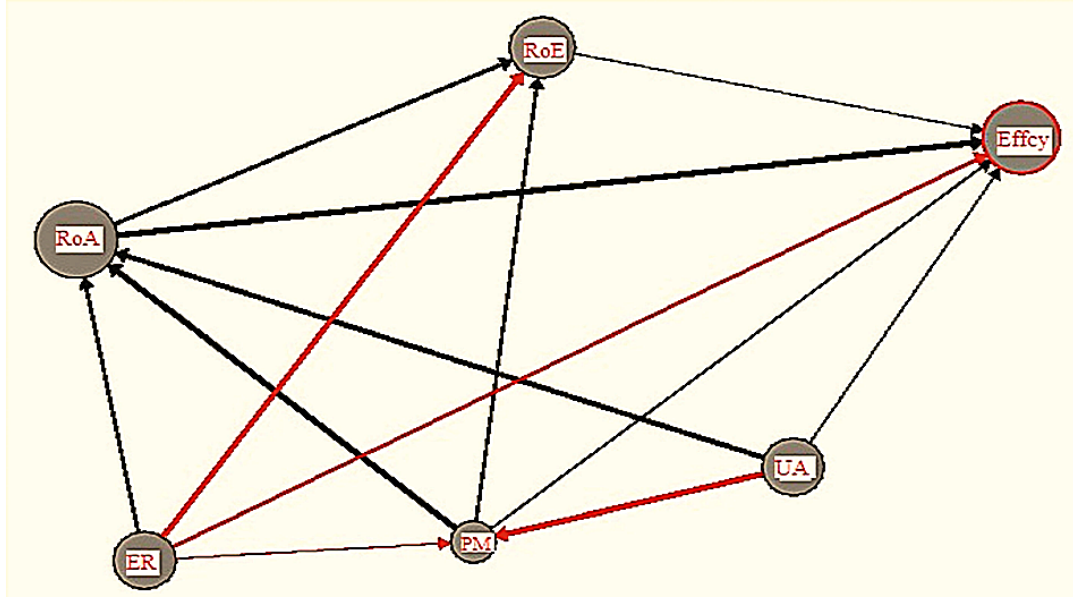


Figure 9.1: The Fuzzy Cognitive Model for the bank's efficiency data

noted above would make the outcomes inaccurate. Some important insights about the FCM in Figure 1 are as follows. The positive weight is indicated by the black edge, and the red edge indicates the negative weight. Moreover, a positive correlation is indicated by the black line, and the red one shows a negative correlation. Furthermore, the thickness of the line addresses the strength of the correlation. The nodes in the graph are Utilization of Assets (UA), Profit Margin (PM), Return on Equity (ROE), Expenses Ratio (ER), Return on Assets (ROA) and Efficiency (Effcy) as explained in table 9.1.

According to the domain expert's knowledge in banking efficiency, the representations of the DE-FCM were most accurate for the two models developed in the experiment. There were only two violations of both positive and negative correlations in the nodes of the FCM, yet it is only the expenses ratio that should be having a negative correlation with the other concepts i.e. four negative correlations. It would be mostly the case because expenses are unlikely to be resulting in profits. The DE-FCM in sequential training mode noted three negative correlations relating to the expense ratio. There were eight correct positive correlations when they should be nine in total. There are possibilities that the models match all the expectations of the domain expert

with an exhaustive list of ratios other than the five utilized as input concepts. The DE-FCM is both descriptive and predictive in its role in explaining five financial ratios. In particular, the sequentially trained DE-FCM should be useful for causal visualization of the five input concepts.

9.9 Conclusions and Future Directions

In the chapter, DE-trained and Hebbian-trained FCM algorithms for efficiency prediction are proposed. The data used, was taken from 27 Indian banks, and the efficiency concept would be the output while five ratios namely Utilization of Assets, Profit Margin, Return on Equity, Expenses Ratio and Return on Assets act as input concepts in the model. The use of sequential and batch modes for training the FCM gave varied results of Mean Absolute Percentage Error (MAPE) on the efficiency concept. The automation of decision making through graphic visualization was one of the intention of the modeling.

In this chapter, a few financial ratios were used to model efficiency, and this approach limits its generalizability to the rest of the banking sector. Nonetheless, in future, we can have an extension of the study to cover additional financial ratios with a direct and indirect influence on efficiency as determined by the expert domain. FCM is used for descriptive visualization and also for predicting of efficiency of banks against various input concepts.

In future, new convergence criteria can also be employed. Also, in future, one can come up with improved training algorithms. In future, there can be models of hybrid training approaches of the Differential Evolution and Hebbian. Further, the idea presented so far from the findings and discussion in this study is that, there are ways to enhance training approaches for FCMs through the formulation of the problem in a multi-objective optimization framework. Further, if-then rules can be generated while building FCM such that it becomes an expert system capable of autonomous sequential decision making based on prescribed input parameters.

Chapter 10

NSGA-II application in Security: A case of key generation for plain text in stream cipher

10.1 Introduction

Cryptography is defined as the study to provide security to information from outsiders who are also called adversaries [210, 292]. Specifically, cryptography involves the encrypting and decrypting of text to stop adversaries from accessing any private message or network. Encryption is the process by which messages are encoded. It does not necessarily block any attempt to access the message and stop the interceptor. The basic encryption procedure involves the message called plain text and is scrambled using an encryption algorithm which results in a ciphertext that needs to be decrypted. A pseudo-random number generator is used for generating a key. Decrypting the message without the key would be difficult. The intended receiver could easily decipher the message with the key shared by the sender. However, deciphering would not be possible for the adversaries without a key.

10.1.1 Symmetric-key Cryptography

Cryptography has two main fields which utilize keys. First is the symmetric-key cryptography in which sender and receiver have access to the key. The second is the public-key cryptography wherein the sender has the public key for the encryption and the

recipient has the private key to decode the ciphertext. One of the advantages of the symmetric-key encryption is that it is easy to use. In addition, it is quicker to encrypt than the public-key encryption. Also, it uses less computational power. If there is a trade off of a key, only the messages between the sender and receiver with that particular key are affected. There is no need to worry about the other messages exchanged with other individuals and other messages are kept secure. Figure 10.1 shows the symmetric-key cryptography process that uses a key.

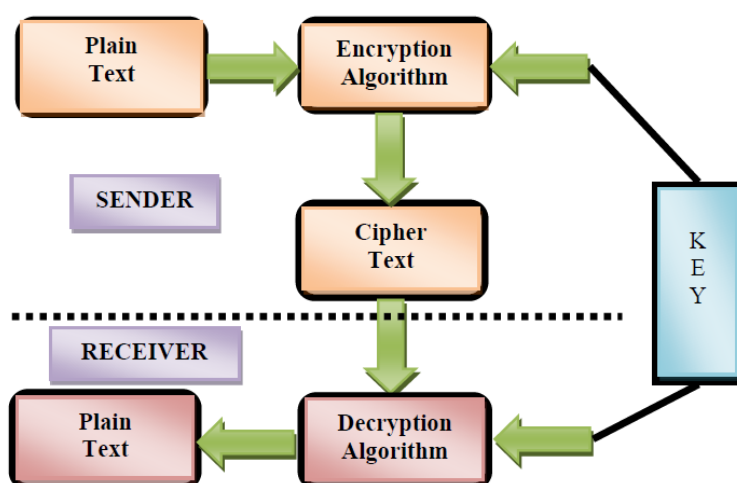


Figure 10.1: Block Diagram of Symmetric-key Cryptography

Two of the cryptographic approaches used in symmetric-key are stream cipher [229] and block cipher [267]. The stream cipher generates the keystream which is equal to the length of plain text. This would allow the encryption of the plain text. Block cipher creates smaller plain text of fixed blocks. These fixed blocks have the same length as the key. Then it encrypts the plain text of blocks using the same key. Steam cipher works better than block cipher when it comes to large data. It is quicker than the block cipher, and only small memory is needed. Figure 10.2 shows how a stream cipher operates.

10.1.2 Random Key Generation

Random numbers serve as keys in cryptography. Keys or random numbers are generated through a Random Number Generator (RNG) or a Pseudo Random Number

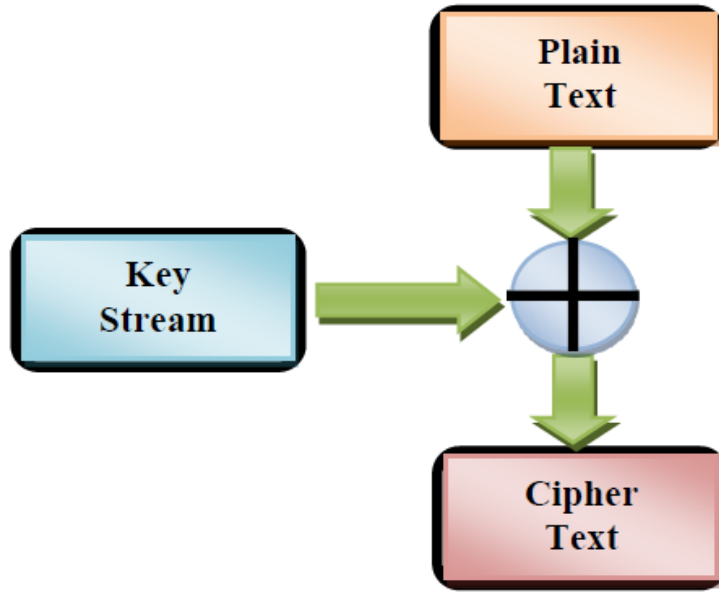


Figure 10.2: Block Diagram of Stream Cipher

Generator (PRNG) [28]. If the PRNG is compromised, then the cryptographic algorithm is compromised. Therefore, it is essential to use a PRNG which is hard to crack by any attacker. The randomness can be guaranteed by the evolutionary computing algorithms can be used in generating random numbers that are hard to break. This is because the PRNG is wrapped with evolutionary computing algorithms. In [328], there is an evolutionary computing algorithm-based PRNG (EAPRNG) proposed to provide a security layer around the main or mother PRNG. Four key generation algorithms (KGA) are proposed based on differential evolution (DE) [293], Harmony Search (HS), [104], and NSGA-II [78] namely NSGA-II Key Generation Algorithm (NSGA-II-KGA), IMHS+DE Key Generation Algorithm (IMHS+DE-KGA), IMHS Key Generation Algorithm (IMHS-KGA), and DE Key Generation Algorithm (DE-KGA). The advantage of NSGA-II is that it generates a Pareto front of uniformly distributed solutions under reasonable time complexity [78]. Harmony search is preferred over genetic algorithms based on the results presented by Geem et. al. in [104]. DE is preferred because of its high mutation rate [293].

10.1.3 Codes and Ciphers

Encoding requires a dictionary which is referred to as a code table for generated codes. The codes of the code table are used for decoding the message of the sender [292]. The sender generates the code table from the Mutated Huffman Tree (MHT) algorithm and encode the message. Then the recipient would decode the message through the same code table that was utilized by the sender.

10.2 Related Work in Cryptography

In cryptography, finding cryptographically sound boolean functions, changing hardware for RSA systems, cryptographic hash functions, block ciphers, designs of S-boxes and random sequences are some of the various uses of evolutionary computing according to Picek and Golub [242]. In [179], the uses the Computational Intelligence (CI) methods in solving problems from cryptography and cryptanalysis. This chapter aims to identify the CI methods that would solve the difficult problems of cryptanalysis.

In [134], three Bio Inspired Cryptographic approaches are proposed with the genetic algorithm (GA) [110], artificial immune system (AIS), and artificial neural network (ANN). The use of genetic algorithms in cryptography is provided in [79, 249]. It shows a methodology of using evolutionary computing algorithms in cryptology. In [140], a method on the usage of neural networks and genetic algorithms for generating a public key of cryptography is proposed. [47, 136] shows the uses of shifting in cryptography and computer security. In [211], the cryptographically sound S-boxes were created using genetic algorithms and hill climbing.

Swarm intelligence [67, 151] based key generation algorithm was created through an ant colony optimization [286]. In [285], the particle swarm optimization based key generation is discussed. Both utilized a single objective function that is dependent on the number of matches of keys in key streams with those in plain text. Authors in [284, 287] used the ant colony based key generation algorithm for the image encryption and in applications of cellular networks. In [23], a simulated annealing and genetic based cipher text generation algorithm for stream cipher is developed.

In [328], an EAPRNG is proposed that provides security around the main PRNG. In [123, 135], an ant system based RNG and genetic programming based PRNG are proposed. In [113], cellular automata based random number generation is proposed.

The comparison of the cellular automata and linear feedback shift registers is provided in [158].

10.3 Problem Definition

Works on evolutionary computing based methods in cryptanalysis is more than evolutionary computing based key generation. Therefore, we developed four evolutionary computing based key generation algorithms (KGAs) through NSGA-II, DE, IMHS + DE, and IMHS. All of these KGAs help in making the stream ciphers more secure. We also used a mutated Huffman tree- based encoding scheme wherein a compression is achieved apart from the encryption. The process is depicted in Figure 10.3.

10.4 Motivation

Our purpose is to propose an effective key generation technique by using evolutionary computing algorithms. Specifically, the evolutionary computing procedures allow the PRNG to be more secure because they hide the mother PRNG [328]. Therefore, the mother PRNG is protected by evolutionary computing algorithms and the attacker finds it hard to crack the mother PRNG.

Also, we have noticed that the evolutionary computing technique-based key generation with compression and encryption was not discussed before. For compression, we used the Mutated Huffman Coding technique [122] that gives additional security to the plain text.

Our third purpose is to use entropy for key generation so as to come up with a uniformly distributed key stream [71]. This makes it harder for the attacker to break the key as explained by the chaos theory and cryptography [71]. Therefore, we modelled the key generation problem as a bi-objective optimization with entropy as the second objective.

10.5 Contribution

The contribution of the chapter is as follows:

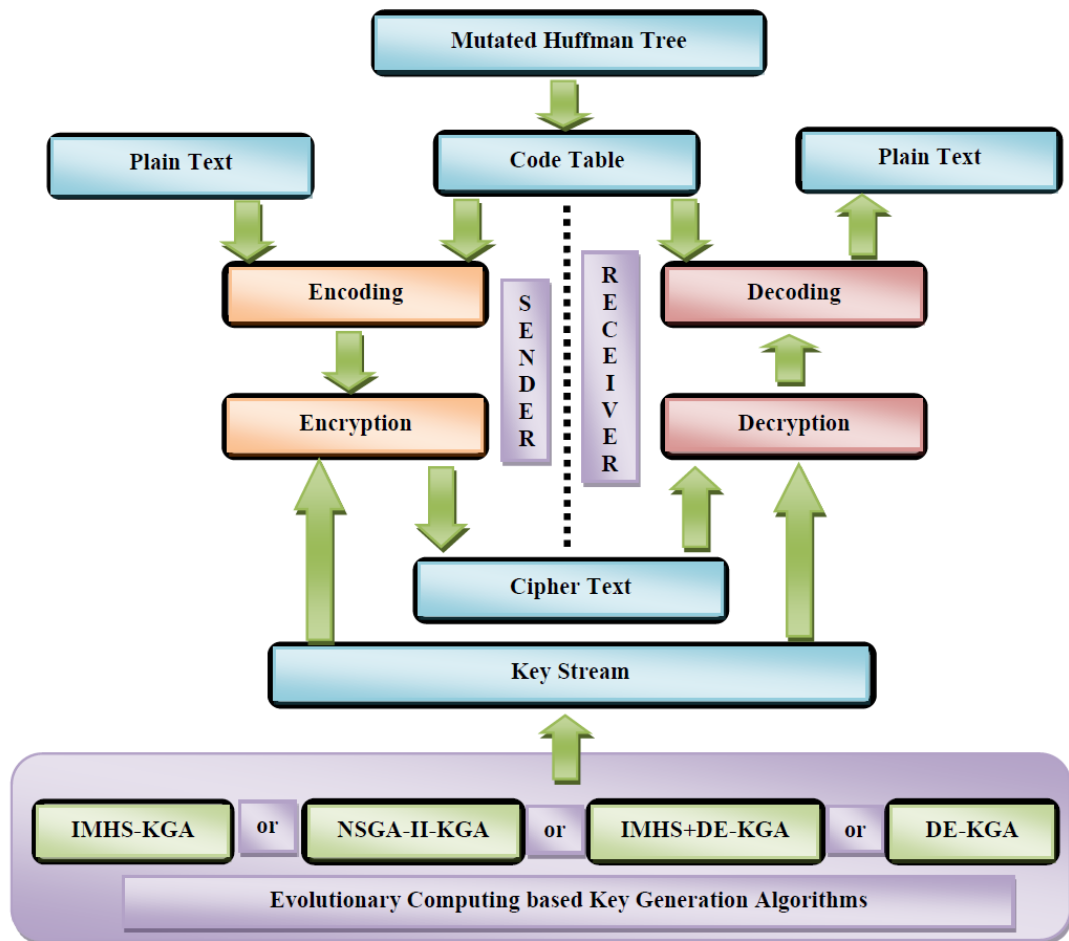


Figure 10.3: Block Diagram of the Process

-
1. This chapter provides the use of bi-objective evolutionary computing optimization algorithm viz., NSGA-II which has never been employed for key generation.
 2. The proposed method creates a Pareto-optimal solutions which give users additional keys in case some of the keys get compromised.
 3. Mutated Huffman coding is employed for compression.
 4. Entropy is used when the number of unique characters in the plain text are less than the key stream and when repetition of characters in the keystream is allowed.

10.6 Proposed Evolutionary Computing based Key Generation

The key generation is modeled as a bi-objective and single objective combinatorial optimization problem. The evolutionary computing algorithms are used to generate the keys in the single and bi-objective optimization framework. The evolutionary computing algorithms have a fitness function that is based on the keys generated. This chapter provides two objectives as the fitness functions. ASCII code values from 33 and 127 are used in the objective functions. The values 33 and 127 serve as the minimum and maximum bounds for the solution vector.

10.6.1 Objective Functions

In [286], the first objective function F_1 is the fraction of count of intersection of ASCII characters (AC) in the key stream (K) i.e. $AC \in K$ from the evolutionary computing algorithm and the unique ASCII characters (UAC) in the plain text (P) i.e. $UAC \in P$ to the length of the key stream. The objective of function F_1 lies between 0 and 1.

$$\text{Maximize } (F_1) = \frac{\text{Count}((AC \in K) \text{ intersection } (UAC \in P))}{\text{Length}(K)}$$

The second objective function is the entropy of the key stream generated and it should be equal to 1. This means that the key is random and hard to break. The objective function F_2 lies between 0 and 1.

$$\text{Maximize } (F_2) = - \sum_{i=1}^{\text{Length of Character Code Table}} x_i \log_{\text{keystream length}} x_i$$

$$x_i = \frac{\text{frequency of a particular character in key stream}}{\text{length of key stream}}$$

$$\text{Ideally } x_i = \frac{1}{\text{keystream length}}$$

$$\text{When all } x_i = \frac{1}{\text{key stream length}}$$

(uniformly distributed) then entropy will be equals to 1.

A bi-objective evolutionary optimization algorithm called NSGA-II is invoked to solve the problem, where the two objectives F_1 and F_2 are to be maximized.

10.6.2 Proof of Maximizing Entropy ensures Uniformly Distributed Keys

Let 'P' be the length of the key which is uniformly distributed. Then $1/P_i = n$ since $P_i = 1/n$ be the probability of each character in the key. It is known that the arithmetic mean (AM) is greater than the geometric mean (GM) based on the weighted AM-GM inequality. Therefore, the AM of probabilities of the key is greater than the GM of probabilities.

$$\begin{aligned} AM &\geq GM \\ \sum_{i=1}^n \left(P_i \times \frac{1}{P_i} \right) &\geq \left(\prod_{i=1}^n \frac{1}{P_i} \right)^{P_i} \\ \sum_{i=1}^n 1 &\geq \left(\prod_{i=1}^n \frac{1}{P_i} \right)^{P_i} \\ n &\geq \left(\prod_{i=1}^n \frac{1}{P_i} \right)^{P_i} \\ &\text{taking 'log' on both sides} \end{aligned}$$

$$\log n \geq \log \left(\prod_{i=1}^n \frac{1}{P_i} \right)^{P_i}$$

$$\log n \geq \sum_{i=1}^n P_i \log \frac{1}{P_i}$$

$$\log n \geq \sum_{i=1}^n -P_i \log P_i$$

$$\text{Max Entropy} \geq \text{Shannon's Entropy}$$

Therefore by Shannon's entropy, the keys are guaranteed to be uniformly distributed and are random. Thus, Shannon's entropy is an effective tool in measuring the uniformly distributed keys as shown in [264].

10.6.3 Threshold for the Fitness Function

If the fitness values for the two objectives are lower than the threshold 'T', then the key must be regenerated until higher fitness values are obtained. See Figure 10.4.

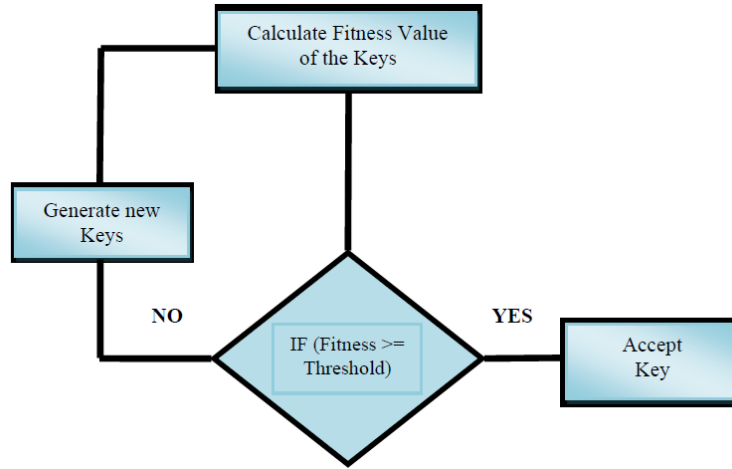


Figure 10.4: Block Diagram for the Threshold

The threshold 'T' for IMHS+DE, DE, and IMHS is 0.75 for F_1 . The thresholds of NSGA-II i.e. for the F_1 is set to 0.75, while for F_2 is set to 0.65. Thresholds are necessary for a faster convergence. There are cases wherein the second objective F_2 has less characters in the code table compared to the key length and therefore some of the

characters get repeated. Thus, the entropy F_2 will be lower than the ideal maximum '1'. That is why the threshold must be set lower than the maximum possible value of F_2 .

10.6.4 Combinatorial Explosion of the Problem

There are 95 ASCII characters starting from 33 to 127 in the ASCII code table that are utilized in the current problem. Two cases arise to produce a key stream by satisfying the two objectives:

1. The unique characters in the plain text are more in length compared to the key stream
2. The unique characters in the plain text are lesser than the key stream

In case 1, to generate a key stream of length 'r', a permutation of 95 characters taken 'r' at a time without repetition is needed. The value of 'r' ranges from 1 to 95. In the worst case, we need to search $\sum_{r=1}^{95} 95P_r$ key stream space.

In case 2, to generate a key stream of length 'r', a permutation of 95 characters taken 'r' at a time with repetition is needed. The value of 'r' ranges from 1 to 95. In the worst case, we need to search $\sum_{r=1}^{keystream\ length} 95^r$ key stream space.

10.6.5 Key generation using EC techniques

IMHS, DE, IMHS+DE and NSGA-II-based key generation process is presented in following subsections:

10.6.5.1 IMHS - Key Generation Algorithm

Improved Modified Harmony Search (IMHS) is run for maximum number of iterations. IMHS has one explicit parameters, i.e. 'par' and is fixed between 0 and 1. 'hmcr' parameter is linearly incremented from 0.7 to 0.9. (Please see Figure 10.5)

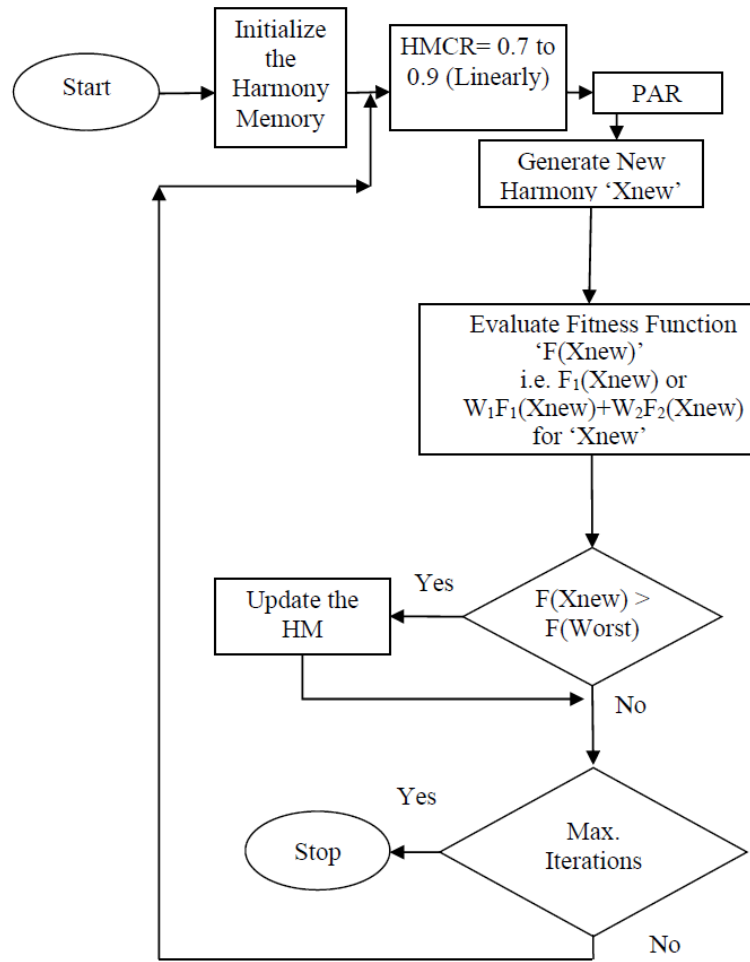


Figure 10.5: Flowchart for IMHS-Key Generation Algorithm

10.6.5.2 DE - Key Generation Algorithm

DE is used to search for the keys. Crossover rate and mutation in DE do the job of exploration and exploitation. Therefore, tuning of CR is an important task. DE replaces the current solution in the selection. 'CR' is the DE parameter called rate of crossover, which is the range of 0 and 1. Also, 'F' is another DE parameter called differential weight, which is in the range of 0 and 2. However, here we fixed F to be '1' because it is an integer optimization problem. DE is run for maximum number of iterations. (Please see Figure 10.6)

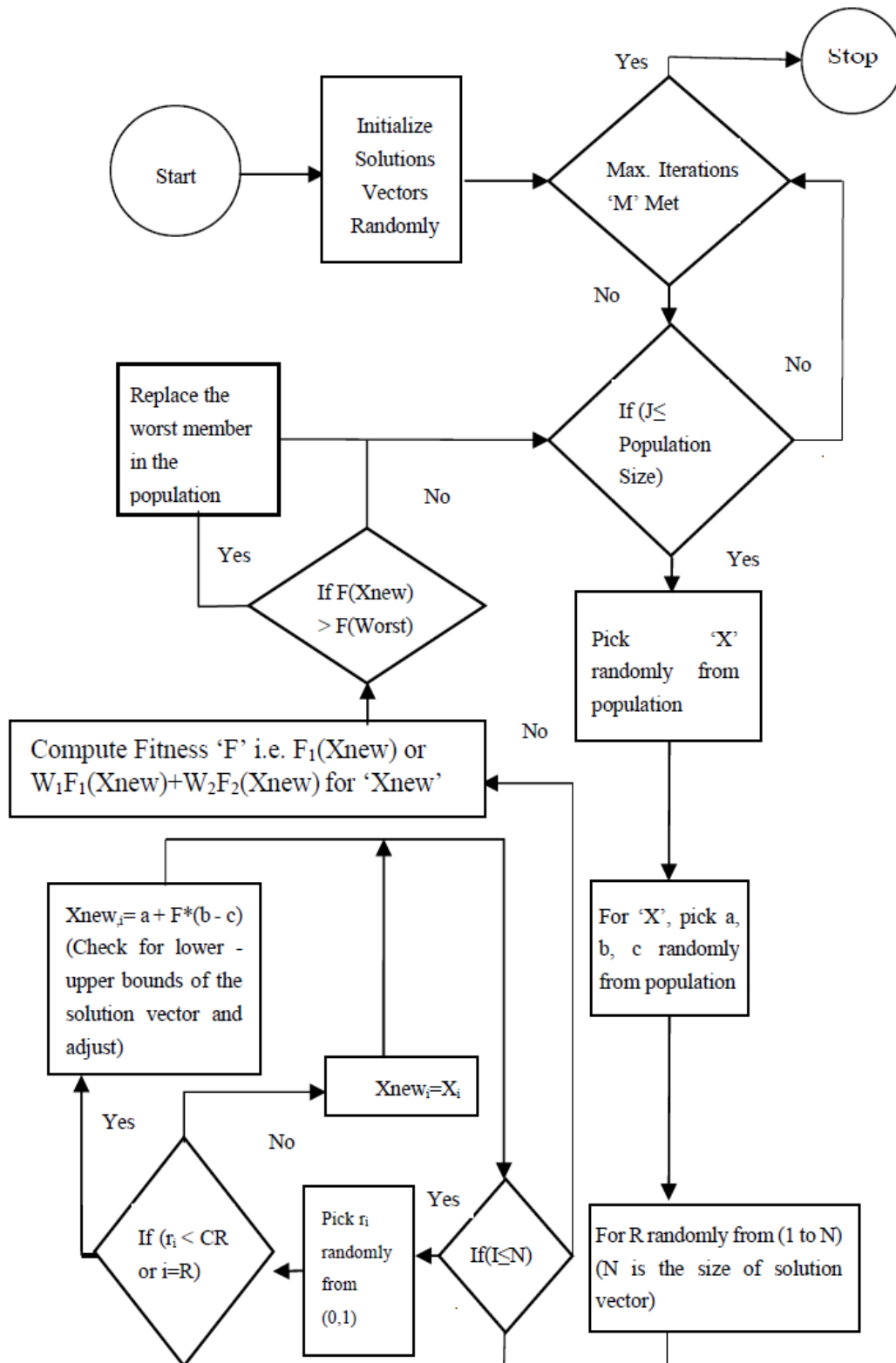


Figure 10.6: Flowchart for DE-Key Generation Algorithm

10.6.5.3 IMHS+DE - Key Generation Algorithm

The IMHS+DE key generation is performed in two phases. IMHS is run in first phase and DE is run in second phase. The solutions from first phase are passed to second phase. IMHS-based key generation algorithm is run for half number of iterations and DE-based key generation algorithm is run for half number of iterations. This is called IMHS+MDE-based key generation algorithm. IMHS+MDE has three explicit parameters, i.e. ‘par’, ‘CR’ and ‘F’. These parameters are fixed between 0 and 1. ‘hmcr’ parameter is linearly incremented from 0.7 to 0.9. ‘F’ is fixed at 1 as the problem is modeled as an integer optimization problem.

10.6.5.4 NSGA-II - Key Generation Algorithm

Please see Figure 10.7 for NSGA-II-key generation algorithm

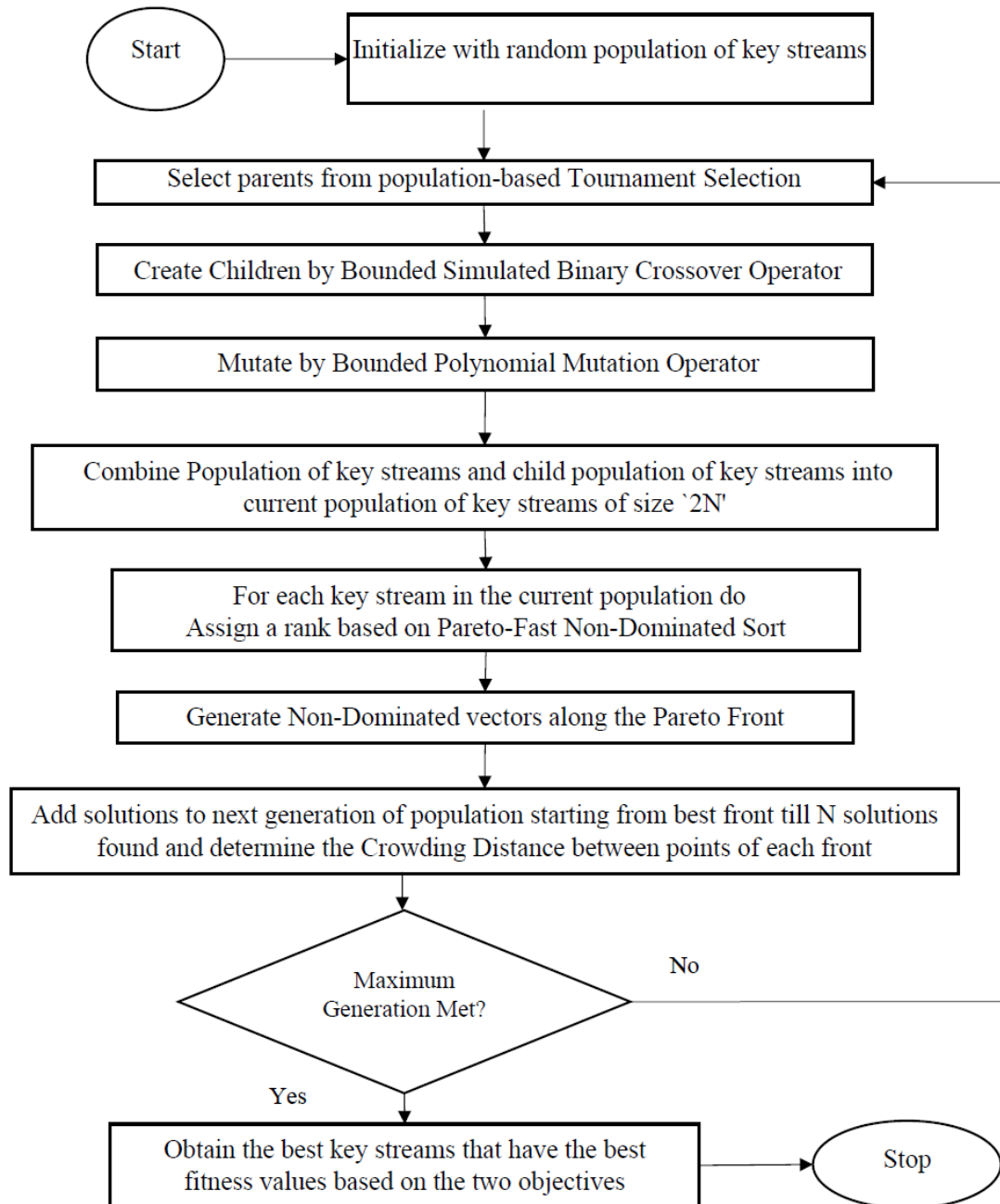


Figure 10.7: Flowchart for NSGA-II-Key Generation Algorithm

10.7 Mutated Huffman Code Table

10.7.1 Huffman Tree Coding

The Huffman coding tree [133] has two branches. The left branch is ‘0’ while the right branch is ‘1’. Under the Huffman tree encoding technique, a table is generated from the probabilities of the frequency of occurrence wherein fewer bits are used compared to the regular encoding techniques. The frequency of occurrence is calculated for every character or symbol for all the frequencies. Then the values are arranged from the highest to lowest and a tree is created. The symbol that has the highest frequency contains minimal code word length. Therefore, compression can also be achieved in this process.

10.7.2 Mutated Huffman Tree (MHT) Coding

If the labels of two branches of the inner codes at same level are exchanged, the probability of the codes and the length of the codes do not vary. This is the reason why the Huffman tree can be mutated randomly at same level [122].

10.8 Key Streams Generation

In cryptography, one key idea is that key stream length has to be ideally equal to the message length [30]. However, under practical circumstances, this is not possible because key stream and plain text lengths are never the same. But, still the key streams can be produced by adding a predetermined value (PV) to the keystream and the process is repeated until the length of the plain text (P).

$$PV = Integer \left(\frac{length(P)}{2} \right)$$

$$K \epsilon B_i = K \epsilon B_{i-1} + PV \quad for \ i > 2$$

PV is added till there is no resemblance of

previous encoded block values

B_1 is the block of the initial keystream generated. Where $K \epsilon B_i$ is the keystream of the B_i block generated by adding a predetermined value to the key stream of B_{i-1}

10.9 Procedure and Case Studies

10.9.1 Example Procedure

In the following, we illustrate the proposed method using a simple example.

Let us use “goodperson1199” as an plain text. There are ten unique characters in this plain text which are g, o, d, p, e, r, s, n, 1, and 9. Therefore, the mutated Huffman code table has ten unique entries. The keystream is obtained from an evolutionary computing-based key generation algorithm. Then, the Huffman tree algorithm was utilized to make the Huffman coding tree of the unique characters in ‘goodperson1199’. We randomly mutate the inner nodes of the tree. For the example given, let the key streams produced by key generation algorithms are K_1 : “dopggk2” and K_2 : “god199p”.

Looking at NSGA-II, the objective functions value of F_1 (0.714) and the objective function value of F_2 (0.612) did not cross the threshold for K_1 . However, the values of F_1 (0.857) and F_2 (0.755) crossed the threshold values for K_2 . Therefore, K_2 is recognized as the key. The same procedure is followed for IMHS+DE, DE, IMHS key generation algorithms that have F_1 as the objective function.

Once the key generation is done, the Huffman tree is created based on the occurrence of the unique characters in the plain text and this Huffman tree is mutated randomly. Keystream equivalent to plain text length is generated by adding a predetermined value to the repeated keystream. The keystream generated from the procedure explained above and the plain text encoded using the mutated Huffman code table is illustrated in Figure 10.8. The Huffman tree is mutated at the level. For example, it is mutated at nodes 6 and 7. If the character in the key stream is not in the Huffman table, it will be replaced with ASCII value. XOR operation is performed for encryption (see Table 10.1). The receiver decrypts with the secret key by the XOR operation. Then, the decrypted messages are decoded through the mutated Huffman code table.

10.9.2 Case Study

The message for the case study is taken from the UCI Machine learning repository. It is taken from “Mini NewsGroups – Sci.crypto” of “Twenty NewsGroups” dataset [8] with text file number “14989” (shown in Figures 10.9 and 10.10). Two cases presented above are tested (refer to section 10.6.4). The first case has a key length of 15 and the second case has a key length of 68. Both cases have 64 unique characters in the

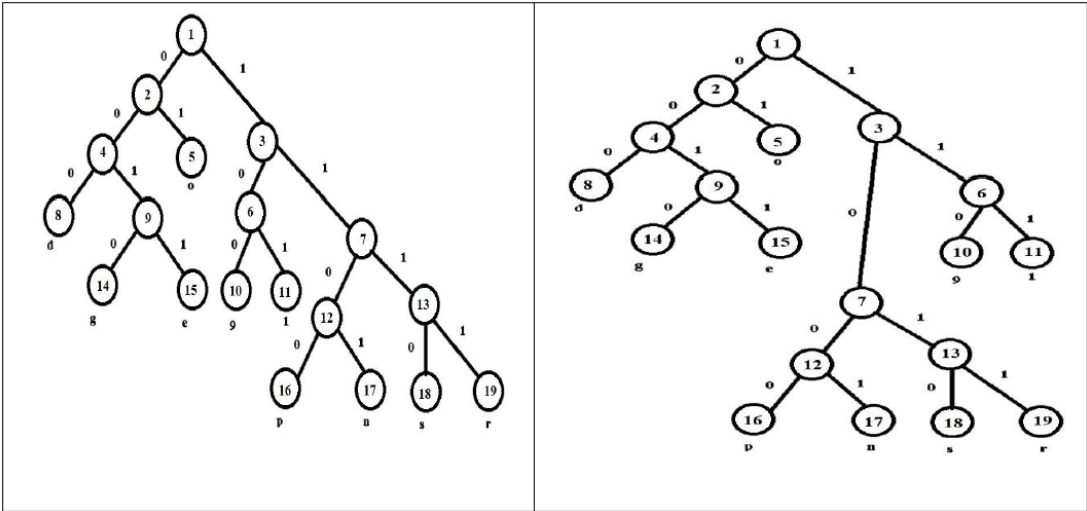


Figure 10.8: Huffman Tree for the “goodperson1199” and Mutated Huffman Tree mutated at nodes 6 and 7

Table 10.1: Encoding and Encryption Example

Key	Encoded	Adding	Encoded	Cipher
Stream	Key	PV	Plain	Text
Stream	Stream		Text	
“g”	2	2	2	0
“o”	1	1	1	0
“d”	0	0	1	1
“1”	7	7	0	7
“9”	6	6	8	14
“9”	6	6	3	5
“p”	8	8	11	3
“g”	2	16	10	26
“o”	1	15	1	14
“d”	0	14	9	7
“1”	7	21	7	18
“9”	6	20	7	19
“9”	6	20	6	18
“p”	8	22	6	16

character code table. The first case has more unique characters in the code table than the key length. Therefore, the single objective KGAs were used. In the second case , there are fewer characters in the code table than the key length. So, the bi-objective NSGA-II-KGA was used. Refer to tables 10.2 and 10.3. The ASCII characters “e”, “g”, “n” , “x”, “s”, “a”, “k”, “r”, “u” are repeated in case-2. The second objective has entropy less than 1.

```

Path:
cantaloupe.srv.cs.cmu.edu!crabapple.srv.cs.cmu.edu!fs7.ece.cmu.edu!europa.eng.gtefs
d.com!news.ans.net!cmcl2!arizona!ho
From: ho@cs.arizona.edu (Hilarie Orman)
Newsgroups: sci.crypt
Subject: Re: Licensing of public key implementations
Message-ID: <36179@optima.cs.arizona.edu>
Date: 6 Apr 93 00:12:20 GMT
References: <930403152101.890833@DOCKMASTER.NCSC.MIL>
Sender: news@cs.arizona.edu
Organization: U of Arizona, CS Dept, Tucson
Lines: 6
With regard to your speculations on NSA involvement in the creation of PKP, I find
that it fails the test of Occam's butcher knife. Never attribute to conspiracy
what can be explained by forthright greed.
Hilarie Orman

```

Figure 10.9: Sample Text Snippet

```

Plain Text:
Path:cantaloupe.srv.cs.cmu.edu!crabapple.s
rv.cs.cmu.edu!fs7.ece.cmu.edu!europa.eng.g
tefsd.com!news.ans.net!cmcl2!arizona!ho
From:ho@cs.arizona.edu(HilarieOrman)Newsgroup
s:sci.cryptSubject:Re:Licensingofpublickey
implementationsMessage-ID:<36179@optima.cs
.arizona.edu>Date:6Apr9300:12:20GMTReferen
ces:<930403152101.890833@DOCKMASTER.NCSC.M
IL>Sender:news@cs.arizona.eduOrganization:
UofArizonaCSDeptTucsonLines:6Withregardtoy
ourspeculationsonNSAinvolvementinthecreati
onofPKPIfindthatitfailsthetestofOccam'sbut
cherknife.Neverattributetoconspiracywhatca
nbeexplainedbyforthrightgreed.HilarieOrman

Unique Character in Plain Text used to prepare Huffman Code Table (64 in length):
- : ! . ' ( ) @ < > 0 1 2 3 4 5 6 7 8 9 a A b c C d D e E f F g G h H i I j k K l
L m M n N o O p P r R s S t T u U v w W x y z

```

Figure 10.10: Plain Text and Unique ASCII Characters

Table 10.2: Results of Case Study (Case-1)

Single Objective: IMHS+DE-KGA _{WS}	Single Objective: DE-KGA _{WS}	Single Objective : IMHS-KGA _{WS}
Memory Used in Bytes: 51049384 bytes	Memory Used in Bytes: 50536640 bytes	Memory Used in Bytes: 51189344 bytes
Key Length: 15	Key Length: 15	Key Length: 15
key: z 6 2 L : d > O @ I v 3 E p '	key: 0 l - p u W D < 7 E f K ' @ d	key: K 9 R n 7 M j 6 S 5 z . i I P
Optimum: 1	Optimum: 1	Optimum: 1
Encryption Time: 2.106994 secs	Encryption Time: 1.148825 secs	Encryption Time: 1.487555 secs
Decryption Time: 0.003 secs	Decryption Time: 0.006 secs	Decryption Time: 0.004 secs

Table 10.3: Results of Case Study (Case-2)

NSGA-II – KGA
Memory Used in Bytes: 31196040 bytes
Key Length: 68
Key: 3 e 6) i . : v D 8 ' g w z u c < N R t 5 U o g S e k j m n x L s f T r H s I M 7 F O R a a p P (h x - 9 k b l l d u ! n E y G W 2 0 4
Optimum 1: 1
Optimum 2: 0.956516222068868
Encryption Time: 1.15626 mins
Decryption Time: 0.0008742809 secs

10.10 Experimental Results

The results are obtained on the processor with CPU frequency of 1.80 GHz and 860 MHz. The processor configuration was decreased so that it would be equal to the above-mentioned frequency for the comparison with the results gathered in [286]. CPU power utility of Fedora 23 operating system was used to scale down the CPU frequency. The code was written in R tool as a single thread. The evolutionary computing algorithms utilized are stochastic. Therefore, an average of 30 runs of various plain text lengths were taken. In Figures 10.11, 10.12 and Tables 10.4, 10.5 and 10.6, OBJ_{SO} represents the single objective function (i.e. F_1) based KGAs' which are IMHS+DE-KGA, DE-KGA, IMHS-KGA. OBJ_{WS} represents the weighted sum (i.e. $W_1F_1 + W_2F_2$) based KGAs' which are IMHS+DE-KGA_{WS}, DE-KGA_{WS}, and IMHS-KGA_{WS}. OBJ_{BO} represents bi-objective (i.e. F_1 and F_2) based KGA which is NSGA-II-KGA.

Experiments were performed in 2 cases. First case has 22 unique characters and the second case has 26 unique characters of the mutated Huffman code table. For each case, there are five different plain texts with lengths 21600, 7200, 6912, 4800, and 3600. So, there are 5 plain texts for the two cases of unique characters. The keys of lengths are 68, 41, 30, and 15 which were also generated by the evolutionary computing based KGAs and utilized in the process of encryption. For each case of unique characters for five plain texts, the algorithm is run 30 times which resulted in 150 experiments. Therefore, the average of the 150 experiments was computed. Results are indicated in Tables 10.4 and 10.6. The NSGA-II-KGA algorithm generated keys with the two objective functions F_1 and F_2 respectively, where, both F_1 and F_2 are maximized. Based on the results, it is observed that as the length of the key increases, it becomes more difficult to get the optimal value.

Also, entropy is utilized as the objective function F_2 . As there are repetitions allowed for the keys of length 68. This makes the entropy low and so the objective function F_2 is also low. Therefore, the threshold of F_2 is set as 0.65. Once the keys are distributed uniformly with the second objective function F_2 , this would add security to the key and makes it harder for the attacker to break the key. It is also noted that although there are two objectives, the NSGA-II has the best performance among the four proposed KGA's with respect to optimum values of F_1 . However, NSGA-II also consumes the more time.

Wilcoxon Signed Rank test is performed between the KGA with the best average fitness (marked in bold) and the rest. This is shown in Tables 10.4 and 10.6 in bold. This shows that the means of the fitness values of KGAs differ or not. If the p-value is lower than 0.05, then the null hypothesis would not be accepted. This only shows that the two compared sets of values differ statistically. On the contrary, a p-value that is higher than 0.05 then the null hypothesis is accepted, proving that the two compared pair of values do not differ statistically. The tables show the p-value below the average fitness value. Table 10.5 shows the improvement of the entropy of the keys. F_1 is the only objective function used in a single objective framework. Thus, we computed the entropy separately to know if the entropy has improved in the bi-objective framework. The results are illustrated in Figures 10.11 and 10.12. As entropy increases, keys are distributed uniformly, thereby making them tough to break.

The improvement of the entropy of the keys in bi-objective framework over single objective framework is presented in Table 10.5. In single objective framework, only F_1 is utilized. Therefore, we computed entropy separately to check whether there is any improvement achieved in entropy in the bi-objective framework. As seen in Table 10.5, there is a marginal improvement of entropy over the single objective framework. Figures 10.11 and 10.12 illustrate the same. By increasing the entropy, the keys are more uniformly distributed and harder to crack. Table 10.5 and Figures 10.11, 10.12 present the comparison between the F_2 (Entropy) values, (i.e. OBJ_{BO}) generated by NSGA-II-KGA and F_2 (Entropy) values, (i.e. OBJ_{SO}), computed in the single objective KGAs.

Weighted sum based objective function (i.e. OBJ_{ws}) is computed by multiplying weights W_1 and W_2 with the objective functions F_1 and F_2 respectively. Weighted Sum (WS) based key generation using two objective functions (F_1 and F_2) for IMHS+DE-KGA_{WS}, DE-KGA_{WS}, and IMHS-KGA_{WS} are presented in Table 10.6. The DE-KGA_{WS} performed better than other weighted sum based KGAs' in terms of the objective value computation. Bi-objective and scalar optimization framework of KGAs (i.e. especially NSGA-II-KGA and DE-KGA_{WS}) performed better than single objective framework in terms of the objective function values.

Table 10.4: Average optimal objective function values of OBJ_{BO} and OBJ_{SO}

Character Code Table Size	Key Length	F_1 (Avg) ≥ 0.75 (p-value)	F_2 (Avg) ≥ 0.65 (p-value)	F_1 (Avg) ≥ 0.75 (p-value)	F_1 (Avg) ≥ 0.75 (p-value)	F_1 (Avg) ≥ 0.75 (p-value)
		NSGA-II-KGA				
1.80 GHz CPU Frequency						
22	15	0.8968 (0.1871)	0.8519	0.8448 (0.0)	0.8791 (0.0)	0.9075
22	30	0.9806	0.8505	0.8742 (0.0)	0.8491 (0.0)	0.8248 (0.0)
22	41	0.9621	0.7690	0.8886 (0.0)	0.8684 (0.0)	0.8009 (0.0)
22	68	0.9712	0.6919	0.8032 (0.0)	0.8728 (0.0)	0.7839 (0.0)
26	15	0.9444	0.9125	0.8853 (0.0)	0.9137 (0.0)	0.9088 (0.0)
26	30	0.9926	0.8929	0.9055 (0.0)	0.8904 (0.0)	0.8673 (0.0)
26	41	0.9835	0.8214	0.8730 (0.0)	0.8925 (0.0)	0.8440 (0.0)
26	68	0.9901	0.7392	0.8560 (0.0)	0.9050 (0.0)	0.8180 (0.0)
860 MHz CPU Frequency						
22	68	0.9849	0.6876	0.8173 (0.0)	0.8902 (0.0)	0.7782 (0.0)
26	15	0.9617	0.9277	0.8751 (0.0)	0.9057 (0.0)	0.9075 (0.0001)
26	30	0.9482	0.8406	0.8986 (0.0)	0.886 (0.0)	0.85 (0.0)

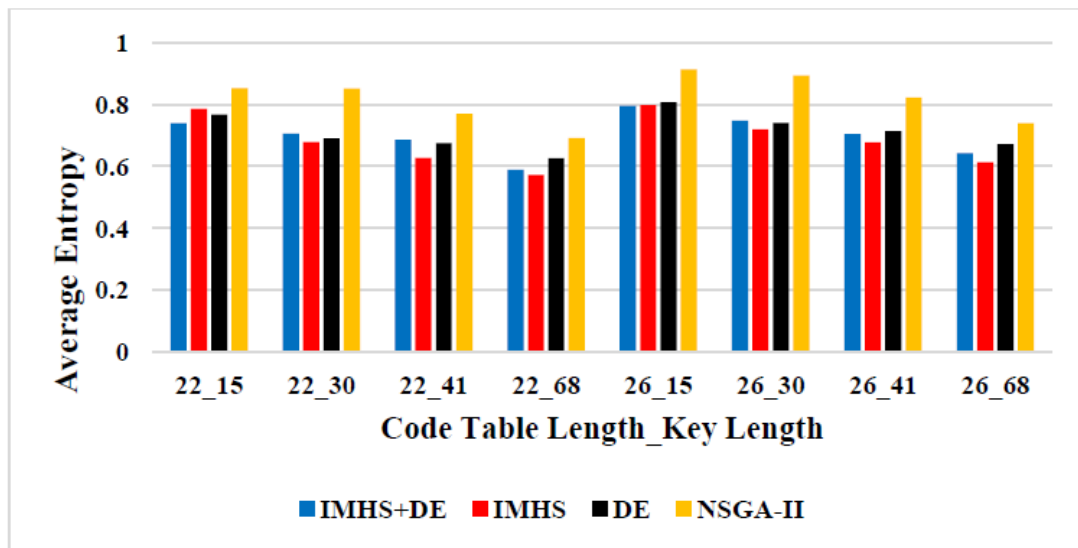


Figure 10.11: Entropy values generated by OBJ_{BO} vs computed from the OBJ_{SO} (1.80 GHz)

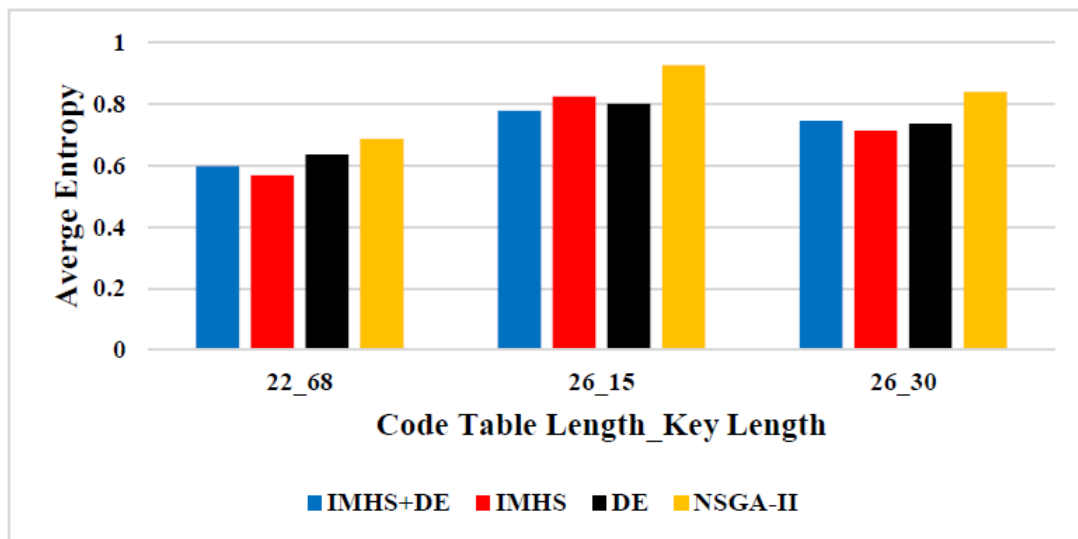


Figure 10.12: Entropy values generated by OBJ_{BO} vs computed from the OBJ_{SO} (860 MHz)

Table 10.5: Comparison of Entropy values generated by OBJ_{BO} vs computed from the OBJ_{SO}

Character Code Table Size	Key Length	IMHS+DE (F_2)	IMHS (F_2)	DE (F_2)	NSGA-II (F_2)	Percentage Improvement Over IMHS+DE (KGA)	Percentage Improvement Over IMHS (KGA)	Percentage Improvement Over DE (KGA)
Average Entropy (1.80 GHz CPU Frequency)								
22	15	0.7389	0.7851	0.7660	0.8519	15.2957%	8.5040%	11.2092%
22	30	0.7056	0.6785	0.6903	0.8505	20.5302%	25.3384%	23.2048%
22	41	0.6856	0.6269	0.6750	0.7690	12.1651%	22.6670%	13.9279%
22	68	0.5895	0.5720	0.6261	0.6919	17.3796%	20.9613%	10.5059%
26	15	0.7944	0.7970	0.8070	0.9125	14.8737%	14.4934%	13.0682%
26	30	0.7470	0.7184	0.7404	0.8929	19.5290%	24.2861%	20.5952%
26	41	0.7042	0.6774	0.7125	0.8214	16.6430%	21.2568%	15.2835%
26	68	0.6425	0.6135	0.6710	0.7392	15.0604%	20.4867%	10.1730%
Average Entropy (860 MHz CPU Frequency)								
22	68	0.5972	0.5680	0.6363	0.6876	15.1363%	21.0608%	8.0637%
26	15	0.7781	0.8249	0.8019	0.9277	19.2247%	12.4670%	15.6849%
26	30	0.7461	0.7135	0.7363	0.8406	12.6585%	17.8051%	14.1644%

Table 10.6: Average of optimal objective values of OBJ_{WS}

Character Code Table Size	Key Length	$\frac{W_1 F_1}{W_2 F_2} \geq 0.6$	$\frac{W_1 F_1}{W_2 F_2} \geq 0.6$	$\frac{W_1 F_1}{W_2 F_2} \geq 0.6$	$\frac{W_1 F_1}{W_2 F_2} \geq 0.6$	$\frac{W_1 F_1}{W_2 F_2} \geq 0.6$	$\frac{W_1 F_1}{W_2 F_2} \geq 0.6$	$\frac{W_1 F_1}{W_2 F_2} \geq 0.6$	$\frac{W_1 F_1}{W_2 F_2} \geq 0.6$	$\frac{W_1 F_1}{W_2 F_2} \geq 0.6$	$\frac{W_1 F_1}{W_2 F_2} \geq 0.6$	$\frac{W_1 F_1}{W_2 F_2} \geq 0.6$	$\frac{W_1 F_1}{W_2 F_2} \geq 0.6$
		(Avg) $W_1=0.5$ $W_2=0.5$ (p-value)	(Avg) $W_1=0.25$ $W_2=0.75$ (p-value)	(Avg) $W_1=0.75$ $W_2=0.25$ (p-value)	(Avg) $W_1=0.5$ $W_2=0.5$ (p-value)	(Avg) $W_1=0.25$ $W_2=0.75$ (p-value)	(Avg) $W_1=0.75$ $W_2=0.25$ (p-value)	(Avg) $W_1=0.5$ $W_2=0.5$ (p-value)	(Avg) $W_1=0.25$ $W_2=0.75$ (p-value)	(Avg) $W_1=0.75$ $W_2=0.25$ (p-value)	(Avg) $W_1=0.5$ $W_2=0.5$ (p-value)	(Avg) $W_1=0.25$ $W_2=0.75$ (p-value)	(Avg) $W_1=0.75$ $W_2=0.25$ (p-value)
1.80 GHz CPU Frequency													
22	15	0.7433 (0.0)	0.7217 (0.0)	0.7566 (0.0)	0.8330 (0.0)	0.8098 (0.0)	0.8499 (0.0004)	0.8421 (0.0)	0.8311 (0.0)	0.8727			
22	30	0.7418 (0.0)	0.6944 (0.0)	0.7731 (0.0)	0.7757 (0.0)	0.7489 (0.0)	0.8119	0.7586 (0.0)	0.7232 (0.0)	0.8017 (0.0193)			
22	41	0.7540 (0.0)	0.7070 (0.0)	0.7807 (0.0)	0.7892 (0.0)	0.7412 (0.0)	0.8258	0.7475 (0.0)	0.7036 (0.0)	0.7883 (0.0)			
22	68	0.6593 (0.0)	0.6121 (0.0)	0.7110 (0.0)	0.7680 (0.0)	0.6974 (0.0)	0.8112	0.6978 (0.0)	0.6528 (0.0)	0.7727 (0.0)			
26	15	0.7917 (0.0)	0.7717 (0.0)	0.8014 (0.0)	0.8753 (0.0)	0.8614 (0.0)	0.8909 (0.0)	0.9002 (0.0)	0.8695 (0.0)	0.9262			
26	30	0.5935 (0.0)	0.5803 (0.0)	0.6126 (0.0)	0.8231 (0.0)	0.8022 (0.0)	0.8497	0.7957 (0.0)	0.7637 (0.0)	0.8400 (0.0338)			
26	41	0.7368 (0.0)	0.7059 (0.0)	0.7673 (0.0)	0.8244 (0.0)	0.7877 (0.0)	0.8638	0.7871 (0.0)	0.7524 (0.0)	0.8435 (0.0)			
26	68	0.7027 (0.0)	0.6640 (0.0)	0.7569 (0.0)	0.7981 (0.0)	0.7498 (0.0)	0.8510	0.7386 (0.0)	0.6910 (0.0)	0.7943 (0.0)			
860 MHz CPU Frequency													
22	68	0.6570 (0.0)	0.6123 (0.0)	0.7106 (0.0)	0.7579 (0.0)	0.7051 (0.0)	0.8312	0.7088 (0.0)	0.6530 (0.0)	0.7658 (0.0)			
26	15	0.7836 (0.0)	0.7762 (0.0)	0.8016 (0.0)	0.8727 (0.0)	0.8584 (0.0)	0.8867 (0.0)	0.8975 (0.011)	0.8661 (0.0)	0.9094			
26	30	0.5964 (0.0)	0.5863 (0.0)	0.6162 (0.0)	0.8262 (0.0)	0.7890 (0.0)	0.8596	0.7985 (0.0)	0.7600 (0.0)	0.8387 (0.0)			

10.10.1 Encryption Times Compared

Tables 10.7, 10.8 and 10.9 presents the results of encryption in seconds as in [286] based on which the proposed key generation algorithms are compared. IMHS +DE-KGA and DE-KGA yielded better results for key generation used for encryption compared to other algorithms and the one proposed in [286]. IMHS-KGA did not perform well as the key generation length increases. The main difference between NSGA-II-KGA and other algorithms proposed in this chapter is that NSGA-II-KGA is a bi-objective evolutionary algorithm and others are single-objective evolutionary algorithms.

Encryption times are compared in Tables 10.8 and 10.9 with 1.80 GHz and 860 MHz of processor speeds and we can see that though NSGA-II-KGA encryption time is high for 3600, 4800 and 7200 lengths of plain text but it performed extremely well for higher plain text lengths. Also, a harder objective function F_2 is included to NSGA-II-KGA to increase the security of the key. Hence, it consumes slightly more time to cross the thresholds set for the two objective functions. Since it works with two objective functions, its results cannot be strictly compared with those of the other proposed methods. However, the results of IMHS+DE-KGA, DE-KGA, and IMHS-KGA are compared with those of [286]. It is clear from the Tables 10.7, 10.8 and 10.9 that the IMHS+DE-KGA, DE-KGA, and IMHS-KGA outperformed the ant colony based key generation algorithm in [286] by a significant margin. We can also see that within the proposed techniques, DE-KGA outperformed IMHS+DE-KGA and IMHS-KGA as the plain text length increases.

In Table 10.7, we have compared the key lengths of 30, 41, 68, plain text lengths of 3600, 4800, 6912, 7200, 21600 and code table size of 22 of the proposed key KGAs with the AKGA [286] at 1.80 GHz.

In Tables 10.8 and 10.9, we compared with code table size of 26 apart from 22 at 1.80 GHz. In Table 10.8 and 10.9, we compared the proposed KGAs with 860 MHz, at key length 68 and code table size of 22. In Table 10.8 and 10.9, we compared the same proposed KGAs with 860 MHz, at key length 30 and code table size of 26. Other parameters are not compared as the results of AKGA are not available.

Table 10.7: Results of Encryption Time in seconds

Plain Text Length	Code Table Size (22)											
	Key Lengths											
	30	41	68	30	41	68	30	41	68	30	41	68
	AKGA [286]			IMHS+DE-KGA			DE-KGA			IMHS-KGA		
3600	2	3	4	0.77	1.42	3.00	0.61	1.11	2.90	0.73	1.37	6.70
4800	4	5	6	0.77	1.44	3.15	0.61	1.10	2.84	0.71	1.37	6.74
6912	8	9	10	0.78	1.59	2.98	0.62	1.12	2.99	0.71	1.37	6.65
7200	9	10	11	0.78	1.52	2.95	0.62	1.13	2.85	0.75	1.37	6.68
21600	68	70	125	0.84	1.52	3.39	0.69	1.18	2.93	0.80	1.48	6.84

Table 10.8: Results of Encryption Time in seconds-(a)

Key Length (30)	Plain Text Length			Key Length (68)			Plain Text Length			Key Length (30)			Plain Text Length				
Code Table Size	3600	6912	7200	21600	Code (22)	Table (26)	3600	4800	6912	7200	Code (26)	Table (30)	3600	4800	6912		
	Encryption Time in Seconds				Processor Speeds			Encryption Time in Seconds				Processor Speeds			Encryption Time in Seconds		
AKGA [286]																	
22	2	3	7	67	860 MHz		18	26	46	55	860 MHz		12	22	35		
26	3	8	9	68	1.80 GHz		4	6	10	11	1.80 GHz		3	4	8		
IMHS+DE-KGA																	
22	0.77	0.78	0.78	0.84	860 MHz		6.35	6.35	6.75	6.71	860 MHz		1.59	1.82	1.68		
26	0.77	0.78	0.78	0.86	1.80 GHz		3.00	3.15	2.98	2.95	1.80 GHz		0.77	0.75	0.78		
DE-KGA																	
22	0.61	0.62	0.62	0.69	860 MHz		6.31	6.34	6.46	6.58	860 MHz		1.35	1.37	1.38		
26	0.61	0.64	0.63	0.70	1.80 GHz		2.90	2.84	2.99	2.85	1.80 GHz		0.61	0.68	0.64		
IMHS-KGA																	
22	0.73	0.71	0.75	0.80	860 MHz		15.32	15.94	15.83	14.71	860 MHz		1.40	1.34	1.39		
26	0.70	0.74	0.72	0.84	1.80 GHz		6.70	6.74	6.65	6.68	1.80 GHz		0.70	0.73	0.74		

Table 10.9: Results of Encryption Time in seconds-(b)

Key Length (30)	Plain Length	Text Length	Key Length (68)	Plain Length	Text Length	Key Length (30)	Plain Length	Text					
Processor Speed (1.80 Ghz)	3600	6912	7200	21600	Code Table Size (22)	3600	4800	6912	7200	Code Table Size (26)	3600	4800	6912
	Encryption Time in Seconds		Processor Speeds		Encryption Time in Seconds		Processor Speeds		Encryption Time in Seconds				
NSGA-II-KGA													
22	5.64	5.78	5.99	5.83	860 MHz	258.9	261.7	255.5	261.7	860 MHz	6.93	8.01	6.93
26	5.83	5.94	5.81	5.89	1.80 GHz	87.7	89.4	88.5	89.1	1.80 GHz	5.83	6.21	5.94
IMHS+DE-KGA _{WS} (W ₁ =0.5 and W ₂ =0.5)													
22	1.19	1.77	1.69	1.78	860 MHz	4.51	4.41	4.90	4.87	860 MHz	0.80	0.66	0.91
26	0.53	0.93	0.98	1.13	1.80 GHz	4.06	3.99	4.55	4.31	1.80 GHz	0.53	0.52	0.93
DE-KGA _{WS} (W ₁ =0.5 and W ₂ =0.5)													
22	1.14	1.41	1.45	1.08	860 MHz	4.62	4.78	4.81	4.76	860 MHz	1.30	1.29	1.22
26	1.07	1.39	1.44	1.14	1.80 GHz	4.26	4.19	4.26	4.28	1.80 GHz	1.07	1.20	1.39
IMHS-KGA _{WS} (W ₁ =0.5 and W ₂ =0.5)													
22	1.21	1.49	1.97	1.38	860 MHz	11.72	13.17	10.18	9.72	860 MHz	1.52	1.55	1.58
26	1.21	1.52	1.63	1.40	1.80 GHz	8.48	9.72	8.77	8.33	1.80 GHz	1.21	1.50	1.52

10.10.2 Compression Ratio

The compression percentage was computed in order to check the amount of compression obtained from basic Huffman Coding and Mutated Huffman Coding for text files “paper1”, “paper2”, “paper3”, “paper4”, “paper5” and “paper6” in the standard compression corpus called Calgary [9]. Newline characters (“\n”), NULL characters (“\0”) and tabs (“\t”), which are called escape sequences were removed. It is necessary to remove the escape sequences since these are not part of the plain text characters of the ASCII codes starting from 33 to 127 (ASCII characters include various special characters with space included as one of the characters).

In Calgary Corpus, text files “paper1” has 51119 characters with 92 unique characters, “paper2” has 79718 characters with 88 unique characters, “paper3” has 44998 characters with 82 unique characters, “paper4” has 12870 characters with 77 unique characters, “paper5” has 11444 characters with 87 unique characters and “paper6” has 36230 characters with 91 unique characters after removal of escape sequences. The Huffman Coding provided a good compression ratio. The compression percentage for Huffman Coding reached about 33.42% for “paper2” and it was as low as 26.69% for “paper6”. Mutated Huffman Coding was run 30 times to nullify the effect randomness. As it includes mutation of the Huffman Tree the compression percentage is lesser than basic Huffman Coding. However, the value is not lower than 5.68% for “paper6” and for “paper 2” it was as high as 20.61%. Table 10.10 summarizes the results of the compression percentage.

The Compression Ratio (CR) is computed using the below formulation [149]:

$$CR = \frac{LengthofBits(Huffman_{Code(i)}) \times Frequency(i)}{LengthofBits(Decimal_{Code(i)}) \times Frequency(i)}$$

Where ‘i’ is the index of a particular unique character in the code table.

Now Compression Percentage (CP) is given below [149]: $CP = (1 - CR) \times 100\%$

Table 10.10: Results of Compression Percentages

Text File Name	Huffman Coding	Mutated Huffman Coding (Avg)
Compression Percentage (CP)		
paper1	27.38%	13.76%
paper2	33.42%	20.61%
paper3	32.65%	18.00%
paper4	31.75%	19.39%
paper5	27.90%	6.08%
paper6	26.69%	5.68%

10.10.3 Empirical Attainment Function of the Bi-Objective Optimization

It is difficult to identify a single Pareto optimal front from the 30 runs of a bi-objective evolutionary algorithm. Therefore, Fonseca et al. [98, 99] suggested plotting Empirical Attainment Function (EAF) for the solving the above problem. For the sake of illustration, we plotted the EAF plot for the case of the key length of 68, code table length of 22, plain text length of 7200 and processor configuration of 1.80GHz for the NSGA-II-KGA in Figure 10.13. It illustrates the best, the median and the worst attainment surfaces obtained by the proposed NSGA-II-KGA. The EAF plot [3] depicts k% empirical attainment surfaces out of many runs. We choose the median attainment surface, where the objective function F_1 has an optimal value of 1.0 and objective function F_2 has an optimal value of 0.69. This means that most of the keys will be of higher entropy value. Consequently, the attacker will find it difficult to crack the keys.

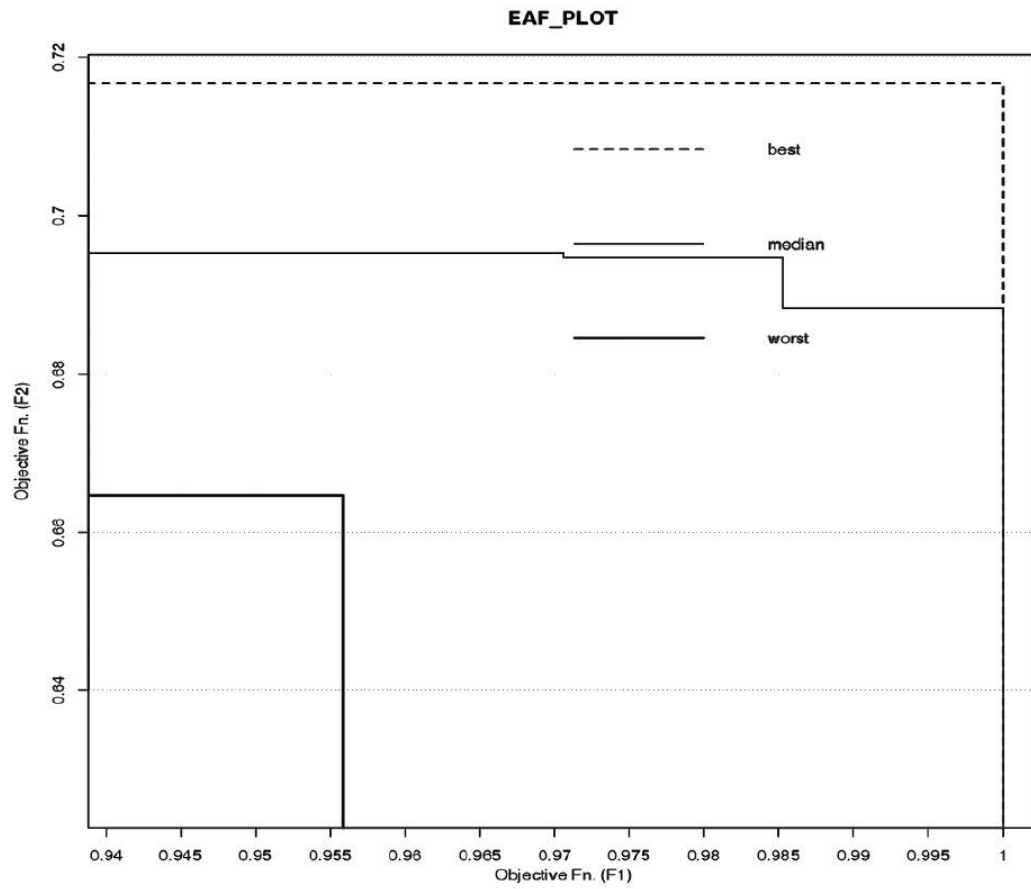


Figure 10.13: EAF of the Bi-objective Optimization using NSGA-II

10.10.4 Parameter Settings

The parameters settings for the evolutionary computing based key generation algorithms are summarized in Table 10.11.

Table 10.11: Parameter Setting

Parameter Setting of IMHS-KGA	
<i>Hmcr</i>	0.7 to 0.9 (Increased Linearly)
<i>Par</i>	0.4
Parameter Setting of DE-KGA	
<i>Crossover Rate</i>	0.55
<i>F</i>	1
Parameter Setting of IMHS+DE-KGA	
<i>Hmcr</i>	0.7 to 0.9 (Increased Linearly)
<i>Par</i>	0.4
<i>Crossover Rate</i>	0.55
<i>F</i>	1
Parameter Setting of NSGA-II-KGA	
<i>Cprob</i>	0.9
<i>Xoverdistance</i>	20
<i>Mprob</i>	0.1
<i>Mutationdistance</i>	3
<i>Tournament Size</i>	2
<i>Population Size</i>	30

10.10.5 Time Complexity of the Key Generation Algorithms, MHT coding, and Encryption

The time complexity of the key generation algorithms (KGAs) is listed below:

1. NSGA-II-KGA takes a time complexity of $O(NM^2)$. Where N (here it is 2) is the number of objectives and M is the population size [78].
2. DE-KGA takes a time complexity of $O(NP \cdot D \cdot OI_{max})$. Where NP is the population size, D is the dimensionality and OI_{max} is the maximum number of outer iterations [74].
3. IMHS-KGA takes a time complexity of $O(D \cdot G_{max})$. Where D is the dimensionality and G_{max} is the maximum number of generations [104].
4. IMHS+DE takes a time complexity of $O(NP \cdot D \cdot OI_{DE} + D \cdot G_{IMHS})$. Where OI_{DE} is the maximum number of outer iterations, DE algorithm and G_{IMHS} are the maximum number of generations of IMHS algorithm.

The time complexity of the Huffman tree coding depends on the number of leaves, i.e. number of characters in the code table. So for 'n' characters in the code table, the complexity is $O(n \log n)$. For mutating the Huffman tree, it takes constant time of $O(c)$, where c is constant. Therefore, the MHT coding takes $O(n \log n) + O(c)$ time complexity.

So, the total time complexity for the KGAs as well as the MHT coding is $O(KGA) + O(n \log n) + O(c)$, where $O(KGA)$ is described in the four cases above.

The time complexity of the encryption depends on the length of the key as well as the size of the plain text i.e.

$$O(\sum_i \text{Keystream Length}_i) * O(\text{PlaintextLength}).$$

10.11 Security Analysis and Comparison

We discuss three ways of security analysis. The first one is checking the key space generated. Second is to check for the classical attacks, and last is inspection of the randomness of the PRNG.

10.11.1 Key Space

Two types attacks on the key stream can be performed. One is by brute force attack, and the other is by using evolutionary computing itself. In the first type, there are two approaches for brute force attack. These are:

1. Using ASCII characters for the brute force attack on the keys
2. Bit-level brute force attack on the keys

Approach 1: Brute force attack on the keys using ASCII characters.

Under the brute force attack using ASCII characters, there are two possible cases that may occur (refer section 10.6.4):

Case 1: Unique characters in the plain text greater than length of the key stream. For case 1, to obtain a key stream of length 'r', 95 characters are used with a permutation of 95 characters taken 'r' at a time without repetition. The value of 'r' ranges from 1 to 95. The worst case would be to search $\sum_{r=1}^{95} 95P_r$ key stream space.

Case 2: Unique characters in the plain text less than the length of the key stream. To generate a key stream of length 'r', the same number of characters are used with

a permutation of 95 characters taken 'r' at a time. However, this time, repetition is allowed. The value of 'r' ranges from 1 to 95. In the worst case, we need to search $\sum_{r=1}^{keystream\ length} 95^r$ key stream space.

In the cases provided above, the key stream length generated is large and it is NP-hard to crack the keys using brute force attack. For example, case 1 has $\sum_{r=1}^{95} 95^{P_r}$ key space and for the maximum key length considered 'R' (i.e. 68, then case 2 has $\sum_{r=1}^{68} 95^r$) the key space is $\sum_{r=1}^{keystream\ length} 95^r$, which are hard to crack as in both cases the search space is large. The key space is very large, so it is impossible for brute force attack to happen.

Approach 2: Brute force attack on the keys at bit-level.

If we consider R-character key length, we have R*8 bits (i.e., for 68 characters key length we have 68*8 = 544 bits). The brute force attack on keys at bit-level is also NP-hard, as $2^{(R*8)}$ (2^{544}) possible combinations have to search for brute force attack. In this current work, we have considered only 68 length key as the maximum length, but even larger length keys are also possible using evolutionary computing.

Another type of attack on keys is using evolutionary algorithms itself [95, 243, 327]. Even this kind of attack is not going to be very easy, the problem is NP-hard and also it requires exploring all random seeds. Also, the speed up achieved for cracking key using evolutionary computing is $2^x/2^y$, where 2^x is the number of computations needed for cracking the key using evolutionary computing which is less than 2^y , where 2^y is the number of computations using brute force attack. If $x > 1024$ and $x < y$, even evolutionary algorithms will find it difficult to crack the key.

10.11.2 Classical Attacks

Jakimoski and Subbalakshmi in [137], performed the security analysis for possible encryption schemes. To avoid the known plain text attack, the following steps were suggested in [122, 137], when using mutated Huffman coding trees, which are also followed:

1. A different basic Huffman tree is generated for every new message.
2. Code word length remains same as that of basic Huffman tree in every mutation step.
3. Mutation is done dynamically.

Cipher text only, Chosen Plain text and Known Plain text attacks [286] With larger key space to a exhaustive search, mutated Huffman character code table and adding predetermined value it becomes hard to perform cipher text only, chosen plain text and known plain text attacks. (Please refer [174, 284, 285, 286])

10.11.3 Test for Cryptographically Secure PRNG

The mother PRNG used in generating ASCII characters for evolutionary computing is tested for pseudo-randomness. The mother PRNG is wrapped around the evolutionary computing based key generation algorithms which are providing a protection around mother PRNG as given in [328]. We are using ASCII characters codes between 33 and 127 generated from mother PRNG wrapped around by evolutionary KGAs. Therefore, if we check that our mother PRNG is pseudo-random it automatically implies the same to evolutionary computing based KGAs which are also PRNGs. There we performed the NIST 800-22rev1a statistical test suite [10] on mother PRNG utilized. We performed the test for 10,000,000 ASCII characters of ‘0’ and ‘1’ divided into 100 sequences. The mother PRNG was found to be pseudo random.

The testing of evolutionary computing based KGAs is not possible, as the NIST 800-22rev1a statistical test suite is utilized for testing ASCII characters of ‘0’ and ‘1’ or bits generated from PRNG. Proposed evolutionary computing based KGAs generate ASCII codes which are decimal values between 33 and 127. It is wrong to map to their corresponding binary values while checking for the pseudo-randomness of ASCII characters. But, if we prove that our mother PRNG is cryptographically secure PRNG and by wrapping mother PRNG around evolutionary computing based KGAs, we are providing security to mother PRNG. Also, the key generated from evolutionary computing based KGAs is also cryptographically secure as it uses the power of mother PRNG.

But, we tested our four key generation algorithms on NIST 800-22rev1a by generating bits (Please refer to [280] for the approaches of comparison). As we can see NSGA-II-KGA and DE-KGA_{WS} performed on par with standard HMAC-DRBG with the mean and standard deviation (Std. Dev.) close to HMAC-DRBG (please see Table 10.12). (The Hash-DRBG and HMAC-DRBG are defined in Perl’s Crypt-DRBG-0.000002 Module, which have been used for the generation of the respective DRBG data for the NIST CS-PRNG test.)

Table 10.12: Comparison of proposed KGAs with Standard DRBGs (HMAC, Hash)

DRBG and KGAs	Mean	Std. Dev.
Hash-DRBG	0.50	0.30
HMAC-DRBG	0.46	0.28
NSGA-II-KGA	0.42	0.29
DE-KGA _{WS}	0.44	0.31
IMHS+DE-KGA _{WS}	0.38	0.29
IMHS-KGA _{WS}	0.26	0.28

10.11.4 Analysis of the proposed KGAs with standard Stream Ciphers

a. RC4: Two main problems faced with RC4 are non-random keys and related keys. The first attack of non- random keys can be overcome if the keys are generated from PRNG. This is ensured by the basic principle of stochastic optimization algorithms that the seed for every run of key generation is set randomly and this principle makes the proposed KGAs generate random keys. Secondly, we have already proved that by utilizing the second objective function of entropy we ensured that the keys are uniformly distributed. Thereby, we ensured that even if the partial key is known and brute key space search is used, it is hard for the attacker to find the key. Also, we can generate large length keys that are uniformly distributed, which turns out to be the primary advantage of the proposed KGAs. By generating uniformly distributed large length keys, the attacker finds it hard to crack the key stream, even if the partial key stream is known (See section 10.11.1, for the key space analysis).

b. Salsa20: In 2012, the attack was improved by Shi et al. [270] against Salsa20/7 (128-bit key) to a time complexity of 2^{109} and Salsa20/8 (256-bit key) to 2^{250} . More random and larger the key, harder it is to crack the Salsa20. The advantage of two of the proposed KGAs namely NSGA-II-KGA and DE-KGA is that we can generate large length keys that are uniformly distributed. Even if the partial key stream is known as explained above for RC4, by generating uniformly distributed large length keys, the attacker finds it hard to crack the key stream (See section 10.11.1, for the key space analysis).

10.12 Conclusions and Future Directions

We conclude that the bi-objective evolutionary optimization algorithm viz., NSGA-II is employed for key generation for the first time and its advantage is the availability of a set of Pareto-optimal solutions to choose from when any of the keys are compromised. Moreover, the combination of this method with the mutated Huffman coding is novel. Further, we infer that the concept of entropy becomes helpful when the number of unique characters in the plain text is less than the length of the key stream and repetitions of the characters are allowed in the key stream.

The EAF plot yields further insights in that the median surface is closer to the best surface, which is an indication that bi-objective optimization framework did the trick. We choose the median empirical attainment surface, where the number of keys with high entropy is present, as illustrated in Figure 10.13. The population-based evolutionary computing based key generation has an advantage of picking alternate keys from the population of solutions.

This novel cryptographic scheme has wide application in various fields like finance, military, mobile, e-commerce, multimedia, etc., where data has to be sent over the medium securely. The evolutionary key generation along with mutated Huffman coding is a novel method. In future, the applications of the proposed method in performing secure financial transactions shall be explored, where encryption and compression play a critical role.

This work has possible applications in the payment systems as well as the wireless sensor networks where the computational power is limited. As we have demonstrated by the memory utilization as well as the encryption times that our model can be employed when the computational power is limited. The above metrics are comparatively on par with that of [218] (but cannot be compared as written in R, which is much slower than JAVA). In future, if a pseudo-randomness test suite is built for the ASCII characters (base > 2 and base < 95), apart from '0' and '1' (base 2), we will test our evolutionary key generation algorithms on this kind of pseudo-randomness tests. Testing randomness via words (ASCII characters) with a given frequency [329] is performed utilizing objective function two i.e. entropy. Maximum entropy ensures that the words (ASCII characters) are uniformly distributed across the key.

The present work is implemented in R-tool. Even though it is not implemented in hardware or low-level languages (for example VHDL or Verilog), it can be implemented in future. It is noteworthy that it has proved its efficacy in the high-level language. The present implementation in hardware or low-level language is constrained by the ability to implement in hardware which is out of the scope of the present research article.

The one primary reason for choosing ASCII over UNICODE is that ASCII takes less memory than UNICODE and UNICODE also support ASCII characters. This makes our implementation more useful for low memory devices. Also, most of the ASCII characters are supported by the physical keyboards whereas the UNICODE characters are not supported by the physical keyboards. This makes our implementation with ASCII characters general to all the devices.

Chapter 11

Conclusions and directions for future research

11.1 Summary and conclusions

- In this thesis, we reviewed relevant articles which are linked to the effective use of EC techniques to solve analytical problems of BFSI sector and addressed the analytical problems of credit scoring, financial statement fraud detection, MBA, customer segmentation, churn detection, insurance fraud detection and default detection. The thesis also details how EC techniques can help enhance the customer relationship, thereby enhancing the return on investment towards the company.
- We presented the applications of EC approaches for solving some of the operational problems within the banks such as bank efficiency prediction and security. Current work would be useful to the bankers, financial firms along with academicians and researchers in the domains of the BFSI sector.
- Three continuous RCS problems with four and five component reliabilities are considered as the test problems and single objective optimization problems are employed for solving them. Optimization with DE and IMHS+MDE, that has been tested on these RCS problems, achieved improved optima along with better function evaluations. Conclude that the DE and proposed hybrid meta-heuristic proved its effectiveness on RCS problems. We also tested the IMHS+MDE hybrid on CEC 2013 LSGO benchmark. By testing the EC techniques i.e. HS, IMHS,

DE, MDE and IMHS+MDE techniques on the standard optimization problems and on CEC 2013 LSGO test suite, we proved the effectiveness of our proposed EC techniques (i.e. IMHS, MDE, IMHS+MDE). But, considering the manual tuning of parameters for the DE, MDE, IMHS, HS and IMHS+MDE, we moved towards an adaptive version of DE where automated tuning of control parameters happens. We also investigated the effectiveness of multi-objective NSGA-II for one of the operational problem i.e. security. Therefore, we investigated the analytical and operational problems in single, multi-objective and hybrid optimization frameworks.

- We proposed ADE-based FSS using four classification algorithms and employed MCC as the fitness function. The ADE-FSS is employed on three datasets, which have significance to banking domain, i.e. credit scoring and fraud detection. This is the first study where MCC is employed as an evaluation measure for feature subset selection. In this study, we also proposed an adaptive version of the wrapper where there is no need to tune the parameters of the wrapper manually.
- Here in this study, we addressed the data mining task of classification. We developed rule-based classification algorithms using DE and ADE-based techniques on the features of credit scoring and financial statement fraud data as well as on the feature subsets from above wrapper-based feature subset selection. The benefits of rule-based models are that they are easy to interpret and can be used to classify the samples rapidly, etc. Rules are interpreted easily by top level managers of the banking firms, financial firms and insurance companies than the black box techniques like artificial neural networks, SVMs, etc.
- In this study, we addressed the data mining task of MBA. We developed high utility itemset and high utility association rule mining algorithms with BDE and ABDE, which are found to be efficient in terms of the amount of itemsets generated at various thresholds. This work turns out to show the power and versatility associated with the BDE and ABDE. One possible benefit of this work is in the retail sector, to allocate and manage the retail shelf-space efficiently, which yields high ROI when it comes to businesses. Another advantage of this study is that,

we analyzed the transactional databases associated with the retail businesses, to recognize which possible items offer high revenue if grouped together.

- In this study, we addressed the data mining tasks of clustering and anomaly detection. We proposed five EC driven subspace-based clustering models for anomaly detection (i.e. HS, IMHS, DE, MDE and IMHS+MDE -based anomaly detection techniques) in the context of keystroke detection, churn detection, insurance fraud detection and default detection. The reason for the focusing on clustering based anomaly detection is that not in all situations labels will be available to perform classification. Therefore, clustering-based anomaly detection techniques are mostly preferred over classification techniques. We compared our results with the clustering and anomaly detection techniques. It is observed that our proposed clustering based anomaly detection technique outperforms the compared techniques. Therefore, it is a viable alternative to one-class classification and also anomaly detection.
- In this work, we solved the problem of regression. We proposed DE-trained, and Hebbian-trained FCM algorithms separately for predicting bank efficiency. In both cases, FCM is trained in two modes, in other words, sequential and batch modes. Minimizing the Mean Absolute Percentage Error (MAPE) error measure (in other words, on the efficiency concept value) is chosen as the fitness function for DE. To automate and to help the decision-maker with multiple solutions from the population of EC-based techniques, we employed Differential Evolution (DE) for training the Fuzzy Cognitive Map (FCM). Therefore, multiple alternative optimal solutions and also automated decision making are the important outcomes out of this study.
- We developed a bi-objective evolutionary optimization algorithm viz., NSGA-II for key generation. It has an advantage of selecting keys from a couple of Pareto-optimal solutions when some of the keys are compromised. Moreover, the blend of the proposed method using the mutated Huffman coding for compression is novel. Further, it can be inferred that the use of entropy becomes helpful if the amount of unique characters in the plain text is less than the length of the keystream and repetitions of characters are allowed in the keystream. This novel

cryptographic scheme has broad application in various fields such as for instance finance, military, mobile, e-commerce, multimedia where data should be sent over the medium securely. Therefore, evolutionary computing-based key generation (i.e. IMHS, DE, IMHS+MDE and NSGA-II-based KGAs) along with mutated Huffman coding is a novel method.

11.2 Future Directions

Some future directions are suggested below:

- EC techniques take more computational time, as tuning the multiple non-adaptive control parameters of the EC techniques take more time. Therefore, in this light, we can move towards adaptive and self-adaptive versions of EC techniques, by which time needed for control parameter tuning of EC techniques will be reduced drastically.
- Further, a number of these problems could be formulated as multi-objective optimization problems, that may pave way for the immediate adoption of evolutionary multi-objective (EMO) algorithms.
- Fuzzy optimization finds a direct application, as several of those problems may be reformulated to model the inherent fuzziness of the objectives and constraints.
- In the age of automation, EC-based techniques provide a great way of automating the conventional methods. EC techniques are an integral part of soft computing and AI. Although some genuinely believe that EC-based automation for solving the analytical and operational problems will undoubtedly be detrimental when it comes to employability of humans. But, it provides a larger depth to the profitability of the firms as well as in making positive decisions for the banks/financial firms.
- In the future, when it comes to FSS problem, we can fix on using one of the classification algorithms based on the results obtained thereby applying various Evolutionary Computing techniques. There is a scope of using EMO algorithms as well by modeling the issue in a multi-objective setting.

-
- In future, better EC techniques both in single and multi-objective frameworks may also be employed to solve the problems of rule based classification, high utility itemset mining and high utility association rule mining.
 - One limitation associated with DE trained FCM study from the domain perspective, is that a limited number of financial ratios are thought to model efficiency. In future, we want to extend this study by including more financial ratios that directly or indirectly influence efficiency. It's going to hopefully, present a significantly better image of the ground realities.
 - In the future, better convergence criteria may be developed and studied for the DE trained FCM approach. Better training algorithms could be proposed by hybridizing both working out algorithms implemented in the present work. Better modes of training can be suggested in future to handle large datasets. Better data-driven approaches for FCMs can be developed by formulating the problem in a multi-objective optimization framework.
 - Even though key generation using EC techniques isn't implemented in hardware or low-level languages, and it can be implemented in the future in a low-level language. It is noteworthy that the implementation has proved its efficacy in the high-level language. Also, the current implementation in hardware or low-level language is also constrained by the capability to implement in hardware.

References

- [1] Accounting scandals. https://en.wikipedia.org/wiki/Accounting_scandals.
- [2] America's Most Admired Companies 2007 - Lehman Brothers Holdings snapshot - FORTUNE. <http://archive.fortune.com/magazines/fortune/mostadmired/2007/snapshots/10312.html>.
- [3] eafplot function. <https://www.rdocumentation.org/packages/eaf/versions/1.07/topics/eafplot>.
- [4] Satyam Scandal. https://en.wikipedia.org/wiki/Satyam_scandal.
- [5] The 10 Worst Corporate Accounting Scandals of All Time - The Accounting Degree Review. <https://www.accounting-degree.org/scandals/>.
- [6] UCI Machine Learning Repository: Statlog (Australian Credit Approval) Data Set. [http://archive.ics.uci.edu/ml/datasets/statlog+\(australian+credit+approval\)](http://archive.ics.uci.edu/ml/datasets/statlog+(australian+credit+approval)).
- [7] UCI Machine Learning Repository: Statlog (German Credit Data Numeric) Data Set. [https://archive.ics.uci.edu/ml/datasets/statlog+\(german+credit+data\)](https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data)).
- [8] UCI Machine Learning Repository: Twenty Newsgroups Data Set. <https://archive.ics.uci.edu/ml/datasets/Twenty+Newsgroups/>, 1999.
- [9] Calgary corpus. <http://www.data-compression.info/Corpora/CalgaryCorpus/>, 2002.
- [10] A Statistical Test Suite for the Validation of Random Number Generators and Pseudo Random Number Generators for Cryptographic Applications. http://csrc.nist.gov/groups/ST/toolkit/rng/documentation_software.html, 2014.

-
- [11] H. A. Abdou. Genetic programming for credit scoring: The case of Egyptian public sector banks. *Expert Systems with Applications*, 36(9):11402–11417, 2009.
- [12] C. C. Aggarwal, A. Hinneburg, and D. A. Keim. On the surprising behavior of distance metrics in high dimensional space. In J. Van den Bussche and V. Vianu, editors, *Database Theory — ICDT 2001*, pages 420–434, London, United Kingdom, 2001. Springer Berlin Heidelberg.
- [13] C. C. Aggarwal and P. S. Yu. Outlier detection for high dimensional data. In T. Sellis and S. Mehrotra, editors, *Proceedings of the 2001 ACM SIGMOD international conference on Management of data - SIGMOD '01*, volume 30, pages 37–46, Santa Barbara, California, USA, 2001. ACM Press.
- [14] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference on Very Large Data Bases*, VLDB '94, page 487–499, Santiago, Chile, 1994. Morgan Kaufmann Publishers Inc.
- [15] M. M. Al-Jarrah. An Anomaly Detector for Keystroke Dynamics Based on Medians Vector Proximity. *Journal of Emerging Trends in Computing and Information Sciences*, 3(6):988–993, 2010.
- [16] H. Al-Sahaf, Y. Bi, Q. Chen, A. Lensen, Y. Mei, Y. Sun, B. Tran, B. Xue, and M. Zhang. A survey on evolutionary machine learning. *Journal of the Royal Society of New Zealand*, 49(2):205–228, apr 2019.
- [17] E. Alba, J. Garcia-Nieto, L. Jourdan, and E.-G. Talbi. Gene selection in cancer classification using PSO/SVM and GA/SVM hybrid algorithms. In *2007 IEEE Congress on Evolutionary Computation*, pages 284–290, Singapore, sep 2007. IEEE.
- [18] R. Aliehyaei and S. Khan. Ant Colony Optimization, Genetic Programming and a hybrid approach for credit scoring: A comparative study. In *The 8th International Conference on Software, Knowledge, Information Management and Applications (SKIMA 2014)*, pages 1–5, Dhaka, Bangladesh, dec 2014. IEEE.
- [19] M. S. Arunkumar, P. Suresh, and C. Gunavathi. High Utility Infrequent Itemset Mining Using a Customized Ant Colony Algorithm. *International Journal of Parallel Programming*, pages 1–17, dec 2018.
- [20] M. Aslan, M. Gunduz, and M. S. Kiran. JayaX: Jaya algorithm with xor operator for binary optimization. *Applied Soft Computing Journal*, 82:1–17, sep 2019.

-
- [21] C. A. S. Assis, A. C. M. Pereira, M. A. Pereira, and E. G. Carrano. A genetic programming approach for fraud detection in electronic transactions. In *2014 IEEE Symposium on Computational Intelligence in Cyber Security (CICS)*, pages 1–8, Orlando, FL, USA, dec 2014. IEEE.
 - [22] W.-H. Au, K. C. C. Chan, and X. Yao. A novel evolutionary data mining algorithm with applications to churn prediction. *IEEE Transactions on Evolutionary Computation*, 7(6):532–545, 2003.
 - [23] W. S. Awad. On the Application of Evolutionary Computation Techniques in Designing Stream Cipher Systems. *International Journal of Computational Intelligence Systems*, 4(5):921–928, 2011.
 - [24] T. Back, D. B. Fogel, and Z. Michalewicz. *Handbook of Evolutionary Computation*. IOP Publishing Ltd., GBR, 1st edition, 1997.
 - [25] T. Back and Thomas. *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford University Press, NewYork, NY, USA, 1996.
 - [26] K. S. Balagani, V. V. Phoha, A. Ray, and S. Phoha. On the discriminability of keystroke feature vectors used in fixed text keystroke authentication. *Pattern Recognition Letters*, 32(7):1070–1080, may 2011.
 - [27] A. Banitalebi, M. I. A. Aziz, and Z. A. Aziz. A self-adaptive binary differential evolution algorithm for large scale binary optimization problems. *Information Sciences*, 367-368:487–511, nov 2016.
 - [28] E. Barker, W. Barker, W. Burr, W. Polk, and M. Smid. NIST special publication 800-57. *NIST Special Publication*, 800(57):1–142, 2007.
 - [29] T. Bellotti and J. Crook. Support vector machines for credit scoring and discovery of significant features. *Expert Systems with Applications*, 36(2):3302–3308, mar 2009.
 - [30] S. M. Bellovin. Frank Miller: Inventor of the One-Time pad. *Cryptologia*, 35(3):203–222, jul 2011.
 - [31] T. Bhat, D. Venkataramani, V. Ravi, and C. Murty. An improved differential evolution method for efficient parameter estimation in biofilter modeling. *Biochemical Engineering Journal*, 28(2):167–176, 2006.
 - [32] S. Boughorbel, F. Jarray, and M. El-Anbari. Optimal classifier for imbalanced

-
- data using Matthews Correlation Coefficient metric. *PLOS ONE*, 12(6):e0177678, jun 2017.
- [33] Y. Boutalis, T. Kottas, and M. Christodoulou. On the existence and uniqueness of solutions for the concept values in Fuzzy Cognitive Maps. In *47th IEEE Conference on Decision and Control*, pages 98–104, Cancun, Mexico, 2008. IEEE.
- [34] Y. Boutalis, T. Kottas, and M. Christodoulou. Adaptive Estimation of Fuzzy Cognitive Maps With Proven Stability and Parameter Convergence. *IEEE Transactions on Fuzzy Systems*, 17(4):874–889, aug 2009.
- [35] J. Bozsik. Genetic algorithm in default forecast. In *Proceedings of the 8th international conference on on applied informatics*, pages 41–51, Eger, Hungary, 2010. IEEE.
- [36] A. Brabazon, J. Cahill, P. Keenan, and D. Walsh. Identifying online credit card fraud using Artificial Immune Systems. In *IEEE Congress on Evolutionary Computation*, pages 1–7, Barcelona, Spain, jul 2010. IEEE.
- [37] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and regression trees*. Wadsworth International Group, Monterey, CA, 1984.
- [38] J. Brest, B. Boskovic, A. Zamuda, I. Fister, and M. S. Maucec. Self-adaptive differential evolution algorithm with a small and varying population size. In *2012 IEEE Congress on Evolutionary Computation*, pages 1–8, Brisbane, QLD, Australia, jun 2012. IEEE.
- [39] J. Brest and M. S. Maucec. Self-adaptive differential evolution algorithm using population size reduction and three strategies. *Soft Computing*, 15(11):2157–2174, nov 2011.
- [40] J. Brest, A. Zamuda, I. Fister, and M. S. Maucec. Large scale global optimization using self-adaptive differential evolution algorithm. In *IEEE Congress on Evolutionary Computation*, pages 1–8, Barcelona, Spain, jul 2010. IEEE.
- [41] M. M. Breunig, H.-P. Kriegel, R. T. Ng, J. Sander, M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. LOF: identifying density-based local outliers. In *ACM SIGMOD international conference on Management of data*, volume 29, pages 93–104, Dallas, Texas, USA, 2000. ACM Press.
- [42] S. Bueno and J. L. Salmeron. Benchmarking main activation functions in fuzzy cognitive maps. *Expert Systems with Applications*, 36(3):5221–5229, apr 2009.

-
- [43] B. Busta and R. Weinberg. Using Benford’s law and neural networks as a review procedure. *Managerial Auditing Journal*, 13(6):356–366, aug 1998.
- [44] Y. Caballero, D. Alvarez, R. Bello, and M. M. Garcia. Feature Selection Algorithms Using Rough Set Theory. In L. D. Mourelle, N. Nedjah, J. Kacprzyk, and A. Abraham, editors, *Seventh International Conference on Intelligent Systems Design and Applications (ISDA 2007)*, pages 407–411, Rio de Janeiro, Brazil, oct 2007. IEEE.
- [45] W. Cai, X. Lin, X. Huang, and H. Zhong. In *2009 International Conference on Computational Intelligence and Software Engineering*, pages 1–4, Wuhan, China. IEEE.
- [46] T. G. Calderon and J. J. Cheh. A roadmap for future neural networks research in auditing and risk assessment. *International Journal of Accounting Information Systems*, 3(4):203–236, dec 2002.
- [47] J. C. H. Castro and P. I. Vinuela. Evolutionary Computation in computer security and cryptography. *New Generation Computing*, 23(3):193–199, 2005.
- [48] L. D. Castro and J. Timmis. *Artificial immune systems: a new computational intelligence approach*. Springer Science & Business Media, London, UK, 2002.
- [49] M. Cecchini, H. Aytug, G. J. Koehler, and P. Pathak. Detecting Management Fraud in Public Companies. *Management Science*, 56(7):1146–1160, jul 2010.
- [50] M. J. Cerullo and V. Cerullo. Using neural networks to predict financial reporting fraud: Part 1. *Computer Fraud & Security*, 1999(5):14–17, may 1999.
- [51] M. J. Cerullo and V. Cerullo. Using neural networks to predict financial reporting fraud: Part 2. *Computer Fraud & Security*, 1999(6):14–17, jun 1999.
- [52] R. Chan, Q. Yang, and Y.-D. Shen. Mining high utility itemsets. In *Third IEEE International Conference on Data Mining*, pages 19–26, Melbourne, FL, USA, 2003. IEEE Comput. Soc.
- [53] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Computing Surveys*, 41(3):1–58, jul 2009.
- [54] P. Chaudhary, A. Mondal, and P. K. Reddy. A Flexible and Efficient Indexing Scheme for Placement of Top-Utility Itemsets for Different Slot Sizes. In P. Reddy, S. Sureka, A Chakravarthy, and S. Bhalla, editors, *International Conference on Big Data Analytics*, pages 257–277, Hyderabad, India, dec 2017. Springer, Cham.

-
- [55] N. Chauhan and V. Ravi. Differential evolution and threshold accepting hybrid algorithm for unconstrained optimisation. *International Journal of Bio-Inspired Computation*, 2(3/4):169, 2010.
- [56] D. Chen, S. L. Sain, and K. Guo. Data mining for the online retail industry: A case study of RFM model-based customer segmentation using data mining. *Journal of Database Marketing & Customer Strategy Management*, 19(3):197–208, sep 2012.
- [57] F.-L. Chen and F.-C. Li. Combination of feature selection approaches with SVM in credit scoring. *Expert Systems with Applications*, 37(7):4902–4909, jul 2010.
- [58] G. Chen, M. Firth, D. N. Gao, and O. M. Rui. Ownership structure, corporate governance, and fraud: Evidence from China. *Journal of Corporate Finance*, 12(3):424–448, jun 2006.
- [59] Q. Chen, D. Zhang, L. Wei, and H. Chen. A modified genetic programming for behavior scoring problem. In *IEEE Symposium on Computational Intelligence and Data Mining, CIDM 2007*, pages 535–539, Honolulu, HI, USA, 2007. IEEE.
- [60] Y. Chen, S. Mabu, and K. Hirasawa. Genetic relation algorithm with guided mutation for the large-scale portfolio optimization. *Expert Systems with Applications*, 38(4):3353–3363, 2011.
- [61] Y.-W. Chen and C.-J. Lin. Combining SVMs with various feature selection strategies. In I. Guyon, M. Nikravesh, S. Gunn, and L. A. Zadeh, editors, *Feature Extraction: Foundations and Applications*, pages 315–324. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.
- [62] Y. W. C. Chien and Y. L. Chen. Mining associative classification rules with stock trading data-A GA-based method. *Knowledge-Based Systems*, 23(6):605–614, 2010.
- [63] C.-H. Chou, C.-C. Lin, Y.-H. Liu, and F. Chang. A prototype classification method and its use in a hybrid solution for multiclass pattern recognition. *Pattern Recognition*, 39(4):624–634, apr 2006.
- [64] R. Choudhuri and V. Ravi. A Hybrid Harmony Search and Modified Great Deluge Algorithm for Unconstrained Optimisation. Technical report, IDRBT, Hyderabad, Telangana, 2010.
- [65] C.-T. Chu, S. K. Kim, Y.-A. Lin, Y. Yu, G. Bradski, A. Y. Ng, and K. Olukotun. Map-reduce for machine learning on multicore. In B. Schölkopf, J. C. Platt, and

-
- T. Hoffman, editors, *Proceedings of the 19th International Conference on Neural Information Processing Systems*, pages 281–288, Canada, 2006. MIT Press.
- [66] L.-Y. Chuang, C.-H. Yang, and C.-H. Yang. Tabu Search and Binary Particle Swarm Optimization for Feature Selection Using Microarray Data. *Journal of Computational Biology*, 16(12):1689–1703, dec 2009.
- [67] A. Coloni, M. Dorigo, and V. Maniezzo. Distributed Optimization by Ant Colonies. In *Proceedings of the European Conference on Artificial Life*, pages 134–142, Paris, France, 1991.
- [68] M. Corazza, S. Funari, and R. Gusso. Particle swarm optimization for preference disaggregation in multicriteria credit scoring problems. In M. Corazza and C. Pizzi, editors, *Mathematical and Statistical Methods for Actuarial Sciences and Finance*, pages 119–130. Springer International Publishing, Cham, 2014.
- [69] B. Crawford, R. Soto, G. Astorga, J. García, C. Castro, and F. Paredes. Putting continuous metaheuristics to work in binary search spaces. *Complexity*, 2017, 2017.
- [70] K. D. Crawford and R. L. Wainwright. Applying Genetic Algorithms to Outlier Detection. In L. J. Eshelman, editor, *Proceedings of the 6th International Conference on Genetic Algorithms*, pages 546–550, San Francisco, 1995. M. Kaufman.
- [71] F. Dachsel and W. Schwarz. Chaos and cryptography. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 48(12):1498–1509, 2001.
- [72] S. Das, A. Mukhopadhyay, A. Roy, A. Abraham, and B. K. Panigrahi. Exploratory Power of the Harmony Search Algorithm: Analysis and Improvements for Global Numerical Optimization. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 41(1):89–106, feb 2011.
- [73] S. Das, S. S. Mullick, and P. N. Suganthan. Recent advances in differential evolution-An updated survey. *Swarm and Evolutionary Computation*, 27:1–30, apr 2016.
- [74] S. Das and P. N. Suganthan. Differential Evolution: A Survey of the State-of-the-Art. *IEEE Transactions on Evolutionary Computation*, 15(1):4–31, feb 2011.
- [75] D. Dasgupta. *Artificial immune systems and their applications*. Springer Berlin Heidelberg, Berlin, Heidelberg, 1999.

-
- [76] D. Dasgupta. Advances in artificial immune systems. *Comp. Intell. Mag.*, 1(4):40–49, Nov. 2006.
- [77] M. Dash and H. Liu. Feature selection for classification. *Intelligent Data Analysis*, 1(1-4):131–156, jan 1997.
- [78] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.
- [79] B. Delman. *Genetic algorithms in cryptography*. PhD thesis, Department of Computer Engineering, Kate Gleason College of Engineering, Rochester Institute of Technology, 2004.
- [80] Y. Deng and Y. Zhong. Keystroke Dynamics User Authentication Based on Gaussian Mixture Model and Deep Belief Nets. *ISRN Signal Processing*, 2013:1–7, 2013.
- [81] A. Deshmukh and L. Talluru. A rule-based fuzzy reasoning system for assessing the risk of management fraud. *International Journal of Intelligent Systems in Accounting, Finance & Management*, 7(4):223–241, dec 1998.
- [82] J. A. Dickerson and B. Kosko. Virtual Worlds as Fuzzy Cognitive Maps. *Presence: Teleoperators and Virtual Environments*, 3(2):173–189, jan 1994.
- [83] G. Dong, K. K. Lai, and L. Zhou. Simulated Annealing Based Rule Extraction Algorithm for Credit Scoring Problem. In *2009 International Joint Conference on Computational Sciences and Optimization*, pages 22–26, Sanya, Hainan, China, 2009. IEEE.
- [84] W. Duch, R. Setiono, and J. Zurada. Computational intelligence methods for rule-based data understanding. *Proceedings of the IEEE*, 92(5):771–805, may 2004.
- [85] G. Dueck and T. Scheurer. Threshold Accepting: A General Purpose Optimization Algorithm. *Journal of Computational Physics*, 90:161–175, 1990.
- [86] E. Duman and I. Elikucuk. Applying migrating birds optimization to credit card fraud detection. In J. Li, L. Cao, C. Wang, K. C. Tan, B. Liu, J. Pei, and V. S. Tseng, editors, *Trends and Applications in Knowledge Discovery and Data Mining*, pages 416–427, Gold Coast, QLD, Australia, 2013. Springer Berlin Heidelberg.

-
- [87] E. Duman, M. Uysal, and A. F. Alkaya. Migrating Birds Optimization: A new metaheuristic approach and its performance on quadratic assignment problem. *Information Sciences*, 217:65–77, 2012.
- [88] A. E. Eiben, A. E. Koudijs, and F. Slisser. Genetic modelling of customer retention. In W. Banzhaf, R. Poli, M. Schoenauer, and T. C. Fogarty, editors, *Genetic Programming*, pages 178–186. Springer Berlin Heidelberg, Paris, France, 1998.
- [89] A. E. Eiben and S. K. Smit. Evolutionary algorithm parameters and methods to tune them. In Y. Hamadi, E. Monfroy, and F. Saubion, editors, *Autonomous Search*, pages 15–36. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.
- [90] T. Eltaieb and A. Mahmood. Differential Evolution: A Survey and Analysis. *Applied Sciences*, 8(10):1945, oct 2018.
- [91] A. Engelbrecht and G. Pampara. Binary differential evolution strategies. In *2007 IEEE Congress on Evolutionary Computation*, pages 1942–1947, Singapore, Singapore, sep 2007. IEEE.
- [92] K. M. Fanning and K. O. Cogger. Neural network detection of management fraud using published financial data. *International Journal of Intelligent Systems in Accounting, Finance & Management*, 7(1):21–41, mar 1998.
- [93] K. M. Fanning and K. O. Cogger. Neural network detection of management fraud using published financial data. *International Journal of Intelligent Systems in Accounting, Finance & Management*, 7(1):21–41, mar 1998.
- [94] E. H. Feroz, T. M. Kwon, V. S. Pastena, and K. Park. The efficacy of red flags in predicting the SEC’s targets: an artificial neural networks approach. *International Journal of Intelligent Systems in Accounting, Finance & Management*, 9(3):145–157, sep 2000.
- [95] B. N. Ferriman. *Cryptanalysis of the RC4 Stream Cipher using Evolutionary Computation Methods*. PhD thesis, University of Guelph, 2013.
- [96] I. Fister, I. J. Fister, and J. B. Zumer. Memetic artificial bee colony algorithm for large-scale global optimization. In *2012 IEEE Congress on Evolutionary Computation*, pages 1–8, Brisbane, QLD, Australia, jun 2012. IEEE.
- [97] D. J. Fogarty. Using genetic algorithms for credit scoring system maintenance functions. *International Journal of Artificial Intelligence & Applications (IJAIA)*, 3(6):1–8, 2012.

-
- [98] C. M. Fonseca, V. G. da Fonseca, and L. Paquete. Exploring the Performance of Stochastic Multiobjective Optimisers with the Second-Order Attainment Function. In E. Coello Coello, C.A. Hernández Aguirre, A. Zitzler, editor, *Evolutionary Multi-Criterion Optimization (EMO-2005)*, pages 250–264, Guanajuato, Mexico, 2005. Springer, Berlin, Heidelberg.
- [99] C. M. Fonseca, A. P. Guerreiro, M. López-Ibáñez, and L. Paquete. On the computation of the empirical attainment function. In R. H. C. Takahashi, K. Deb, E. F. Wanner, and S. Greco, editors, *Proceedings of the 6th international conference on Evolutionary multi-criterion optimization*, pages 106–120, Ouro Preto, Brazil, 2011. Springer.
- [100] P. Fournier-Viger, C.-W. Wu, S. Zida, and V. S. Tseng. FHM: Faster High-Utility Itemset Mining Using Estimated Utility Co-occurrence Pruning. In J. Andreasen, T. Christiansen, H. Cubero and Z. Raś, editors, *21st International Symposium on Methodologies for Intelligent Systems*, pages 83–92, Roskilde, Denmark, 2014. Springer, Cham.
- [101] A. A. Freitas. A Review of evolutionary Algorithms for Data Mining. In O. Maimon and L. Rokach, editors, *Soft Computing for Knowledge Discovery and Data Mining*, chapter 4, pages 79–111. Springer US, Boston, MA, 2008.
- [102] M. F. A. Gadi, X. Wang, and A. P. do Lago. Credit card fraud detection with artificial immune system. In P. J. Bentley, D. Lee, and S. Jung, editors, *Artificial Immune Systems*, pages 119–131. Springer Berlin Heidelberg, Phuket, Thailand, 2008.
- [103] P. Ganghishetti and R. Vadlamani. Association rule mining via evolutionary multi-objective optimization. In M. N. Murty, X. He, R. R. Chillarige, and P. Weng, editors, *Multi-disciplinary Trends in Artificial Intelligence*, pages 35–46. Springer International Publishing, Bangalore, India, 2014.
- [104] Z. W. Geem, J. H. Kim, and G. Loganathan. A New Heuristic Optimization Algorithm: Harmony Search. *SIMULATION*, 76(2):60–68, feb 2001.
- [105] M. Gendreau and J.-Y. Potvin. *Handbook of metaheuristics*. Springer US, Boston, MA, 2010.
- [106] R. Giot, B. Dorizzi, and C. Rosenberger. A review on the public benchmark databases for static keystroke dynamics. *Computers & Security*, 55:46–61, nov 2015.

-
- [107] F. Glover. Tabu Search - Part II. *ORSA journal on Computing*, 2(1):4–32, 1989.
- [108] F. Glover. Tabu search Part I. *ORSA Journal on Computing*, 1(3):190–206, 1989.
- [109] R. Y. Goh and L. S. Lee. Credit Scoring: A Review on Support Vector Machines and Metaheuristic Approaches. *Advances in Operations Research*, 2019, 2019.
- [110] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., USA, 1st edition, 1989.
- [111] N. Gordini. Genetic Algorithms for Small Enterprises Default Prediction: Empirical Evidence from Italy. In M. K. Pour, editor, *Nature-Inspired Computing: Concepts, Methodologies, Tools, and Applications*, chapter 23, pages 571–607. IGI Global, Hershey, PA, USA, 2017.
- [112] P. Greenberg. *CRM at the Speed of Light, Third Edition: Essential Customer Strategies for the 21st Century*. McGraw-Hill, Inc., USA, 3 edition, 2004.
- [113] S.-u. Guan and S. Zhang. An evolutionary approach to the design of controllable cellular automata structure for random number generation. *IEEE Transactions on Evolutionary Computation*, 7(1):23–36, 2003.
- [114] Y. Guo and C. Dong. A novel intelligent credit scoring method using MOPSO. In *29th Chinese Control and Decision Conference, CCDC 2017*, pages 6584–6588, Chongqing, China, jul 2017. IEEE.
- [115] Y. Guo, J. He, L. Xu, and W. Liu. A novel multi-objective particle swarm optimization for comprehensible credit scoring. *Soft Computing*, 23(18):9009–9023, sep 2018.
- [116] I. Guyon and A. Elisseeff. An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research*, 3(Mar):1157–1182, 2003.
- [117] N. S. Halvaie and M. K. Akbari. A novel model for credit card fraud detection using Artificial Immune Systems. *Applied Soft Computing*, 24:40–49, 2014.
- [118] J. Han, M. Kamber, and J. Pei. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edition, 2011.
- [119] J. V. Hansen, J. B. McDonald, and J. D. Stice. Artificial Intelligence and Generalized Qualitative-Response Models: An Empirical Test on Two Audit Decision-Making Domains. *Decision Sciences*, 23(3):708–723, may 1992.
- [120] J. A. Hartigan and M. A. Wong. A K-Means Clustering Algorithm. *Applied Statistics*, 28(1):100–108, 1979.

-
- [121] Z. He, S. Deng, and X. Xu. An Optimization Model for Outlier Detection in Categorical Data. In D. Huang, X. Zhang, and G. Huang, editors, *International Conference on Intelligent Computing*, pages 400–409, Hefei, China, 2005. Springer, Berlin, Heidelberg.
- [122] H. Hermassi, R. Rhouma, and S. Belghith. Joint compression and encryption using chaotically mutated Huffman trees. *Communications in Nonlinear Science and Numerical Simulation*, 15(10):2987–2999, 2016.
- [123] J. Hernandez, A. Sez nec, and P. Isasi. On the design of state-of-the-art pseudo-random number generators by means of genetic programming. In *Congress on Evolutionary Computation*, pages 1510–1516, Portland, OR, USA, 2004. IEEE.
- [124] R. Hochreiter. Computing trading strategies based on financial sentiment data using evolutionary optimization. In R. Matoušek, editor, *Mendel 2015*, pages 181–191. Springer International Publishing, Brno, Czech Republic, 2015.
- [125] V. Hodge and J. Austin. A Survey of Outlier Detection Methodologies. *Artificial Intelligence Review*, 22(2):85–126, oct 2004.
- [126] N. P. Holden and A. A. Freitas. A hybrid PSO/ACO algorithm for classification. In *Proceedings of the 2007 GECCO conference companion on Genetic and evolutionary computation - GECCO '07*, page 2745, London, United Kingdom, 2007. ACM Press.
- [127] B. Hoogs, T. Kiehl, C. Lacombe, and D. Senturk. A genetic algorithm approach to detecting temporal patterns indicative of financial statement fraud. *Intelligent Systems in Accounting, Finance and Management*, 15(1-2):41–56, jan 2007.
- [128] R. Horst and P. M. Pardalos. *Handbook of global optimization*. Springer US, Boston, MA, 1995.
- [129] C.-L. Huang, M.-C. Chen, and C.-J. Wang. Credit scoring with a data mining approach based on support vector machines. *Expert Systems with Applications*, 33(4):847–856, nov 2007.
- [130] C.-L. Huang, M.-C. Chen, and C.-J. Wang. Credit scoring with a data mining approach based on support vector machines. *Expert Systems with Applications*, 33(4):847–856, nov 2007.
- [131] J.-J. Huang, G.-H. Tzeng, and C.-S. Ong. Two-stage genetic programming (2SGP) for the credit scoring model. *Applied Mathematics and Computation*, 174(2):1039–1053, 2006.

-
- [132] S.-M. Huang, D. C. Yen, L.-W. Yang, and J.-S. Hua. An investigation of Zipf's Law for fraud detection. *Decision Support Systems*, 46(1):70–83, dec 2008.
- [133] D. A. Huffman. A method for the construction of minimum-redundancy codes. *Resonance*, 11(2):91–99, feb 2006.
- [134] S. Ibrahim and M. A. Maarof. A review on biological inspired computation in cryptology. *Jurnal Teknologi Maklumat*, 17(1):90–98, 2005.
- [135] J. Isaacs, R. Watkins, and S. Foo. Evolving ant colony systems in hardware for random number generation. In *Congress on Evolutionary Computation*, pages 1450–1455, Honolulu, U. S. A., 2002. IEEE.
- [136] P. Isasi and J. C. Hernandez. Introduction to the Applications of Evolutionary Computation in Computer Security and Cryptography. *Computational Intelligence*, 20(3):445–449, 2004.
- [137] G. Jakimoski and K. P. Subbalakshmi. Cryptanalysis of some multimedia encryption schemes. *IEEE Transactions on Multimedia*, 10(3):330–338, apr 2008.
- [138] L. Jayashree, R. Lakshmi Devi, N. Papandrianos, and E. I. Papageorgiou. Application of Fuzzy Cognitive Map for geospatial dengue outbreak risk prediction of tropical regions of Southern India. *Intelligent Decision Technologies*, 12(2):231–250, mar 2018.
- [139] L. S. Jayashree, N. Palakkal, E. I. Papageorgiou, and K. Papageorgiou. Application of fuzzy cognitive maps in precision agriculture: a case study on coconut yield management of southern India's Malabar region. *Neural Computing and Applications*, 26(8):1963–1978, nov 2015.
- [140] S. Jhajharia, S. Mishra, and S. Bali. Public key cryptography using neural networks and genetic algorithms. In *Sixth International Conference on Contemporary Computing (IC3)*, Noida, India, aug 2013. IEEE.
- [141] J. Ji, N. Zhang, C. Liu, and N. Zhong. An Ant Colony Optimization Algorithm for Learning Classification Rules. In *2006 IEEE/WIC/ACM International Conference on Web Intelligence (WI 2006 Main Conference Proceedings)(WI'06)*, pages 1034–1037, Hong Kong, China, dec 2006. IEEE.
- [142] T. Jirapech-Umpai and S. Aitken. Feature selection and classification for microarray data analysis: evolutionary methods for identifying predictive genes. *BMC Bioinformatics*, 6(1):148, jun 2005.

-
- [143] R. Joyce and G. Gupta. Identity authentication based on keystroke latencies. *Communications of the ACM*, 33(2):168–176, feb 1990.
- [144] P. Kaelo and M. Ali. A numerical study of some modified differential evolution algorithms. *European Journal of Operational Research*, 169(3):1176–1184, mar 2006.
- [145] S. Kannimuthu and K. Premalatha. Discovery of High Utility Itemsets Using Genetic Algorithm with Ranked Mutation. *Applied Artificial Intelligence*, 28(4):337–359, apr 2014.
- [146] D. Karaboga and B. Akay. A survey: Algorithms simulating bee swarm intelligence. *Artificial Intelligence Review*, 31(1-4):61–85, 2009.
- [147] G. V. Kass. An Exploratory Technique for Investigating Large Quantities of Categorical Data. *Applied Statistics*, 29(2):119, 1980.
- [148] A. Kaveh and S. Talatahari. Particle swarm optimizer, ant colony strategy and harmony search scheme hybridized for optimization of truss structures. *Computers & Structures*, 87(5-6):267–283, mar 2009.
- [149] X. Kavousianos, E. Kalligeros, and D. Nikolos. Optimal selective huffman coding for test-data compression. *IEEE transactions on computers*, 56(8):1146–1152, 2007.
- [150] J. Kennedy and R. Eberhart. Particle swarm optimization. In *International Conference on Neural Networks (ICNN’95)*, pages 1942–1948, Perth, WA, Australia, 1995. IEEE.
- [151] J. Kennedy and R. Eberhart. A discrete binary version of the particle swarm algorithm. In *IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation*, volume 5, pages 4104–4108. IEEE, Orlando, FL, USA, 1997.
- [152] K. S. Killourhy and R. A. Maxion. Comparing anomaly-detection algorithms for keystroke dynamics. In *2009 IEEE/IFIP International Conference on Dependable Systems & Networks*, pages 125–134, Lisbon, Portugal, jun 2009. IEEE.
- [153] J. H. Kim, H. M. Lee, D. Jung, and A. Sadollah. Performance measures of metaheuristic algorithms. In J. H. Kim and Z. W. Geem, editors, *Harmony Search Algorithm*, pages 11–17, Seoul, South Korea, 2016. Springer Berlin Heidelberg.
- [154] M.-J. Kim and I. Han. The discovery of experts’ decision rules from qualitative

-
- bankruptcy data using genetic algorithms. *Expert Systems with Applications*, 25(4):637–646, nov 2003.
- [155] E. Kirkos, C. Spathis, and Y. Manolopoulos. Data Mining techniques for the detection of fraudulent financial statements. *Expert Systems with Applications*, 32(4):995–1003, may 2007.
- [156] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. *Optimization by Simulated Annealing*, page 551–567. MIT Press, Cambridge, MA, USA, 1988.
- [157] H. C. Koh and C. K. Low. Going concern prediction using data mining techniques. *Managerial Auditing Journal*, 19(3):462–476, apr 2004.
- [158] I. Kokolakis, I. Andreadis, and P. Tsalides. Comparison between cellular automata and linear feedback shift registers based pseudo-random number generators. *Microprocessors and Microsystems*, 20(10):643–658, 1997.
- [159] A. Konak, D. W. Coit, and A. E. Smith. Multi-objective optimization using genetic algorithms: A tutorial. *Reliability Engineering & System Safety*, 91(9):992–1007, 2006.
- [160] P. Korosec, K. Tashkova, and J. Silc. The differential Ant-Stigmergy Algorithm for large-scale global optimization. In *IEEE Congress on Evolutionary Computation*, pages 1–8, Barcelona, Spain, jul 2010. IEEE.
- [161] E. Koskivaara. Different Pre-Processing Models for Financial Accounts when Using Neural Networks for Auditing. In *European Conference on Information Systems (ECIS)*, pages 326–332, Vienna, Austria, 2000. AIS eLibrary.
- [162] B. Kosko. Fuzzy cognitive maps. *International Journal of Man-Machine Studies*, 24(1):65–75, jan 1986.
- [163] P. Kotler and K. L. Keller. *Marketing Management*. Pearson Education, Chennai, India, 2012.
- [164] J. R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA, USA, 1992.
- [165] V. Kožený. Genetic algorithms for credit scoring: Alternative fitness function performance comparison. *Expert Systems with Applications*, 42(6):2998–3004, 2015.
- [166] H.-P. Kriegel, M. Schubert, and A. Zimek. Angle-based outlier detection in high-dimensional data. In *Proceedings of the 14th ACM SIGKDD International Con-*

ference on Knowledge Discovery and Data Mining, KDD '08, pages 444–452, New York, NY, USA, 2008. ACM.

- [167] G. J. Krishna and V. Ravi. Modified harmony search applied to reliability optimization of complex systems. In J. H. Kim and Z. W. Geem, editors, *Proceedings of the 2nd International Conference on Harmony Search Algorithm*, volume 382, pages 169–180, Seoul, South Korea, 2015. Springer Verlag.
- [168] G. J. Krishna and V. Ravi. Evolutionary computing applied to customer relationship management: A survey. *Engineering Applications of Artificial Intelligence*, 56:30–59, 2016.
- [169] G. J. Krishna and V. Ravi. Outlier Detection using Evolutionary Computing. In *Proceedings of the International Conference on Informatics and Analytics - ICIA-16*, pages 17:1–17:6, Pudicherry, India, 2016. ACM Press.
- [170] G. J. Krishna and V. Ravi. Hybrid Evolutionary Algorithm for Optimizing Reliability of Complex Systems. In A. Abraham, A. K. Cherukuri, P. Melin, and N. Gandhi, editors, *International Conference on Intelligent Systems Design and Applications*, volume 941, pages 108–119, Vellore, India, 2018. Springer Verlag.
- [171] G. J. Krishna and V. Ravi. Evolutionary Computing Applied to Solve Some Operational Issues in Banks. In S. Datta and J. P. Davim, editors, *Optimization in Industry. Management and Industrial Engineering*, chapter 3, pages 31–53. Springer, Cham, 2019.
- [172] G. J. Krishna and V. Ravi. Feature Subset Selection using Adaptive Differential Evolution: An Application to Banking. In *Proceedings of the ACM India Joint International Conference on Data Science and Management of Data*, pages 157–163, Kolkata, India, 2019. ACM Press.
- [173] G. J. Krishna and V. Ravi. Keystroke based User Authentication using Modified Differential Evolution. In *Proceedings of the 2019 IEEE Region 10 Conference (TENCON)*, pages 739–744, Kochi, Kerala, India, 2019. IEEE.
- [174] G. J. Krishna, V. Ravi, and S. Nagesh Bhattu. Key Generation for Plain Text in Stream Cipher via Bi-Objective Evolutionary Computing. *Applied Soft Computing*, 70:301–317, 2018.
- [175] G. J. Krishna, M. Smruthi, V. Ravi, and B. Shandilya. Differential Evolution Trained Fuzzy Cognitive Map: An Application to Modeling Efficiency in Banking. In A. Abraham, A. K. Cherukuri, P. Melin, and N. Gandhi, editors, *International*

-
- Conference on Intelligent Systems Design and Applications*, volume 941, pages 1–11, Vellore, India, 2018. Springer Verlag.
- [176] V. Kumar and W. Reinartz. *Customer relationship management: Concept, strategy, and tools*. Springer Science & Business Media, Berlin, Heidelberg, 2012.
- [177] R. J. Kuo and C. W. Shih. Association rule mining through the ant colony system for National Health Insurance Research Database in Taiwan. *Computers & Mathematics with Applications*, 54(11–12):1303–1318, 2007.
- [178] G.-C. Lan, T.-P. Hong, and V. S. Tseng. An efficient projection-based indexing approach for mining high utility itemsets. *Knowledge and Information Systems*, 38(1):85–107, jan 2014.
- [179] E. C. Laskari, G. C. Meletiou, Y. C. Stamatiou, and M. N. Vrahatis. *Cryptography and Cryptanalysis Through Computational Intelligence*, pages 1–49. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.
- [180] L. J. Latecki, A. Lazarevic, and D. Pokrajac. Outlier Detection with Kernel Density Functions. In P. Perner, editor, *Machine Learning and Data Mining in Pattern Recognition*, pages 61–75, Leipzig, Germany, 2007. Springer Berlin Heidelberg.
- [181] A. LaTorre, S. Muelas, and J.-M. Peña. A MOS-based dynamic memetic differential evolution algorithm for continuous optimization: a scalability test. *Soft Computing*, 15(11):2187–2199, nov 2011.
- [182] A. LaTorre, S. Muelas, and J.-M. Pena. Multiple Offspring Sampling in Large Scale Global Optimization. In *2012 IEEE Congress on Evolutionary Computation*, pages 1–8, Brisbane, QLD, Australia, jun 2012. IEEE.
- [183] A. LaTorre, S. Muelas, and J.-M. Pena. Large scale global optimization: Experimental results with MOS-based hybrid algorithms. In *2013 IEEE Congress on Evolutionary Computation*, pages 2742–2749, Cancun, Mexico, jun 2013. IEEE.
- [184] A. LaTorre, S. Muelas, and J.-M. Peña. A comprehensive comparison of large scale global optimizers. *Information Sciences*, 316:517–549, 2015.
- [185] T.-S. Lee and I.-F. Chen. A two-stage hybrid credit scoring model using artificial neural networks and multivariate adaptive regression splines. *Expert Systems with Applications*, 28(4):743–752, may 2005.
- [186] T.-S. Lee, C.-C. Chiu, C.-J. Lu, and I.-F. Chen. Credit scoring using the hybrid

-
- neural discriminant technique. *Expert Systems with Applications*, 23(3):245–254, oct 2002.
- [187] H.-q. Li and L. Li. A Novel Hybrid Particle Swarm Optimization Algorithm Combined with Harmony Search for High Dimensional Optimization Problems. In *The 2007 International Conference on Intelligent Pervasive Computing (IPC 2007)*, pages 94–97, Jeju City, South Korea, oct 2007. IEEE.
- [188] X. Li, K. Tang, M. N. Omidvar, Z. Yang, and K. Qin. Benchmark Functions for the CEC’2013 Special Session and Competition on Large Scale Global Optimization. Technical report, 2013.
- [189] C.-W. Lin, T.-P. Hong, and W.-H. Lu. An effective tree structure for mining high utility itemsets. *Expert Systems with Applications*, 38(6):7419–7424, jun 2011.
- [190] J. C. W. Lin, L. Yang, P. Fournier-Viger, T. P. Hong, and M. Voznak. A binary PSO approach to mine high-utility itemsets. *Soft Computing*, 21(17):5103–5121, sep 2017.
- [191] J. C.-W. Lin, L. Yang, P. Fournier-Viger, J. M.-T. Wu, T.-P. Hong, L. S.-L. Wang, and J. Zhan. Mining high-utility itemsets based on particle swarm optimization. *Engineering Applications of Artificial Intelligence*, 55(C):320–330, oct 2016.
- [192] C. Liu and B. Li. Memetic algorithm with adaptive local search depth for large scale global optimization. In *2014 IEEE Congress on Evolutionary Computation (CEC)*, pages 82–88, Beijing, China, jul 2014. IEEE.
- [193] J. Liu and K. Tang. Scaling up covariance matrix adaptation evolution strategy using cooperative coevolution. In H. Yin, K. Tang, Y. Gao, F. Klawonn, M. Lee, T. Weise, B. Li, and X. Yao, editors, *Intelligent Data Engineering and Automated Learning – IDEAL 2013*, pages 350–357, Hefei, China, 2013. Springer Berlin Heidelberg.
- [194] S. Liu, Q. Wang, and S. Lv. Application of Genetic Programming in credit scoring. In *2008 Chinese Control and Decision Conference*, pages 1106–1110, Yantai, Shandong, China, jul 2008. IEEE.
- [195] Y. Liu, W.-k. Liao, and A. Choudhary. A two-phase algorithm for fast discovery of high utility itemsets. In T. B. Ho, D. Cheung, and H. Liu, editors, *Advances in Knowledge Discovery and Data Mining*, pages 689–695, Hanoi, Vietnam, 2005. Springer Berlin Heidelberg.

-
- [196] Y. Liu, Z. Qin, Z. Shi, and J. Chen. Rule Discovery with Particle Swarm Optimization. In C. Chi and K. Lam, editors, *Advanced Workshop on Content Computing (AWCC)*, pages 291–296, ZhenJiang, JiangSu, China, nov 2004. Springer, Berlin, Heidelberg.
- [197] M. Lotfi Shahreza, D. Moazzami, B. Moshiri, and M. Delavar. Anomaly detection using a self-organizing map and particle swarm optimization. *Scientia Iranica*, 18(6):1460–1468, dec 2011.
- [198] J. M. Luna, J. R. Romero, and S. Ventura. Grammar-based multi-objective algorithms for mining association rules. *Data & Knowledge Engineering*, 86:19–37, 2013.
- [199] S. Mabu, K. Hirasawa, and J. Hu. A graph-based evolutionary algorithm: Genetic network programming (gnp) and its extension using reinforcement learning. *Evol. Comput.*, 15(3):369–398, Sept. 2007.
- [200] C. Magnusson, A. Arppe, T. Eklund, B. Back, H. Vanharanta, and A. Visa. The language of quarterly reports as an indicator of change in the company’s financial status. *Information & Management*, 42(4):561–574, may 2005.
- [201] S. Mahdavi, M. E. Shiri, and S. Rahnamayan. Metaheuristics in large-scale global continues optimization: A survey. *Information Sciences*, 295:407–428, 2015.
- [202] Y. Maheshkumar and V. Ravi. A Modified Harmony Search Threshold Accepting Hybrid Optimization Algorithm. In C. Sombattheera, A. Agarwal, S. Udgata, and K. Lavangnananda, editors, *Multi-disciplinary Trends in Artificial Intelligence (MIWAI)*, pages 298–308, Hyderabad, India, 2011. Springer.
- [203] Y. Maheshkumar, V. Ravi, and A. Abraham. A particle swarm optimization-threshold accepting hybrid algorithm for unconstrained optimization. *Neural Network World*, 23(3):191–221, 2013.
- [204] S. Maheshwary, S. Ganguly, and V. Pudi. Deep Secure : A Fast and Simple Neural Network based approach for User Authentication and Identification via Keystroke Dynamics. In B. Drury, M. Madden, B. O’Sullivan, and J. Ueyama, editors, *IWAISe, International Joint Conference on Artificial Intelligence (IJCAI)*, pages 59–66, Melbourne, Australia, 2017. IWAISe proceedings.
- [205] S. Mahfoud and G. Mani. Financial forecasting using genetic algorithms. *Applied Artificial Intelligence*, 10(6):543–566, dec 1996.

-
- [206] T. Mai, B. Vo, and L. T. Nguyen. A lattice-based approach for mining high utility association rules. *Information Sciences*, 399:81–97, aug 2017.
- [207] J. Mantyjarvi, J. Koivumaki, and P. Vuori. Keystroke recognition for virtual keyboard. In *Proceedings. IEEE International Conference on Multimedia and Expo*, volume 2, pages 429–432, Lausanne, Switzerland, 2002. IEEE.
- [208] B. Matthews. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochimica et Biophysica Acta (BBA) - Protein Structure*, 405(2):442–451, oct 1975.
- [209] R. I. McKay, N. X. Hoai, P. A. Whigham, Y. Shan, and M. O’Neill. Grammar-based Genetic Programming: a survey. *Genetic Programming and Evolvable Machines*, 11(3-4):365–396, may 2010.
- [210] A. J. Menezes, S. A. Vanstone, and P. C. V. Oorschot. *Handbook of Applied Cryptography*. CRC Press, Inc., USA, 1st edition, 1996.
- [211] W. Millan, L. Burnett, G. Carter, A. Clark, and E. Dawson. Evolutionary heuristics for finding cryptographically strong s-boxes. In V. Varadharajan and Y. Mu, editors, *Information and Communication Security*, pages 263–274, Sydney, NSW, Australia, 1999. Springer Berlin Heidelberg.
- [212] Misinem, A. A. Bakar, A. R. Hamdan, and M. Z. A. Nazri. A Rough set outlier detection based on Particle Swarm Optimization. In *2010 10th International Conference on Intelligent Systems Design and Applications*, pages 1021–1025, Cairo, Egypt, nov 2010. IEEE.
- [213] S. Modi and S. Elliott. Keystroke Dynamics Verification Using a Spontaneously Generated Password. In *Proceedings 40th Annual 2006 International Carnahan Conference on Security Technology*, pages 116–121, Lexington, KY, USA, oct 2006. IEEE.
- [214] C. Mohan and K. Shanker. Reliability optimization of complex systems using random search technique. *Microelectronics Reliability*, 28(4):513–518, jan 1988.
- [215] A. W. Mohemmed, M. Zhang, and W. N. Browne. Particle swarm optimisation for outlier detection. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation - GECCO ’10*, page 83, New York, New York, USA, 2010. ACM Press.
- [216] D. Molina, M. Lozano, and F. Herrera. MA-SW-Chains: Memetic algorithm based on local search chains for large scale continuous global optimization. In

-
- IEEE Congress on Evolutionary Computation*, pages 1–8, Barcelona, Spain, jul 2010. IEEE.
- [217] M. B. Mzoughia and M. Limam. A multicriterion segmentation approach based on CLV components. In *IEEE 15th International Symposium on Computational Intelligence and Informatics (CINTI)*. IEEE, Budapest, Hungary, 2014.
- [218] A. Nadeem and A. M. Javed. Performance Comparison of Data Encryption Algorithms. In *International Conference on Information and Communication Technologies*, pages 84–89, Karachi, Pakistan, 2005. IEEE.
- [219] R. Nath, B. Rajagopalan, and R. Ryker. Determining the saliency of input variables in neural network classifiers. *Computers & Operations Research*, 24(8):767–773, aug 1997.
- [220] E. W. T. Ngai, L. Xiu, and D. C. K. Chau. Application of data mining techniques in customer relationship management: A literature review and classification. *Expert Systems with Applications*, 36(2):2592–2602, 2009.
- [221] L. T. T. Nguyen, T. Mai, and B. Vo. High utility association rule mining. In P. Fournier-Viger, J. C.-W. Lin, R. Nkambou, B. Vo, and V. S. Tseng, editors, *High-Utility Pattern Mining: Theory, Algorithms and Applications*, pages 161–174. Springer International Publishing, Cham, 2019.
- [222] M. Nguyen, D. Shi, and C. Quek. A nature inspired Ying–Yang approach for intelligent decision support in bank solvency analysis. *Expert Systems with Applications*, 34(4):2576–2587, may 2008.
- [223] I.-S. Oh, J.-S. Lee, and B.-R. Moon. Hybrid genetic algorithms for feature selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(11):1424–1437, nov 2004.
- [224] M. N. Omidvar, X. Li, and X. Yao. Cooperative Co-evolution with delta grouping for large scale non-separable function optimization. In *IEEE Congress on Evolutionary Computation*, pages 1–8, Barcelona, Spain, jul 2010. IEEE.
- [225] C. Ong, J. Huang, and G. Tzeng. Building credit scoring models using genetic programming. *Expert Systems with Applications*, 29(1):41–47, 2005.
- [226] Y.-S. Ong, M. Lim, and X. Chen. Memetic Computation—Past, Present & Future [Research Frontier. *IEEE Computational Intelligence Magazine*, 5(2):24–31, may 2010.

-
- [227] K. R. Opara and J. Arabas. Differential Evolution: A survey of theoretical analyses. *Swarm and Evolutionary Computation*, 44:546–558, feb 2019.
 - [228] M. H. Ozçelik, E. Duman, M. Isik, and T. Cevik. Improving a credit card fraud detection system using genetic algorithm. In *2010 International Conference on Networking and Information Technology*, pages 436–440, Manila, Philippines, 2010. IEEE.
 - [229] C. Paar and J. Pelzl. Stream ciphers. In *Understanding Cryptography: A Textbook for Students and Practitioners*, pages 29–54. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.
 - [230] R. Pacheco, A. Martins, R. Barcia, and S. Khator. A hybrid intelligent system applied to financial statement analysis. In *Proceedings of IEEE 5th International Fuzzy Systems*, pages 1007–1012, New Orleans, LA, USA, 2002. IEEE.
 - [231] B. Padmanabhan and A. Tuzhilin. On the Use of Optimization for Data Mining: Theoretical Interactions and eCRM Opportunities. *Management Science*, 49(10):1327–1343, 2003.
 - [232] P.-F. Pai, M.-F. Hsu, and M.-C. Wang. A support vector machine-based model for detecting top management fraud. *Knowledge-Based Systems*, 24(2):314–321, mar 2011.
 - [233] G. K. Palshikar. The hidden truth—frauds and their control: a critical application for business intelligence. *Intelligent Enterprise*, 5(9):46–51, 2002.
 - [234] H. M. Pandey. Jaya a novel optimization algorithm: What, how and why? In *Proceedings of the 6th International Conference - Cloud System and Big Data Engineering (Confluence)*, pages 728–730, Noida, India, jul 2016. IEEE.
 - [235] E. Papageorgiou, A. Markinos, and T. Gemtos. Fuzzy cognitive map based approach for predicting yield in cotton crop production as a basis for decision support system in precision agriculture application. *Applied Soft Computing*, 11(4):3643–3657, jun 2011.
 - [236] E. I. Papageorgiou. Learning Algorithms for Fuzzy Cognitive Maps—A Review Study. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(2):150–163, mar 2012.
 - [237] E. I. Papageorgiou and J. L. Salmeron. A Review of Fuzzy Cognitive Maps Research During the Last Decade. *IEEE Transactions on Fuzzy Systems*, 21(1):66–79, feb 2013.

-
- [238] A. Papoulis and S. U. Pillai. *Probability, Random Variables, and Stochastic Processes*. McGraw Hill, Boston, USA, fourth edition, 2002.
- [239] R. Parpinelli, H. Lopes, and A. Freitas. Data mining with an ant colony optimization algorithm. *IEEE Transactions on Evolutionary Computation*, 6(4):321–332, aug 2002.
- [240] K. E. Parsopoulos, E. I. Papageorgiou, P. P. Groumpos, and M. N. Vrahatis. Evolutionary Computation Techniques for Optimizing Fuzzy Cognitive Maps in Radiation Therapy Systems. In K. Deb, editor, *Genetic and Evolutionary Computation Conference*, pages 402–413, Seattle, Washington, USA, 2004. Springer, Berlin, Heidelberg.
- [241] S. Peng, Q. Xu, X. B. Ling, X. Peng, W. Du, and L. Chen. Molecular classification of cancer types from microarray data using the combination of genetic algorithms and support vector machines. *FEBS Letters*, 555(2):358–362, dec 2003.
- [242] S. Picek and M. Golub. On evolutionary computation methods in cryptography. In *34th International Convention MIPRO*, pages 1496–1501, Opatija, Croatia, 2011. IEEE.
- [243] I. Polak and M. Boryczka. Genetic Algorithm in Stream Cipher Cryptanalysis. In M. Núñez, N. Nguyen, D. Camacho, and Trawiński B., editors, *7th International Conference Computational Collective Intelligence*, pages 149–158, Madrid, Spain, 2015. Springer, Cham.
- [244] G. Pradeep and V. Ravi. Fuzzy multi-objective association rule mining using evolutionary computation. In A. Kumar, M. K. Dash, S. K. Trivedi, and T. K. Panda, editors, *Handbook of Research on Intelligent Techniques and Modeling Applications in Marketing Analytics*, pages 119–148. IGI Global, Hershey, PA, USA, 2017.
- [245] G. Pradeep, V. Ravi, K. Nandan, B. L. Deekshatulu, I. Bose, and A. Aditya. Fraud Detection in Financial Statements Using Evolutionary Computation Based Rule Miners. In B. Panigrahi, P. Suganthan, and S. Das, editors, *Swarm, Evolutionary, and Memetic Computing (SEMCCO)*, pages 239–250, Bhubaneswar, India, dec 2014. Springer, Cham.
- [246] C. Pramodh, V. Ravi, and T. Nagabhushanam. Indian banks’ productivity ranking via Data Envelopment Analysis and Fuzzy Multi-Attribute Decision-Making

-
- hybrid. *International Journal of Information and Decision Sciences*, 1(1):44, 2008.
- [247] R. S. Gaines; W. Lisowski; S. J. Press; N. Shapiro. Authentication by Keystroke Timing: Some Preliminary Results Same as Report (SAR). Technical report, 1980.
- [248] S. Ramamoorti, A. D. Bailey Jr, and R. O. Traver. Risk assessment in internal auditing: a neural network approach. *International Journal of Intelligent Systems in Accounting, Finance & Management*, 8(3):159–180, sep 1999.
- [249] R. Ratan. Applications of genetic Algorithms in Cryptology. In S. Science, editor, *Third International Conference on Soft Computing for Problem Solving*, pages 821–831, New Delhi, India, 2014. Springer.
- [250] V. Ravi. Optimization of Complex System Reliability by a Modified Great Deluge Algorithm. *Asia-Pacific Journal of Operational Research*, 21(04):487–497, dec 2004.
- [251] V. Ravi, B. Murty, and J. Reddy. Nonequilibrium simulated-annealing algorithm applied to reliability optimization of complex systems. *IEEE Transactions on Reliability*, 46(2):233–239, jun 1997.
- [252] V. Ravi, P. Reddy, and H.-J. Zimmermann. Fuzzy global optimization of complex system reliability. *IEEE Transactions on Fuzzy Systems*, 8(3):241–248, jun 2000.
- [253] V. Ravi and H.-J. Zimmermann. A neural network and fuzzy rule base hybrid for pattern classification. *Soft Computing*, 5(2):152–159, apr 2001.
- [254] S. Ravindran, C. Gautam, and A. Tiwari. Keystroke user recognition through extreme learning machine and evolving cluster method. In *2015 IEEE International Conference on Computational Intelligence and Computing Research (IC-CIC)*, pages 1–5, Madurai, India, dec 2015. IEEE.
- [255] P. Ravisankar and V. Ravi. Financial distress prediction in banks using Group Method of Data Handling neural network, counter propagation neural network and fuzzy ARTMAP. *Knowledge-Based Systems*, 23(8):823–831, dec 2010.
- [256] P. Ravisankar, V. Ravi, G. Raghava Rao, and I. Bose. Detection of financial statement fraud and feature selection using data mining techniques. *Decision Support Systems*, 50(2):491–500, jan 2011.

-
- [257] A. P. Reynolds, G. Richards, B. de la Iglesia, and V. J. Rayward-Smith. Clustering Rules: A Comparison of Partitioning and Hierarchical Clustering Algorithms. *Journal of Mathematical Modelling and Algorithms*, 5(4):475–504, dec 2006.
 - [258] I. Rish. An empirical study of the naive Bayes classifier. In *IJCAI-workshop on empirical methods in artificial intelligence*, pages 41–46, Seattle, Washington, USA, 2001.
 - [259] A. Robins. Sequential learning in neural networks: A review and a discussion of pseudorehearsal based methods. *Intelligent Data Analysis*, 8(3):301–322, 1997.
 - [260] S. Russell and P. Norvig. *Artificial intelligence: a modern approach*. Pearson Education, Harlow, Essex, UK, 3rd edition, 2003.
 - [261] J. Sahoo, A. K. Das, and A. Goswami. An efficient approach for mining association rules from high utility itemsets. *Expert Systems with Applications*, 42(13):5754–5778, aug 2015.
 - [262] F. O. Sameer, M. R. Abu Bakar, A. A. Zaidan, and B. B. Zaidan. A new algorithm of modified binary particle swarm optimization based on the Gustafson-Kessel for credit risk assessment. *Neural Computing and Applications*, 31(2):337–346, feb 2019.
 - [263] K. Sarath and V. Ravi. Association rule mining using binary particle swarm optimization. *Engineering Applications of Artificial Intelligence*, 26(8):1832–1840, sep 2013.
 - [264] W. Schindler. *Random Number Generators for Cryptographic Applications*, pages 5–23. Springer US, Boston, MA, 2009.
 - [265] B. Scholkopf and A. J. Smola. *Learning with kernels : support vector machines, regularization, optimization, and beyond*. MIT Press, Cambridge, MA, USA, 2002.
 - [266] S. C. Shah and A. Kusiak. Data mining and genetic algorithm based gene/SNP selection. *Artificial Intelligence in Medicine*, 31(3):183–196, jul 2004.
 - [267] C. E. Shannon. Communication theory of secrecy Systems. *Bell System Technical Journal*, 28(4):656–715, oct 1949.
 - [268] N. Sharma, N. Arun, and V. Ravi. An ant colony optimisation and Nelder-Mead simplex hybrid algorithm for training neural networks: an application to bankruptcy prediction in banks. *International Journal of Information and Decision Sciences*, 5(2):188, 2013.

-
- [269] P. S. Shelokar, V. K. Jayaraman, and B. D. Kulkarni. Ant algorithm for single and multiobjective reliability optimization problems. *Quality and Reliability Engineering International*, 18(6):497–514, nov 2002.
- [270] Z. Shi, B. Zhang, D. Feng, and W. Wu. Improved key recovery attacks on reduced-round salsa20 and chacha. In T. Kwon, M.-K. Lee, and D. Kwon, editors, *Information Security and Cryptology – ICISC 2012*, pages 337–351, Seoul, Korea (Republic of), 2013. Springer Berlin Heidelberg.
- [271] R. E. Shiffler. Maximum Z Scores and Outliers. *The American Statistician*, 42(1):79–80, feb 1988.
- [272] K.-S. Shin, T. S. Lee, and H.-j. Kim. An application of support vector machines in bankruptcy prediction model. *Expert Systems with Applications*, 28(1):127–135, jan 2005.
- [273] K.-S. Shin and Y.-J. Lee. A genetic algorithm application in bankruptcy prediction modeling. *Expert Systems with Applications*, 23(3):321–328, oct 2002.
- [274] S. K. Smit and A. E. Eiben. Parameter tuning of evolutionary algorithms: Generalist vs. specialist. In C. Di Chio, S. Cagnoni, C. Cotta, M. Ebner, A. Ekárt, A. I. Esparcia-Alcazar, C.-K. Goh, J. J. Merelo, F. Neri, M. Preuß, J. Togelius, and G. N. Yannakakis, editors, *Applications of Evolutionary Computation*, pages 542–551. Springer Berlin Heidelberg, Istanbul, Turkey, 2010.
- [275] J. E. Sohl and A. Venkatachalam. A neural network approach to forecasting model selection. *Information & Management*, 29(6):297–303, dec 1995.
- [276] M. Sokolova and G. Lapalme. A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4):427–437, jul 2009.
- [277] N. Soltani, M. K. Akbari, and M. Sargolzaei Javan. A new user-based model for credit card fraud detection based on artificial immune system. In *The 16th CSI International Symposium on Artificial Intelligence and Signal Processing (AISIP 2012)*, pages 29–33, Shiraz, Fars, Iran, 2012. IEEE.
- [278] W. Song and C. Huang. Discovering High Utility Itemsets Based on the Artificial Bee Colony Algorithm. In D. Phung, V. Tseng, G. Webb, B. Ho, M. Ganji, and L. Rashidi, editors, *Advances in Knowledge Discovery and Data Mining*, pages 3–14, Melbourne, VIC, Australia, jun 2018. Springer, Cham.

-
- [279] W. Song and C. Huang. Mining High Utility Itemsets Using Bio-Inspired Algorithms: A Diverse Optimal Value Framework. *IEEE Access*, 6:19568–19582, mar 2018.
- [280] J. Soto. Statistical testing of random number generators. In *Proceedings of the 22nd National Information Systems Security Conference*, Arlington, Virginia, USA, 1999. NIST.
- [281] T. Sousa, A. Silva, and A. Neves. A Particle Swarm Data Miner. In F. Pires and S. Abreu, editors, *11th Portuguese Conference on Artificial Intelligence, EPIA*, pages 43–53, Beja, Portugal, 2003. Springer, Berlin, Heidelberg.
- [282] C. Spathis, M. Doumpos, and C. Zopounidis. Detecting falsified financial statements: a comparative study using multicriteria analysis and multivariate statistical techniques. *European Accounting Review*, 11(3):509–535, sep 2002.
- [283] D. F. Specht. Probabilistic neural networks. *Neural Networks*, 3(1):109–118, jan 1990.
- [284] N. K. Sreelaja and G. A. V. Pai. Swarm intelligence based key generation for text encryption in cellular networks. In *3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE '08)*, Bangalore, India, jan 2008. IEEE.
- [285] N. K. Sreelaja and G. A. V. Pai. Design of stream cipher for text encryption using Particle Swarm Optimization based key generation. *Journal of Information Assurance and Security*, 4(1):30–41, 2009.
- [286] N. K. Sreelaja and G. A. V. Pai. Swarm intelligence based key generation for stream cipher. *Security and Communication Networks*, 4(2):181–194, aug 2009.
- [287] N. K. Sreelaja and G. A. V. Pai. Stream cipher for binary image encryption using ant colony optimization based key generation. *Applied Soft Computing*, 12(9):2879–2895, sep 2012.
- [288] V. Srikrishna, R. Ghosh, V. Ravi, and K. Deb. Elitist Quantum-Inspired Differential Evolution Based Wrapper for Feature Subset Selection. In A. Bikakis and X. Zheng, editors, *Multi-disciplinary Trends in Artificial Intelligence (MIWAI)*, pages 113–124, Fuzhou, Fujian, China, nov 2015. Springer, Cham.
- [289] M. Srinivas and G. P. Rangaiah. Differential Evolution with Tabu List for Global Optimization and Its Application to Phase Equilibrium and Parameter Estima-

-
- tion Problems. *Industrial and Engineering Chemistry Research*, 46(10):3410–3421, 2007.
- [290] W. Stach, L. Kurgan, and W. Pedrycz. Data-driven Nonlinear Hebbian Learning method for Fuzzy Cognitive Maps. In *IEEE International Conference on Fuzzy Systems (IEEE World Congress on Computational Intelligence)*, pages 1975–1981, Hong Kong, China, jun 2008. IEEE.
- [291] W. Stach, L. Kurgan, W. Pedrycz, and M. Reformat. Genetic learning of fuzzy cognitive maps. *Fuzzy Sets and Systems*, 153(3):371–401, aug 2005.
- [292] W. Stallings. *Cryptography and Network Security: Principles and Practice*. Prentice Hall Press, USA, 6th edition, 2013.
- [293] R. Storn and K. Price. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4):341–359, Dec. 1997.
- [294] H. Su, Y. Yang, and L. Zhao. Classification rule discovery with DE/QDE algorithm. *Expert Systems with Applications*, 37(2):1216–1222, mar 2010.
- [295] G. Sun, R. Zhao, and Y. Lan. Joint operations algorithm for large-scale global optimization. *Applied Soft Computing*, 38:1025–1039, 2016.
- [296] G. G. Sundarkumar and V. Ravi. A novel hybrid undersampling method for mining unbalanced datasets in banking and insurance. *Engineering Applications of Artificial Intelligence*, 37:368–377, jan 2015.
- [297] T. Takahama and S. Sakai. Large scale optimization by differential evolution with landscape modality detection and a diversity archive. In *2012 IEEE Congress on Evolutionary Computation*, pages 1–8, Brisbane, QLD, Australia, jun 2012. IEEE.
- [298] K. Tang, X. Li, P. Suganthan, Z. Yang, and T. Weise. Benchmark functions for the CEC’2010 special session and competition on large scale global optimization. Technical report, Nature Inspired Computation and Applications Laboratory, USTC, China, 2010.
- [299] K. Tang, X. Yáo, P. Suganthan, and C. MacNish. Benchmark functions for the CEC’2008 special session and competition on large scale global optimization. Technical report, USTC, China, 2007.
- [300] L. C. Thomas. A survey of credit and behavioural scoring: forecasting financial

-
- risk of lending to consumers. *International Journal of Forecasting*, 16(2):149–172, apr 2000.
- [301] R. Tibshirani. Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.
- [302] F. A. Tillman, C. L. Hwang, and W. Kuo. *Optimization of systems reliability*. Marcel Dekker, New York, 1980.
- [303] J. Tolvi. Genetic algorithms for outlier detection and variable selection in linear regression models. *Soft Computing*, 8(8):527–533, aug 2004.
- [304] J. Tvrdík. Adaptation in differential evolution: A numerical comparison. *Applied Soft Computing*, 9(3):1149–1155, jun 2009.
- [305] J. Tvrdík, R. Poláková, J. Veselský, and P. Bujok. Adaptive variants of differential evolution: Towards control-parameter-free optimizers. In I. Zelinka, V. Snášel, and A. Abraham, editors, *Handbook of Optimization: From Classical to Modern Approach*, pages 423–449. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.
- [306] G. Üstünkar, S. Özögür-Akyüz, G. W. Weber, C. M. Friedrich, and Y. Aydın Son. Selection of representative SNP sets for genome-wide association studies: a metaheuristic approach. *Optimization Letters*, 6(6):1207–1218, aug 2012.
- [307] J. Uthayakumar and T. Vengattaraman. Swarm intelligence based classification rule induction (CRI) framework for qualitative and quantitative approach: An application of bankruptcy prediction and credit risk analysis. *Journal of King Saud University - Computer and Information Sciences*, In press, 2017.
- [308] D. Van den Poel and B. Larivière. Customer attrition analysis for financial services using proportional hazard models. *European Journal of Operational Research*, 157(1):196–217, 2004.
- [309] A. Vishal and V. Ravi. International Journal of Information Systems and Social Change (IJISSC). *International Journal of Information Systems and Social Change (IJISSC)*, 4(3):33–56, 2012.
- [310] B. Waad, B. M. Ghazi, and L. Mohamed. A New Feature Selection Technique Applied to Credit Scoring Data Using a Rank Aggregation Approach Based on : Optimization , Genetic Algorithm and Similarity. In K.-M. O. Bryson and C. Barclay, editors, *Knowledge Discovery Process and Methods to Enhance Organizational Performance*, pages 347–375. CRC Press, Taylor & Francis Group, Boca Raton, FL, USA, 2014.

-
- [311] S. H. Walker and D. B. Duncan. Estimation of the probability of an event as a function of several independent variables. *Biometrika*, 54(1):167–79, jun 1967.
- [312] C. M. Wang and Y. F. Huang. Evolutionary-based feature selection approaches with new criteria for data mining: A case study of credit approval data. *Expert Systems with Applications*, 36(3 PART 2):5900–5908, 2009.
- [313] H. Wang, Z. Wu, S. Rahnamayan, and D. Jiang. Sequential DE enhanced by neighborhood search for Large Scale Global Optimization. In *IEEE Congress on Evolutionary Computation*, pages 1–7, Barcelona, Spain, jul 2010. IEEE.
- [314] Y. Wang and B. Li. Two-stage based ensemble optimization for large-scale global optimization. In *IEEE Congress on Evolutionary Computation*, pages 1–8, Barcelona, Spain, jul 2010. IEEE.
- [315] D. Wolpert and W. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, apr 1997.
- [316] N. Wong, P. Ray, G. Stephens, and L. Lewis. Artificial immune systems for the detection of credit card fraud: an architecture, prototype and preliminary results. *Information Systems Journal*, 22(1):53–76, 2012.
- [317] J. M. T. Wu, J. Zhan, and J. C. W. Lin. An ACO-based approach to mine high-utility itemsets. *Knowledge-Based Systems*, 116:102–113, jan 2017.
- [318] P. Xuan, M. Guo, J. Wang, C. Wang, X. Liu, and Y. Liu. Genetic algorithm-based efficient feature selection for classification of pre-miRNAs. *Genetics and Molecular Research*, 10(2):588–603, apr 2011.
- [319] G. F. Yang, S. Mabu, K. Shimada, and K. Hirasawa. An evolutionary approach to rank class association rules with feedback mechanism. *Expert Systems with Applications*, 38(12):15040–15048, 2011.
- [320] X.-S. Yang. *Nature-inspired metaheuristic algorithms*. Luniver press, United Kingdom, 2008.
- [321] X.-S. Yang. Harmony Search as a Metaheuristic Algorithm. In Zong Woo Geem, editor, *Music-Inspired Harmony Search Algorithm*, pages 1–14. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.
- [322] Z. Yang, K. Tang, and X. Yao. Large scale evolutionary optimization using cooperative coevolution. *Information Sciences*, 178(15):2985–2999, aug 2008.
- [323] H. Yao and H. J. Hamilton. Mining itemset utilities from transaction databases. *Data & Knowledge Engineering*, 59(3):603–626, dec 2006.

-
- [324] H. Yao, H. J. Hamilton, and C. J. Butz. A Foundational Approach to Mining Itemset Utilities from Databases. In *Proceedings of the 2004 SIAM International Conference on Data Mining*, pages 482–486. Society for Industrial and Applied Mathematics, Philadelphia, PA, apr 2004.
- [325] C.-C. Yeh, D.-J. Chi, and M.-F. Hsu. A hybrid approach of DEA, rough set and support vector machines for business failure prediction. *Expert Systems with Applications*, 37(2):1535–1541, mar 2010.
- [326] I. C. Yeh and C. H. Lien. The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert Systems with Applications*, 36(2):2473–2480, 2009.
- [327] H. Younis, W. Awad, and A. Abd. *Attacking Stream Cipher Systems Using Genetic Algorithm*. PhD thesis, Department of Computer Science, Basrah University, 2000.
- [328] Y. Zeng and J. Ma. Pseudorandom number generators based on evolutionary algorithm. In *Congress on Evolutionary Computation*, pages 1662–1668, Canberra, Australia, 2003. IEEE.
- [329] H. Zenil. *Randomness Through Computation: Some Answers, More Questions*. World Scientific Publishing Co., Inc., USA, 2011.
- [330] G. Zhang, B. Eddy Patuwo, and M. Y. Hu. Forecasting with artificial neural networks:: The state of the art. *International Journal of Forecasting*, 14(1):35–62, mar 1998.
- [331] K. Zhang and B. Li. Cooperative Coevolution with global search for large scale global optimization. In *2012 IEEE Congress on Evolutionary Computation*, pages 1–7, Brisbane, QLD, Australia, jun 2012. IEEE.
- [332] L. Zhang, G. Fu, F. Cheng, J. Qiu, and Y. Su. A multi-objective evolutionary approach for mining frequent and high utility itemsets. *Applied Soft Computing Journal*, 62:974–986, jan 2018.
- [333] Y. Zhang, S. Wang, P. Phillips, and G. Ji. Binary PSO with mutation operator for feature selection using decision tree applied to spam detection. *Knowledge-Based Systems*, 64:22–31, jul 2014.
- [334] S.-Z. Zhao, P. N. Suganthan, and S. Das. Dynamic multi-swarm particle swarm optimizer with sub-regional harmony search. In *IEEE Congress on Evolutionary Computation*, pages 1–8, Barcelona, Spain, jul 2010. IEEE.

-
- [335] X. Zhao, J. Zeng, Y. Gao, and Y. Yang. Particle Swarm Algorithm for Classification Rules Generation. In *Sixth International Conference on Intelligent Systems Design and Applications*, volume 2, pages 957–962, Jinan, China, oct 2006. IEEE.
- [336] M. Zhong, M. Zhong, J. Hecker, I. Maidhof, P. Shibly, M. Georgiopoulos, G. Anagnostopoulos, M. Mollaghasemi, and S. Richie. Probabilistic Neural Network: Comparison of the Cross-validation Approach and a Fast Heuristic to Choose the Smoothing Parameters, 2005.
- [337] Y. Zhong, Y. Deng, and A. K. Jain. Keystroke dynamics for user authentication. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 117–123, Providence, RI, USA, jun 2012. IEEE.
- [338] W. Zhou and G. Kapoor. Detecting evolutionary financial statement fraud. *Decision Support Systems*, 50(3):570–575, feb 2011.
- [339] S. Zida, P. Fournier-Viger, J. C. W. Lin, C. W. Wu, and V. S. Tseng. EFIM: A highly efficient algorithm for high-utility itemset mining. In G. Sidorov and S. Galicia-Haro, editors, *Mexican International Conference on Artificial Intelligence*, volume 9413, pages 530–546, Cuernavaca, Mexico, 2015. Springer Verlag.
- [340] M. Zihayat and A. An. Mining top-k high utility patterns over data streams. *Information Sciences*, 285:138–161, nov 2014.
- [341] A. Zimek, E. Schubert, and H.-P. Kriegel. A survey on unsupervised outlier detection in high-dimensional numerical data. *Statistical Analysis and Data Mining*, 5(5):363–387, oct 2012.
- [342] E. Zitzler, M. Laumanns, and L. Thiele. Spea2: Improving the strength pareto evolutionary algorithm. *TIK-report*, 103, 2001.

List of Journal Publications

1. Gutha Jaya Krishna and Vadlamani Ravi. Evolutionary computing applied to customer relationship management: A survey. *Engineering Applications of Artificial Intelligence*, 56:30–59, 2016.
2. Gutha Jaya Krishna, Vadlamani Ravi, and S. Nagesh Bhattu. Key Generation for Plain Text in Stream Cipher via Bi-Objective Evolutionary Computing. *Applied Soft Computing*, 70:301–317, 2018.

List of Book Chapter Publications

1. Gutha Jaya Krishna and Vadlamani Ravi. Evolutionary Computing Applied to Solve Some Operational Issues in Banks. In Shubhabrata Datta and J. Paulo Davim, editors, *Optimization in Industry. Management and Industrial Engineering*, chapter 3, pages 31–53. Springer, Cham, 2019.

List of Conference Publications

1. Gutha Jaya Krishna and Vadlamani Ravi. Modified harmony search applied to reliability optimization of complex systems. In Joong Hoon Kim and Zong Woo Geem, editors, *Proceedings of the 2nd International Conference on Harmony Search Algorithm*, volume 382, pages 169–180, Seoul, South Korea, 2015. Springer Verlag.
2. Gutha Jaya Krishna and Vadlamani Ravi. Outlier Detection using Evolutionary Computing. In *Proceedings of the International Conference on Informatics and Analytics - ICIA-16*, pages 17:1–17:6, Pudicherry, India, 2016. ACM Press.
3. Gutha Jaya Krishna and Vadlamani Ravi. Hybrid Evolutionary Algorithm for Optimizing Reliability of Complex Systems. In Ajith Abraham, Aswani Kumar Cherukuri, Patricia Melin, and Niketa Gandhi, editors, *International Conference on Intelligent Systems Design and Applications*, volume 941, pages 108–119, Vellore, India, 2018. Springer Verlag.
4. Gutha Jaya Krishna, Meesala Smruthi, Vadlamani Ravi, and Bhamidipati Shandilya. Differential Evolution Trained Fuzzy Cognitive Map: An Application to Modeling Efficiency in Banking. In Ajith Abraham, Aswani Kumar Cherukuri, Patricia Melin, and Niketa Gandhi, editors, *International Conference on Intelligent Systems Design and Applications*, volume 941, pages 1–11, Vellore, India, 2018. Springer Verlag.
5. Gutha Jaya Krishna and Vadlamani Ravi. Feature Subset Selection using Adaptive Differential Evolution: An Application to Banking. In *Proceedings of the ACM India Joint International Conference on Data Science and Management of Data*, pages 157–163, Kolkata, India, 2019. ACM Press.
6. Gutha Jaya Krishna and Vadlamani Ravi. Keystroke based User Authentication using Modified Differential Evolution. In *Proceedings of the 2019 IEEE Region 10 Conference (TENCON)*, pages 739–744, Kochi, Kerala, India, 2019. IEEE.

7. Gutha Jaya Krishna and Vadlamani Ravi. Anomaly Detection using Modified Differential Evolution: An Application to Banking and Insurance. In *Proceedings of 11th International Conference Soft Computing and Pattern Recognition (SoCPaR)*, Hyderabad, India, 2019. Springer.

Under Review

1. Gutha Jaya Krishna and Vadlamani Ravi. Mining Top High Utility Association Rules using Differential Evolution. *Under Review in Engineering Applications of Artificial Intelligence (Elsevier)*.
2. Gutha Jaya Krishna and Vadlamani Ravi. High Utility Itemset Mining using Binary Differential Evolution: An Application to Customer Segmentation. *Under Review in Expert Systems with Applications (Elsevier)*.
3. Gutha Jaya Krishna and Vadlamani Ravi. Rule-based Classification using Differential Evolution: An Application to Banking. *Under Review in Applied Soft Computing (Elsevier)*.
4. Gutha Jaya Krishna and Vadlamani Ravi. Hybrid of Harmony Search and Differential Evolution for Large Scale Global Optimization. *Under Review in Memetic Computing (Springer)* .

Appendix A

Overview of Evolutionary Computing Techniques

1. **Genetic Algorithm (GA)** [110] can be defined as a meta-heuristic that imitates natural selection, one of Darwin's principles. A population of solutions are used to search through the solution space. There are four steps within the usage of GA: 1. Encoding 2. Selection 3. Genetic Operators 4. Termination

Advantages: 1. GA can be used with the variable length chromosome encoding and the overall fitness of the population never diminishes as the generations increase. 2. GA can solve multi-modular multi-dimensional, non-differential, non-continuous, and non-linear problems.

Disadvantages: 1. GA converges slowly 2. There is no evidence that GA converging to form the global optimum.

2. **Genetic Programming (GP)** [164] is best defined as a strategy that is EC based and can identify computer programs that are run on the computers which have the capability to resolve problems of optimization. GP is defined on directions of fitness function of how well a computer program has performed a job. It is a specialization of GA where every individual is a program. It is an optimization procedure used to upgrade a population of programs as per a fitness landscape controlled by a program's capacity to perform a given computational job.

Advantages: 1. Information that is dependent on the domain is not required. 2. It produces solutions which resemble programs.

Disadvantages: 1. Experimentation i.e. trial and error is the basis of tuning parameters.

3. **Genetic Network Programming (GNP)** [199] This is a successor to Genetic Programming. Using graphs over trees is the basis of creating GNP. Successful and productive management of situations that are dynamic in nature is the prime objective. Characteristics of GNP are; 1. GNP programs are created out of various nodes by the associations with one another. 2. Node reuse in GNP is encouraged by graph structure. 3. There is no terminal node and node associations execute GNP. As a result, moves of the past node are used to determine and successive movement which is used as a memory function.

Advantages: 1. The focal point of GNP is a programmed alternative of the automatic selection significant number of inputs and actions. 2. Problems that are complex and ever changing, can be solved using Genetic Network Programming.

Disadvantages: 1. Its gene structure may end up being unreasonably complex after subsequent iterations. 2. The GNP requirement of memory to locate the optimum is of maximum size of the memory.

4. **Particle Swarm Optimization (PSO)** [150] May be described as an optimization method that is evolutionary and is based on population. Simple heuristics are used to update the position of a particle moving around in search space. The best known location impacts every particles position. However, it is at the same time that the particle is guided to the local best known position. This process mentioned above is instrumental in updating the position of the swarm to the best position.

Advantages: 1. Few parameters to tune and easy to code. 2. Can be run in parallel. 3. Even though they were designed originally for problem that need continuous optimization, there have been reported discrete versions as well.

Disadvantages: 1. PSO gets caught at local optima some times. 2. It isn't good at diversification.

5. **Ant Colony Optimization (ACO)** [67] is a meta-heuristic which mimics optimal way of ants to find food. To start with, once the insect finds a food supply, it leaves "markers" (pheromones) to demonstrate the trail prompting to

the food. Currently once varied ants ‘smell’ the markers, they pursue the trail with a degree of probability. In the event that they’re fruitful, they populate the route with their markers as they bring about the nourishment to the source. This procedure is repeated till the food supply exhausts.

Advantages: 1. Parallelism can be achieved since it’s inherent in the optimization. 2. Better solutions are quickly discovered by positive feedbacks.

Disadvantages: 1. Likelihood with which the ants move changes upon generations. 2. Exploration is purely experimental and not based on theory. 3. The exact time of convergence is difficult to decide, despite the fact that there is guaranteed convergence.

6. **Simulated Annealing (SA)** [156] is understood as a thermal procedure for obtaining low energy states for a metal under the heat bath . The SA involves the two phases: 1. Extend the temperature of the heat bath to the most extreme degree, so that the metal melts. 2. Then, let the metal be cooled down so as to achieve metals low energy stable state. This stage is achieved after lowering the temperature of the heat bath.

Advantages: 1. It is able to keep away from local optima. 2. It guarantees a solution that is optimal. 3. Coding is simple.

Disadvantages: 1. Analysis of the energy state for the problem is required for optimization. 2. If the function that defines objective is difficult to calculate, then finding optima is tedious, given that its speed is moderate. 3. It is a point-based search.

7. **Artificial Immune System (AIS)** [76] is a method inspired by the immune system of humans. Taking cues from the human immune systems, the AIS algorithms are developed. The AIS algorithms exploit the immune system’s qualities of learning and memory.

Advantages: 1. Self-adaptive and driven by information. 2. They are good at estimation of function. 3. Nonlinear models, which makes them adapt to search spaces that are complex.

Disadvantages: 1. Difficult to fine-tune parameters to utilize AIS.

8. **Grammar Guided Genetic Programming (G3P)** [209] is described as an extension of GP. The only difference with GP lies in the fact that they make use

of context free grammars (CFG) for the conceivable answers. All solutions are produced by these CFG's. Encoding is done using derivation trees. For given problems as sentences, building up the formal meaning of the syntactic problem constraints, and utilizing the derivation trees for every sentence to encode these solutions is performed.

Advantages: 1. It utilizes adaptive techniques to search out for the solutions in the search space with restrictions.

Disadvantages: 1. Necessity of defining grammar is the biggest challenge in using G3P.

9. **Migrating Birds Optimization (MBO)** [87] takes cue from the 'V' flight arrangement of birds, which is led by an leader. This is a meta-heuristic methodology in light of the V flight arrangement of the migrating birds which ended up as being a powerful development in energy minimization. The MBO technique is a search procedure based on neighborhood. It begins with various initializations comparing to birds in a 'V' development. Beginning with the first arrangement (with respect to leader bird) and advancing on the lines towards the end. Every solution is attempted to be enhanced by its neighbor solutions.

Advantages: 1. Parallelism can be achieved. 2. A benefit mechanism is observed between solutions.

Disadvantages: 1. Characterized as a neighborhood search process. 2. Speed is moderate.

10. **Data mining by evolutionary learning (DMEL)** [22] is utilized when there is a requirement for precision classification and the assessment needs to be carried out for every forecast. This algorithm is an approach that is based on evolutionary computing. 1. Higher order rules are obtained after the first order rule set is generated using probability. 2. It then identifies the interesting rules. 3. A characterization of the fitness of the chromosome is determined by the probability. 4. Likelihood estimates the probability of predictions made.

Advantages: 1. DMEL can discover interesting rules. 2. Superior to non-EC algorithms as it creates rules.

Disadvantages: 1. Poor forecast capability if the datasets are large.

-
11. **Threshold Accepting (TA)** [85] is a deterministic variant SA. This is best described as a search technique that is point based and local in nature which commences from a solution which is feasible and mostly random. A point to be noted is that this moves further in the neighbouring spaces in search for a better solution and if found, it moves from its position to the other. By virtue of accepting other solutions that are local in nature, threshold acceptance moves out of the local minima provided they are not good enough or worse than the present solutions.

Advantages: 1. It overcomes stopping at local optima. 2. Implementation is easy. 3. Since there is no usage of probabilistic acceptance which is in the case of SA, it demonstrates efficiency in computation.

Disadvantages: 1. Being a local search technique, convergence rates are slow.

12. **Differential Evolution (DE)** [293] is defined as an evolutionary method that is stochastic, parallel, and direct. Each iteration has new solutions prepared from the set of solutions in the population known as mutation. The solutions obtained from mutations are joined together with the target solution, to form a crossover process. Finally, on the basis of fitness, the solutions get chosen. (Refer Appendix B for further details)

Advantages: 1. Will be able to search non-continuous, non-differential, and multi-modular spaces. 2. DE is Characterized by robustness and quick convergence. 3. Regardless of the fact that it is meant to take care of real value problems, it can very well can be applied to discrete valued variables as well.

Disadvantages: 1. If there are non-real valued variables, it becomes mandatory to define an encoding scheme.

13. **Firefly Algorithm (FFA)** [320] is described by the following steps: 1. Fireflies get attracted to other fireflies. 2. Attractiveness depends on brightness and distance. If there is no other firefly near to one, then it will move randomly. 3. The brightness of the firefly depends upon the decision space.

Advantages: 1. There is capability of handling multi-modular solutions. 2. There is diversity within the solutions of the population.

Disadvantages: 1. It tend to get trapped at local optima. 2. Parameters are

fixed and cannot change dynamically. 3. There is no memorizing capability of better solutions.

14. **Tabu Search (TS)** [107, 108] is a point based local search model which will scan the neighbourhood optimally. Best categorized as a dynamic technique for searching the neighbourhood. It finds solutions of problems pertaining to combinatorial (limited solution set) optimization. It likewise utilizes adaptable memory to confine the following solution decisions to some subset of the neighbourhood of a current solution.

Advantages: 1. There is process of exclusion of solutions that are not improving from local optima. 2. Find usage in both continuous and then discrete spaces.

Disadvantages: 1. Evaluation is time consuming given the long list of parameters. 2. Long runs common. 3. Miss the optimum that is global.

15. **Multi-Objective Genetic Algorithm (MOGA)** [159] finds an arrangement of multiple solutions that are non-dominated. It is the characteristic of a Genetic algorithm to search through for non-linear, discontinuous and multi-modal search spaces. Many multi modal genetic algorithms don't require scaling, measuring of objectives or organizing. Therefore, the best means to cope up with multi objective optimization is the genetic algorithm.

Advantages: 1. Can be implemented effectively for combinatorial optimization problems. 2. Suitable if a spread of Pareto-optimal solutions are needed in the objective space

Disadvantages: 1. MOGA is biased towards the shape of the Pareto-Optimal front and to the density of solutions. 2. Convergence is slow or not good to locate a decent spread in the Pareto-Optimal front.

16. **Non-dominated Sorting Genetic Algorithm-II (NSGA-II)** [78]. There are challenges of the EA method that are overcome by the NSGA-II. Generally speaking, NSGA-II implements a quick non dominated methodology for sorting with less computational complexity and time. There are also provisions for creating a mating pool where the best populations of offspring's are selected by combining the child and parent population. (Refer Appendix B for further details)

Advantages: 1. There is preservation of diversity. 2. Complexity is $O(MN^2)$
3. Elitism does no longer allow a discovered Pareto-Optimal solution to be discarded.

Disadvantages: 1. Convergence is limited by Crowding. 2. $2N$ is the defined size for non dominated sorting.

17. **Strength Pareto Evolutionary Algorithm-2 (SPEA-2)** [342] is an improved variation of SPEA. SPEA-2 utilizes its ancestor for a fine-grained fitness assignment, a density estimation strategy, and an enhanced archive truncation.
1. An, improved fitness assignment that considers for every individual what number of individuals it dominates and dominated by. 2. A nearest neighbor density estimation strategy is fused which permits a more exact direction of the search process. 3. New archive truncation techniques ensure the conservation of boundary solutions.

Advantages: 1. Given the Pareto-Optimal front, it makes it easier for solution realization. 2. Employment of methodology of clustering. 3. Is superior to its previous versions on all problems.

Disadvantages: 1. There is an additional parameter that is introduced i.e. 'M' which is the size of the external population. There is a harmony between these external and normal sizes of population. These factors are essential in the working effectiveness of SPEA-2 and generation of a Pareto-Optimal front.

Appendix B

State-of-the-Art Evolutionary Computing Techniques Employed

B.1 Harmony Search

Harmony Search (HS) is a process imitating metaheuristic presented by Geem et al. in [104]. It imitates the technique used by a musician to develop new beats of music from old music or developing new music from nothing. The HS employs four parameters namely PAR, HMCR, BW and HMS. In HS, harmony update is a significant part of HS that produces new solution vectors. The harmony update is performed in a systematic manner using following steps:

1. It picks up the solution from harmony memory if the generated random number lies between 0 and 1 i.e. it is below HMCR.

$$x_{i,j}^{new} = x_{i,j}$$

where $x_{i,j} \in \{x_{1,j}, x_{2,j}, x_{3,j}, \dots, x_{HMS,j}\}$

2. (a) Look into the local neighbourhood of selected solution vector if the generated number is less than PAR.

$$x_{i,j}^{new} = x_{i,j} \pm rand(0, 1) * BW$$

-
- (b) Else if the acquired number is bigger than the PAR, stick with the generated solution vector.
 - 3. If the generated number is higher than HMCR, pick the random value from the domain of decision variable.

$$x_{i,j}^{new} = l_{i,j} + (u_{i,j} - l_{i,j}) * rand(0, 1)$$

The generic procedure for HS are given below:

1. Decide on the size of harmony memory (HM).
2. Start with HM number of solution vectors and measure the related fitness value.
3. HMCR is fixed between 0.7 and 0.99. The value of PAR is fixed between 0.0 and 0.5.
4. The process of updating is performed i.e. old vectors are used or new vectors are generated from the older ones or totally new unique vectors are generated by using the HMCR and PAR.
5. The fitness value is analyzed at this point. If the fitness value is better than worst solution, then substitute the worst solution vectors and the fitness values with the newly generated solution.
6. The step number 3, 4, and 5 should be repeated until the maximum of iterations are met.

B.2 Differential Evolution

It is a population based evolutionary metaheuristic. It is designed for real input parameter-based optimization problems. There are four major parts of DE i.e. the first step is the initialization or assignment of all solutions with random values; mutate the solutions; recombine the solutions and select the final solution (replace the old solution with new one if it is better than the previous one). The new solutions are generated from the existing solution by mutation and crossover. In selection, the newly found solution is analyzed and best suitable solution is selected. There are two

parameters in DE namely CR and F. The working of DE is elaborated in the following steps:

Step-1: Initialization

1. Initialize ‘NP’, the number solution vectors of dimensionality size, ‘Dim’.
2. Check whether the solution vectors are within bounds.
3. Compute the fitness values of the ‘NP’ number of solution vectors.

Step-2: Mutation

4. Let x_1 , x_2 , x_3 and $x^{current}$ be the three unique solution vectors chosen randomly from the population ‘NP’.
5. Perform the mutation using the following equation:

$$u'_i = x_{1,i} + F * (x_{2,i} - x_{3,i})$$

The above equations are of a variant of DE i.e. DE/rand/1/bin. Differential weight ‘F’ is between 0 and 2

Step-3: Recombination

6. Using the mutation vector and crossover rate parameter CR recombination is performed as follows:

$$v_i = \begin{cases} u_i & \text{if } (rand(0,1) \leq CR \mid i = I_{rand}) \\ x_i^{current} & \text{if } (rand(0,1) > CR \text{ and } i \neq I_{rand}) \end{cases}$$

CR is between 0 and 1. I_{rand} is a random dimension between 1 to Dim and I_{rand} is used so that v and $x^{current}$ are different.

Step-4: Selection

7. If $Fitness(v) \leq Fitness(x^{current})$ replace current vector with solution vector ‘v’ in the population.
8. Else do not replace the current vector.
9. Repeat steps 4-8, until a maximum number of iterations.

B.3 Non-dominated Sorting Genetic Algorithm-II (NSGA-II)

NSGA-II [78] is a modified version of NSGA procedure, which uses a better sorting algorithm, incorporates elitism, and does not require niching or any other parameter. In NSGA-II, a population is randomly initialized based on the domain of the decision variables. Then, the population is sorted based on non-domination of solutions resulting in various hierarchical fronts. Once the non-dominated sorting is completed, a crowding distance value is assigned to each solution depending on its closeness to other population members in the objective space. The individuals are selected based on their non-dominated rank and the crowding distance, in that order. Thereafter, recombination and mutation operators are applied to the selected parent solutions to create the new offspring population. This process continues until a termination criterion is reached. At every subsequent generation t , the offspring population Q_t and the current generation population P_t are combined and selection is invoked to select the solutions for the next generation. Elitism is taken care of because all the previous and current best solutions are passed on to the next-generation. Then, the population is sorted based on non-domination. Thereafter, new generation is formed by each front until the population size exceeds the current population size. In case the addition of all the solutions in the front F_i results in the population P_{t+1} exceeding Z , then solutions in front F_i are selected based on their crowding distance in the descending order until the population size is Z . The process is repeated to form subsequent generations. This process is depicted in Figure B.1.

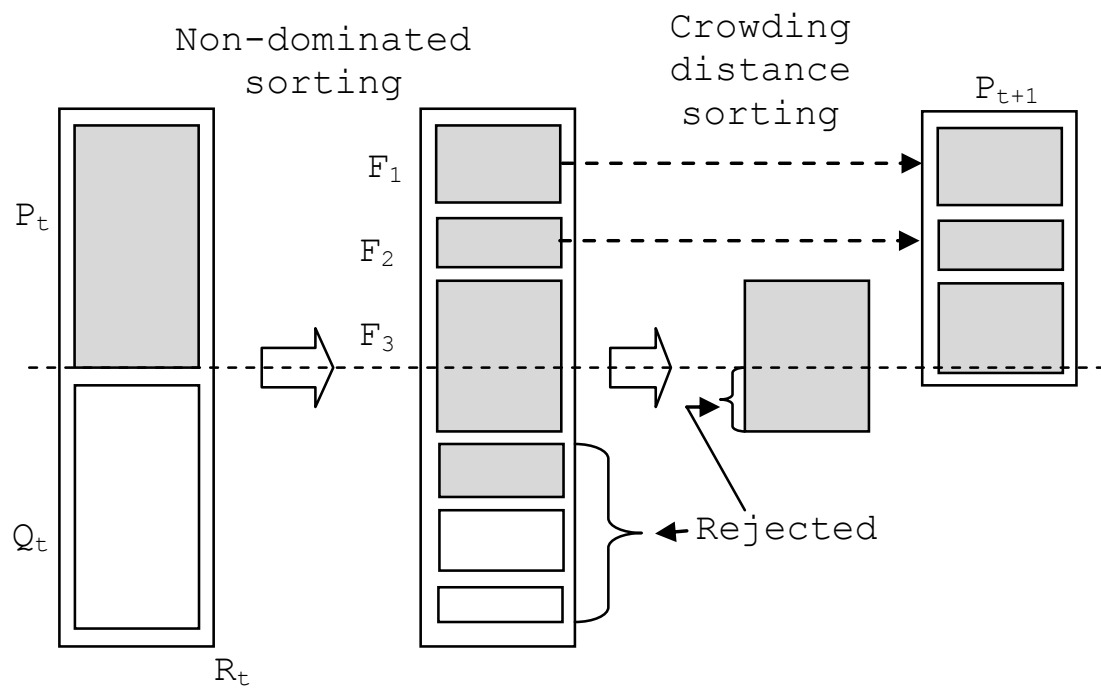


Figure B.1: Non-dominated Sorted Genetic Algorithm-II (NSGA-II)

<u>fast-non-dominated-sort(P)</u>	
for each $p \in P$	
$S_p = \emptyset$	
$n_p = 0$	
for each $q \in P$	
if $(p \prec q)$ then	If p dominates q
$S_p = S_p \cup \{q\}$	Add q to the set of solutions dominated by p
else if $(q \prec p)$ then	
$n_p = n_p + 1$	Increment the domination counter of p
if $n_p = 0$ then	p belongs to the first front
$p_{\text{rank}} = 1$	
$\mathcal{F}_1 = \mathcal{F}_1 \cup \{p\}$	
$i = 1$	Initialize the front counter
while $\mathcal{F}_i \neq \emptyset$	
$Q = \emptyset$	Used to store the members of the next front
for each $p \in \mathcal{F}_i$	
for each $q \in S_p$	
$n_q = n_q - 1$	
if $n_q = 0$ then	q belongs to the next front
$q_{\text{rank}} = i + 1$	
$Q = Q \cup \{q\}$	
$i = i + 1$	
$\mathcal{F}_i = Q$	

Figure B.2: Steps for Non-dominated Sorting

<u>crowding-distance-assignment($\mathcal{I}$)</u>	
$l = \mathcal{I} $	number of solutions in $\mathcal{I}$
for each i , set $\mathcal{I}[i]_{\text{distance}} = 0$	initialize distance
for each objective m	
$\mathcal{I} = \text{sort}(\mathcal{I}, m)$	sort using each objective value
$\mathcal{I}[1]_{\text{distance}} = \mathcal{I}[l]_{\text{distance}} = \infty$	so that boundary points are always selected
for $i = 2$ to $(l - 1)$	for all other points
$\mathcal{I}[i]_{\text{distance}} = \mathcal{I}[i]_{\text{distance}} + (\mathcal{I}[i + 1].m - \mathcal{I}[i - 1].m) / (f_m^{\text{max}} - f_m^{\text{min}})$	

Figure B.3: Steps for Crowding Distance

$R_t = P_t \cup Q_t$	combine parent and offspring population
$\mathcal{F} = \text{fast-non-dominated-sort}(R_t)$	$\mathcal{F} = (\mathcal{F}_1, \mathcal{F}_2, \dots)$, all nondominated fronts of R_t
$P_{t+1} = \emptyset$ and $i = 1$	
until $ P_{t+1} + \mathcal{F}_i \leq N$	until the parent population is filled
crowding-distance-assignment($\mathcal{F}_i$)	calculate crowding-distance in $\mathcal{F}_i$
$P_{t+1} = P_{t+1} \cup \mathcal{F}_i$	include i th nondominated front in the parent pop
$i = i + 1$	check the next front for inclusion
Sort($\mathcal{F}_i, \prec_n$)	sort in descending order using $\prec_n$
$P_{t+1} = P_{t+1} \cup \mathcal{F}_i[1 : (N - P_{t+1})]$	choose the first $(N - P_{t+1})$ elements of $\mathcal{F}_i$
$Q_{t+1} = \text{make-new-pop}(P_{t+1})$	use selection, crossover and mutation to create a new population Q_{t+1}
$t = t + 1$	increment the generation counter

Figure B.4: Steps for Non-dominated Sorted Genetic Algorithm-II (NSGA-II)

Appendix C

Additional Supporting Results of IMHS+MDE on CEC 2013 LSGO Benchmark (Chapter 4)

The results of the proposed algorithm along with the other compared algorithms are presented in Tables C.1 and C.2. The functions, wherein an algorithm beats the other algorithms, are marked in bold. The proposed algorithm is run for fixed FEs as per the CEC 2013 LSGO benchmark (i.e. 3 million FEs for each of the 25 independent runs). Median value of 25 independent runs (i.e. 13th) is used for the statistical analysis as per the benchmark. The winner is decided based on the median value of 25 independent runs (i.e. 13th in the increasing order) for each function.

Table C.1: Results of CEC 2010, CEC 2012, CEC 2013 and SOCO 2011 LSGO winners and proposed hybrid algorithms applied on CEC 2013 LSGO Benchmarks (F1, F2, F3, F4, F5, F6, F7, F8)

Algorithm	Metrics	F1	F2	F3	F4	F5	F6	F7	F8
IMHS+MDE	Best	943082.9	1454.525	20.65726	4.64E+09	590036.6	1058137	92875120	3.03E+13
	Median	4595495	1564.223	20.68594	9.75E+09	932950.4	1059663	1.42E+08	1.53E+14
	Worst	13521885	1642.277	20.70005	1.2E+10	1193703	1061637	2.36E+08	3.26E+14
	Mean	4654139	1556.606	20.6803	9.25E+09	947524.7	1060032	1.5E+08	1.75E+14
	Stddev	3127253	54.91001	0.014517	1.96E+09	150477.7	1049.047	41171630	8.46E+13
MA-SW-Chains	Best	3.03E-14	9.52E+02	2.63E-13	7.20E+08	1.07E+06	9.54E-01	3.36E+05	3.14E+13
	Median	6.15E-13	1.13E+03	6.79E-13	2.70E+09	1.98E+06	6.24E+02	3.99E+06	4.65E+13
	Worst	4.97E-12	1.50E+03	1.24E-12	9.06E+09	7.59E+06	5.79E+04	4.40E+06	6.42E+13
	Mean	1.14E-12	1.18E+03	6.78E-13	3.80E+09	2.26E+06	1.07E+04	3.78E+06	4.63E+13
	Stddev	1.28E-12	1.84E+02	2.28E-13	2.70E+09	1.36E+06	2.09E+04	8.46E+05	9.18E+12
2S-Ensemble	Best	0.00E+00	1.04E-26	2.81E-13	4.18E+09	5.89E+05	1.59E+05	6.88E+05	1.21E+14
	Median	0.00E+00	9.95E-01	3.98E-13	1.13E+10	1.28E+06	2.00E+05	1.54E+06	3.94E+14
	Worst	4.55E-23	5.97E+00	5.72E-13	1.68E+10	1.79E+06	2.25E+05	4.17E+06	6.82E+14
	Mean	2.35E-24	1.39E+02	4.20E-13	1.10E+10	1.33E+06	1.94E+05	1.90E+06	3.85E+14
	Stddev	9.16E-24	1.62E+00	7.82E-14	3.44E+09	2.50E+05	1.64E+04	1.14E+06	1.39E+14
MOS-SOCO2011	Best	0.00E+00	1.10E+01	0.00E+00	3.14E+09	8.51E+06	9.80E+05	9.05E+05	3.92E+14
	Median	0.00E+00	1.96E+01	0.00E+00	1.20E+10	1.09E+07	9.85E+05	1.23E+07	1.13E+15
	Worst	0.00E+00	2.72E+01	0.00E+00	3.24E+10	1.44E+07	9.91E+05	2.20E+08	8.82E+15
	Mean	0.00E+00	1.93E+01	0.00E+00	1.34E+10	1.11E+07	9.85E+05	2.31E+07	1.64E+15
	Stddev	0.00E+00	4.16E+00	0.00E+00	7.69E+09	1.79E+06	3.22E+03	4.42E+07	1.66E+15
jDElscep	Best	0.00E+00	0.00E+00	2.56E-13	1.80E+09	3.24E+06	3.00E-07	9.32E+08	3.51E+13
	Median	0.00E+00	9.95E-01	2.77E-13	7.18E+09	4.83E+06	1.99E-01	3.17E+09	1.05E+14
	Worst	0.00E+00	1.10E+01	2.91E-13	2.05E+10	6.17E+06	5.66E+03	5.89E+09	2.44E+14
	Mean	0.00E+00	1.92E+00	2.78E-13	8.37E+09	4.78E+06	2.42E+02	3.29E+09	1.18E+14
	Stddev	0.00E+00	1.92E+00	2.78E-13	8.37E+09	4.78E+06	2.42E+02	3.29E+09	1.18E+14

Table C.1: Results of CEC 2010, CEC 2012, CEC 2013 and SOCO 2011 LSGO winners and proposed hybrid algorithms applied on CEC 2013 LSGO Benchmarks (F1, F2, F3, F4, F5, F6, F7, F8) (Continued)

	Stddev	0.00E+00	2.67E+00	8.41E-15	4.15E+09	7.60E+05	1.13E+03	1.24E+09	5.28E+13
GaDE	Best	0.00E+00	0.00E+00	2.17E-13	4.68E+09	8.72E+06	1.02E+06	1.97E+09	4.87E+13
	Median	0.00E+00	0.00E+00	2.56E-13	1.01E+10	1.02E+07	1.03E+06	2.68E+09	1.57E+14
	Worst	0.00E+00	1.72E-02	3.16E-13	2.79E+10	1.18E+07	1.03E+06	3.23E+09	8.15E+14
	Mean	0.00E+00	6.88E-04	2.61E-13	1.12E+10	1.02E+07	1.03E+06	2.67E+09	2.08E+14
	Stddev	0.00E+00	3.44E-03	2.95E-14	5.17E+09	9.06E+05	2.03E+03	3.90E+08	1.75E+14
MOS-CEC2012	Best	0.00E+00	1.57E+03	2.09E-12	9.45E+07	1.69E+07	9.85E+05	2.50E+03	1.00E+11
	Median	0.00E+00	1.75E+03	2.35E-12	3.41E+08	2.37E+07	9.88E+05	1.78E+05	5.31E+11
	Worst	0.00E+00	1.94E+03	2.39E+00	5.65E+08	3.43E+07	9.96E+05	3.06E+05	1.83E+12
	Mean	0.00E+00	1.75E+03	6.02E-01	3.07E+08	2.41E+07	9.89E+05	1.46E+05	7.18E+11
	Stddev	0.00E+00	1.06E+02	8.77E-01	1.47E+08	3.93E+06	2.67E+03	1.12E+05	5.42E+11
jDEsps	Best	1.95E-27	4.26E+01	4.97E-14	8.78E+08	2.05E+06	3.37E-01	6.98E+08	3.92E+12
	Median	2.88E-23	9.15E+01	1.10E-13	2.56E+09	3.11E+06	7.07E-01	1.14E+09	4.76E+12
	Worst	4.90E-22	7.65E+02	1.31E-12	2.72E+10	4.78E+06	2.24E+01	1.95E+09	2.27E+14
	Mean	9.17E-23	1.69E+02	1.73E-13	4.41E+09	3.14E+06	4.55E+00	1.24E+09	1.76E+13
	Stddev	1.37E-22	1.83E+02	2.55E-13	5.63E+09	6.38E+05	7.90E+00	4.06E+08	4.44E+13
MOS-CEC2013	Best	0.00E+00	7.06E+02	1.54E-12	4.77E+07	5.34E+06	6.16E+00	6.23E+01	5.49E+11
	Median	0.00E+00	8.24E+02	1.69E-12	7.80E+07	6.95E+06	1.39E+05	1.10E+03	2.82E+12
	Worst	0.00E+00	9.29E+02	1.95E-12	1.60E+08	9.40E+06	2.20E+05	5.30E+04	7.95E+12
	Mean	0.00E+00	8.23E+02	1.69E-12	8.73E+07	6.89E+06	1.43E+05	4.65E+03	2.85E+12
	Stddev	0.00E+00	4.69E+01	9.16E-14	3.11E+07	9.16E+05	6.86E+04	1.06E+04	1.44E+12
DECC-G	Best	8.47E-07	1.24E+03	1.05E-04	8.22E+10	5.73E+06	1.43E+05	4.55E+08	1.44E+15
	Median	2.33E-06	1.31E+03	1.19E+00	2.08E+11	8.19E+06	1.77E+05	9.10E+08	6.10E+15
	Worst	1.22E-05	1.38E+03	1.42E+00	4.12E+11	1.24E+07	2.28E+05	2.20E+09	1.63E+16
	Mean	3.22E-06	1.31E+03	1.09E+00	2.16E+11	8.30E+06	1.74E+05	1.02E+09	6.94E+15

Table C.1: Results of CEC 2010, CEC 2012, CEC 2013 and SOCO 2011 LSGO winners and proposed hybrid algorithms applied on CEC 2013 LSGO Benchmarks (F1, F2, F3, F4, F5, F6, F7, F8) (Continued)

	Stddev	2.83E-06	3.42E+01	3.54E-01	7.76E+10	1.60E+06	2.09E+04	4.89E+08	3.37E+15
CC-CMA-ES	Best	1.72E-09	1.15E+03	1.39E-13	9.86E+08	7.28E+14	1.18E+03	8.14E+04	6.83E+13
	Median	5.56E-09	1.34E+03	1.49E-13	1.88E+09	7.28E+14	2.06E+05	1.36E+06	2.92E+14
	Worst	7.52E-09	1.69E+03	1.63E-13	6.76E+09	7.28E+14	1.00E+06	1.04E+07	6.86E+14
	Mean	5.36E-09	1.37E+03	1.50E-13	2.82E+09	7.28E+14	4.56E+05	2.26E+06	3.32E+14
	Stddev	1.32E-09	1.38E+02	6.87E-15	1.84E+09	5.18E+06	4.56E+05	2.87E+06	1.74E+14

Table C.2: Results of CEC 2010, CEC 2012, CEC 2013 and SOCO 2011 LSGO winners and proposed hybrid algorithms applied on CEC 2013 LSGO Benchmarks (F9, F10, F11, F12, F13, F14, F15)

Algorithm	Metrics	F9	F10	F11	F12	F13	F14	F15
IMHS+MDE	Best	1.93E+08	91892953	4.8E+09	20302.52	3.45E+09	1.74E+10	13768197
	Median	2.52E+08	92948392	1.72E+10	33536.33	4.27E+09	4.5E+10	36003393
	Worst	3.04E+08	93626494	4.91E+10	83580.18	5.85E+09	6.02E+10	77715862
	Mean	2.49E+08	92966830	2.11E+10	39065.84	4.39E+09	4.1E+10	37876036
	Stddev	32949910	479080.6	1.26E+10	15600.81	5.78E+08	1.42E+10	17822924
MA-SW-Chains	Best	8.09E+07	8.63E+01	1.59E+08	1.08E+03	1.64E+07	1.11E+08	4.57E+06
	Median	1.16E+08	3.32E+02	2.10E+08	1.24E+03	1.91E+07	1.47E+08	5.76E+06
	Worst	1.54E+08	1.74E+05	2.42E+08	1.35E+03	2.55E+07	1.70E+08	1.16E+07
	Mean	1.14E+08	3.66E+04	2.10E+08	1.23E+03	1.98E+07	1.45E+08	5.90E+06
	Stddev	2.05E+07	6.17E+04	2.43E+07	8.32E+01	2.30E+06	1.69E+07	1.36E+06
2S-Ensemble	Best	1.05E+08	9.22E+06	8.26E+07	1.07E+03	6.51E+07	3.61E+07	1.87E+06
	Median	1.34E+08	1.44E+07	2.45E+08	1.66E+03	2.14E+08	5.68E+07	2.37E+06
	Worst	1.55E+08	2.20E+07	3.35E+08	2.02E+03	7.31E+08	9.73E+07	2.61E+06
	Mean	1.31E+08	1.43E+07	2.38E+08	1.70E+03	2.80E+08	6.09E+07	2.33E+06

Table C.2: Results of CEC 2010, CEC 2012, CEC 2013 and SOCO 2011 LSGO winners and proposed hybrid algorithms applied on CEC 2013 LSGO Benchmarks (F9, F10, F11, F12, F13, F14, F15) (Continued)

	Stddev	1.51E+07	2.87E+06	6.45E+07	2.20E+02	1.75E+08	1.75E+07	1.74E+05
MOS-SOCO2011	Best	5.02E+08	1.80E+07	3.83E+08	1.91E-02	2.97E+08	1.29E+08	3.40E+07
	Median	9.16E+08	8.78E+07	1.35E+09	7.01E+01	8.37E+08	2.30E+09	1.21E+08
	Worst	1.09E+09	8.97E+07	5.74E+11	2.97E+02	3.65E+09	6.55E+10	2.66E+08
	Mean	8.97E+08	6.65E+07	4.01E+10	8.63E+01	1.13E+09	6.89E+09	1.31E+08
jDElscoP	Stddev	1.39E+08	2.91E+07	1.23E+11	7.74E+01	7.74E+08	1.41E+10	6.02E+07
	Best	2.02E+08	5.61E+02	1.81E+10	9.70E+02	2.17E+10	2.53E+11	2.58E+07
	Median	3.67E+08	1.01E+03	1.22E+11	9.74E+02	3.48E+10	4.89E+11	3.43E+07
	Worst	4.46E+08	1.30E+03	2.15E+11	1.03E+03	4.86E+10	6.58E+11	7.28E+07
GaDE	Mean	3.66E+08	1.06E+03	1.15E+11	9.91E+02	3.49E+10	4.59E+11	3.98E+07
	Stddev	5.57E+07	1.72E+02	5.86E+10	2.70E+01	5.58E+09	1.16E+11	1.50E+07
	Best	4.84E+08	1.52E+07	1.32E+11	1.03E+03	2.12E+10	3.13E+11	7.76E+07
	Median	6.78E+08	2.29E+07	2.41E+11	1.11E+03	2.87E+10	4.19E+11	1.09E+08
MOS-CEC2012	Worst	7.74E+08	3.58E+07	3.51E+11	1.33E+03	3.54E+10	6.57E+11	1.33E+08
	Mean	6.62E+08	2.35E+07	2.40E+11	1.12E+03	2.88E+10	4.21E+11	1.07E+08
	Stddev	7.44E+07	4.59E+06	5.96E+10	6.90E+01	3.41E+09	7.13E+10	1.43E+07
	Best	1.00E+09	8.81E+07	1.85E+07	1.50E-02	4.70E+06	7.96E+06	4.89E+06
jDEsps	Median	1.65E+09	9.01E+07	2.53E+07	1.68E+02	5.90E+06	4.80E+07	7.25E+06
	Worst	2.33E+09	9.07E+07	3.89E+07	8.19E+02	6.84E+06	6.48E+07	1.04E+07
	Mean	1.65E+09	9.00E+07	2.71E+07	2.21E+02	5.84E+06	3.78E+07	7.63E+06
	Stddev	3.47E+08	5.04E+05	5.19E+06	2.29E+02	6.31E+05	2.01E+07	1.53E+06
jDEsps	Best	2.21E+08	8.71E+02	1.03E+10	3.34E+02	2.29E+10	7.54E+10	2.94E+07
	Median	2.74E+08	9.21E+02	2.74E+10	7.89E+02	3.12E+10	1.80E+11	4.59E+07
	Worst	3.21E+08	1.91E+03	5.20E+10	1.71E+03	3.82E+10	4.0BE+11	7.49E+07
	Mean	2.71E+08	1.05E+03	2.90E+10	8.51E+02	3.07E+10	1.93E+11	4.58E+07

Table C.2: Results of CEC 2010, CEC 2012, CEC 2013 and SOCO 2011 LSGO winners and proposed hybrid algorithms applied on CEC 2013 LSGO Benchmarks (F9, F10, F11, F12, F13, F14, F15) (Continued)

	Stddev	2.94E+07	2.43E+02	1.13E+10	3.64E+02	4.16E+09	8.24E+10	1.12E+07
MOS-CEC2013	Best	2.33E+08	4.44E+02	8.89E+06	4.40E-03	3.08E+05	7.83E+06	1.42E+06
	Median	4.18E+08	1.17E+06	1.71E+07	1.56E+01	1.02E+06	1.28E+07	1.68E+06
	Worst	5.01E+08	1.23E+06	2.85E+07	7.48E+02	2.22E+06	2.09E+07	2.03E+06
	Mean	3.99E+08	9.38E+05	1.73E+07	8.13E+01	1.00E+06	1.24E+07	1.71E+06
	Stddev	6.26E+07	4.79E+05	5.04E+06	1.57E+02	5.53E+05	2.86E+06	1.44E+05
DECC-G	Best	3.95E+08	7.87E+06	7.96E+09	3.59E+03	5.15E+09	6.77E+10	9.54E+06
	Median	5.32E+08	2.15E+07	1.25E+11	4.20E+03	8.66E+09	1.23E+11	1.19E+07
	Worst	7.82E+08	5.19E+07	2.32E+11	7.24E+03	1.75E+10	2.32E+11	1.42E+07
	Mean	5.47E+08	2.43E+07	1.21E+11	4.53E+03	9.40E+09	1.36E+11	1.17E+07
	Stddev	8.94E+07	9.75E+06	6.03E+10	9.48E+02	3.15E+09	4.79E+10	1.18E+06
CC-CMA-ES	Best	1.76E+08	1.32E+05	4.17E+07	9.78E+02	1.79E+07	2.27E+07	2.31E+07
	Median	3.57E+08	7.48E+05	9.79E+07	1.03E+03	4.12E+08	3.66E+07	3.05E+07
	Worst	6.77E+08	9.03E+07	5.04E+08	2.23E+03	1.90E+10	5.78E+09	4.29E+07
	Mean	3.82E+08	4.51E+06	1.24E+08	1.33E+03	1.80E+09	3.58E+08	3.13E+07
	Stddev	1.48E+08	1.79E+07	9.88E+07	4.78E+02	4.02E+09	1.15E+09	5.30E+06

Appendix D

Annexure (Fact Sheet of the Publications Online)

Table D.1: Fact sheet of Engineering Applications of Artificial Intelligence journal publication [168]

Title	Evolutionary computing applied to customer relationship management: A survey.
Authors	Gutha Jaya Krishna and Vadlamani Ravi
Publication	Engineering Applications of Artificial Intelligence, Vol. 56, pp. 30–59
ISSN	0952-1976
DOI	https://doi.org/10.1016/j.engappai.2016.08.012
Status	Published
Publisher	Elsevier
Publication Type	Journal
Year	2016
URL	http://www.sciencedirect.com/science/article/pii/S0952197616301439

Table D.2: Fact sheet of Applied Soft Computing journal publication [174]

Title	Key generation for plain text in stream cipher via bi-objective evolutionary computing
Authors	Gutha Jaya Krishna, Vadlamani Ravi and S. Nagesh Bhattu
Publication	Applied Soft Computing, Vol. 70, pp. 301 – 317
ISSN	1568-4946
DOI	https://doi.org/10.1016/j.asoc.2018.05.025
Status	Published
Publisher	Elsevier
Publication Type	Journal
Year	2018
URL	http://www.sciencedirect.com/science/article/pii/S1568494618302989

Table D.3: Fact sheet of Optimization in Industry book chapter publication [171]

Title	Evolutionary Computing Applied to Solve Some Operational Issues in Banks
Book Title	Optimization in Industry: Present Practices and Future Scopes
Authors	Gutha Jaya Krishna and Vadlamani Ravi
Editors	Shubhabrata Datta and J. Paulo Davim
Chapter	Ch. 3, pp. 31–53
Book Series	Part of the Management and Industrial Engineering book series (MINEN)
ISBN	978-3-030-01641-8 (Online); 978-3-030-01640-1 (Print)
ISSN	2365-0540 (Online); 2365-0532 (Print)
DOI	10.1007/978-3-030-01641-8_3
Status	Published
Publisher	Springer International Publishing
Publication Type	Book Chapter
Year	2019
URL	https://doi.org/10.1007/978-3-030-01641-8_3

Table D.4: Fact sheet of ICHSA-2015 publication [167]

Title	Modified Harmony Search Applied to Reliability Optimization of Complex Systems
Authors	Gutha Jaya Krishna and Vadlamani Ravi
Publication	Proceedings of the 2nd International Conference on Harmony Search Algorithm (ICHSA 2015), Seoul, South Korea, 19-21 August 2015, Advances in Intelligent Systems and Computing (AISC), Vol. 382, pp. 169–180
Editors	Joong Hoon Kim and Zong Woo Geem
ISBN	978-3-662-47926-1 (Online); 978-3-662-47925-4 (Print)
ISSN	2194-5365 (Online); 2194-5357 (Print)
DOI	10.1007/978-3-662-47926-1_17
Status	Published
Publisher	Springer Berlin Heidelberg
Publication Type	ICHSA 2015 Conference Proceedings, Harmony Search Algorithm, AISC, Vol. 382
Year	2015
URL	https://doi.org/10.1007/978-3-662-47926-1_17

Table D.5: Fact sheet of ICIA-2016 publication [169]

Title	Outlier Detection using Evolutionary Computing
Authors	Gutha Jaya Krishna and Vadlamani Ravi
Publication	Proceedings of the International Conference on Informatics and Analytics (ICIA 2016), Pondicherry, India, August 25 – 26, 2016, Article 17, pp. 17:1–17:6
ISBN	978-1-4503-4756-3
DOI	10.1145/2980258.2980295
Status	Published
Publisher	Association for Computing Machinery (ACM), New York, NY, USA
Publication Type	ICIA 2016 Conference Proceedings, ACM International Conference Proceeding Series (ICPS)
Year	2016
URL	http://doi.acm.org/10.1145/2980258.2980295

Table D.6: Fact sheet of ISDA-2018 publication [170]

Title	Hybrid Evolutionary Algorithm for Optimizing Reliability of Complex Systems
Authors	Gutha Jaya Krishna and Vadlamani Ravi
Publication	Proceedings of the 18th International Conference on Intelligent Systems Design and Applications (ISDA 2018), Vellore, India, December 6–8, 2018, Advances in Intelligent Systems and Computing (AISC), Vol. 941, pp. 108–119
Editors	Ajith Abraham, Aswani Kumar Cherukuri, Patricia Melin and Niketa Gandhi
ISBN	978-3-030-16660-1 (Online); 978-3-030-16659-5 (Print)
ISSN	2194-5365 (Online); 2194-5357 (Print)
DOI	10.1007/978-3-030-16660-1_11
Status	Published
Publisher	Springer International Publishing, Cham
Publication Type	ISDA 2018 Conference Proceedings, Advances in Intelligent Systems and Computing (AISC), Vol. 941
Year	2018
URL	https://doi.org/10.1007/978-3-030-16660-1_11

Table D.7: Fact sheet of ISDA-2018 publication [175]

Title	Differential Evolution Trained Fuzzy Cognitive Map: An Application to Modeling Efficiency in Banking
Authors	Gutha Jaya Krishna, Meesala Smruthi, Vadlamani Ravi and Bhamidipati Shandilya
Publication	Proceedings of the 18th International Conference on Intelligent Systems Design and Applications (ISDA 2018), Vellore, India, December 6–8, 2018, Advances in Intelligent Systems and Computing (AISC), Vol. 941, pp. 1–11
Editors	Ajith Abraham, Aswani Kumar Cherukuri, Patricia Melin and Niketa Gandhi
ISBN	978-3-030-16660-1 (Online); 978-3-030-16659-5 (Print)
ISSN	2194-5365 (Online); 2194-5357 (Print)
DOI	10.1007/978-3-030-16660-1_1
Status	Published
Publisher	Springer International Publishing, Cham
Publication Type	ISDA 2018 Conference Proceedings, Advances in Intelligent Systems and Computing (AISC), Vol. 941
Year	2018
URL	https://doi.org/10.1007/978-3-030-16660-1_1

Table D.8: Fact sheet of CoDS COMAD-2019 publication [172]

Title	Feature Subset Selection using Adaptive Differential Evolution: An Application to Banking
Authors	Gutha Jaya Krishna and Vadlamani Ravi
Publication	Proceedings of the ACM India Joint International Conference on Data Science and Management of Data (CoDS-COMAD 2019), Kolkata, India, January 03–05, 2019, pp. 157–163
ISBN	978-1-4503-6207-8
DOI	10.1145/3297001.3297021
Status	Published
Publisher	Association for Computing Machinery (ACM), New York, NY, USA
Publication Type	CoDS-COMAD 2019 Conference Proceedings, ACM International Conference Proceeding Series (ICPS)
Year	2019
URL	http://doi.acm.org/10.1145/3297001.3297021

Table D.9: Fact sheet of IEEE TENCON-2019 publication [173]	
Title	Keystroke based User Authentication using Modified Differential Evolution
Authors	Gutha Jaya Krishna and Vadlamani Ravi
Publication	Proceedings of the 2019 IEEE Region 10 Conference (IEEE TENCON 2019), Kochi, India, October 17–20, 2019, pp. 739–744
ISBN	978-1-7281-1895-6
ISSN	2159-3450
DOI	10.1109/TENCON.2019.8929371
Status	Published
Publisher	IEEE
Publication Type	TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON) Proceedings
Year	2019
URL	https://ieeexplore.ieee.org/document/8929371

The screenshot shows the article page for "Evolutionary computing applied to customer relationship management: A survey" in the journal "Engineering Applications of Artificial Intelligence". The page includes a sidebar with navigation links (Outline, Abstract, Keywords, Figures, Tables), the main article content (Title, Authors, DOI, Abstract), and a right sidebar with metrics (Citing articles, Article Metrics, Social Media). The abstract discusses the application of evolutionary computing techniques to CRM tasks, highlighting the trend of using such techniques for data mining and optimization.

Outline
 Abstract
 Keywords
 1. Introduction
 2. Review methodology
 3. Distribution of reviewed articles
 4. Review of works in the current study
 5. Discussion of the review and future directions
 6. Conclusions
 Appendix A
 References
 Show full outline

Figures (11)
 Show all figures

Tables (15)
 Table 1
 Table 2
 Table 3
 Table 4
 Table 5
 Table 6

Engineering Applications of Artificial Intelligence
 Volume 55, November 2016, Pages 30–59
ELSEVIER

Evolutionary computing applied to customer relationship management: A survey
 Gutha Jaya Krishna ^{a,*}, Vadlamani Ravi ^a
 Show more
<https://doi.org/10.1016/j.engappai.2016.08.012> Get rights and content

Abstract
 Customer relationship management (CRM) is a customer-centric business strategy which a company employs to improve customer experience and satisfaction by customizing products and services to customers' needs. This strategy, when implemented in totality eventually increases the revenue of the company. Traditionally, data mining (DM) techniques have been applied to solve various analytical CRM tasks. In turn, optimization techniques have long been used for training some of the DM techniques. However, during the past few years, evolutionary techniques have become so powerful and versatile that they can be deployed as a substitute for some DM techniques. This trend caught the attention of the researchers working in the analytical CRM area as they too started solving the CRM tasks using evolutionary techniques alone. In this context, we present a survey of evolutionary computing techniques applied to CRM tasks. In this paper, we surveyed 78 papers that were published during 1998 and 2015, where the application of evolutionary computing (EC) techniques to analytical CRM tasks is the main focus. The survey includes papers involving evolutionary computing techniques applied to the analytical CRM tasks under single- as well as multi-objective optimization framework. The purpose of the survey is to let the reader realize the versatility and power of EC techniques in solving analytical CRM tasks in the service industry and suggesting future directions.

Citing articles (19)
 Article Metrics
 Citations
 Citation indexes: 19
 Captures
 Exports-Saves: 30
 Readers: 148
 Social Media
 Shares, Likes & Comments: 5
 Tweets: 4
 View details

Figure D.1: Screenshot of Engineering Applications of Artificial Intelligence Publication [168]

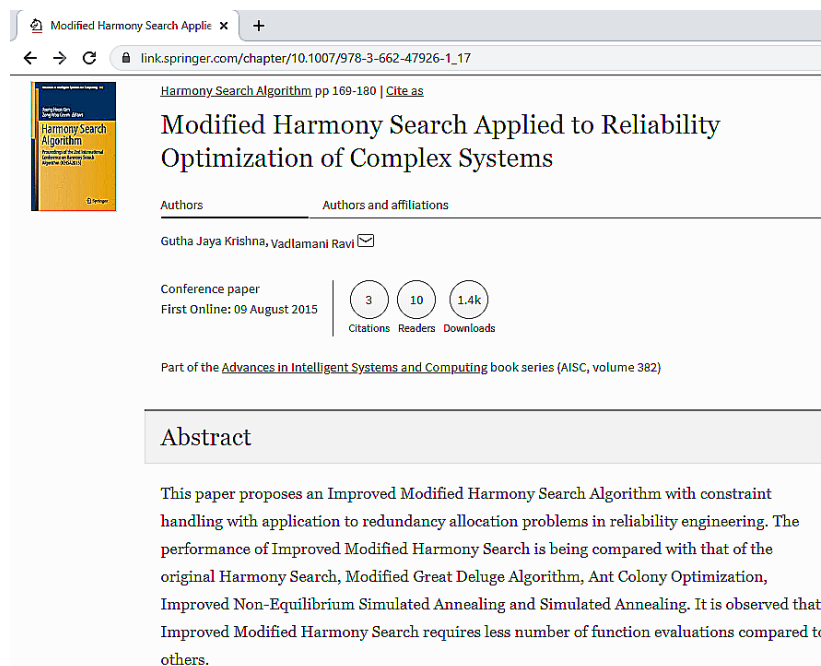


Figure D.4: Screenshot of ICHSA-2015 Publication [167]

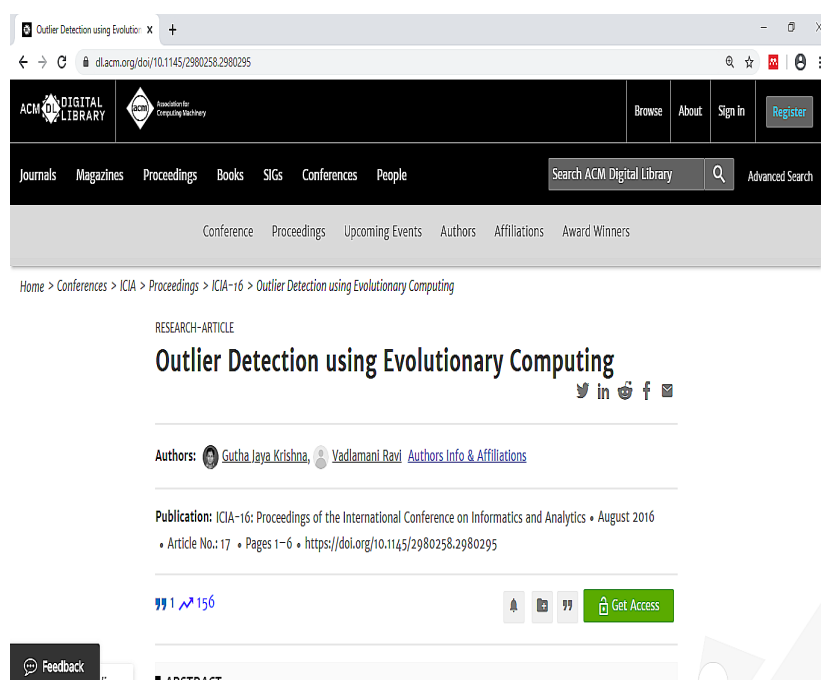


Figure D.5: Screenshot of ICIA-2016 Publication [169]

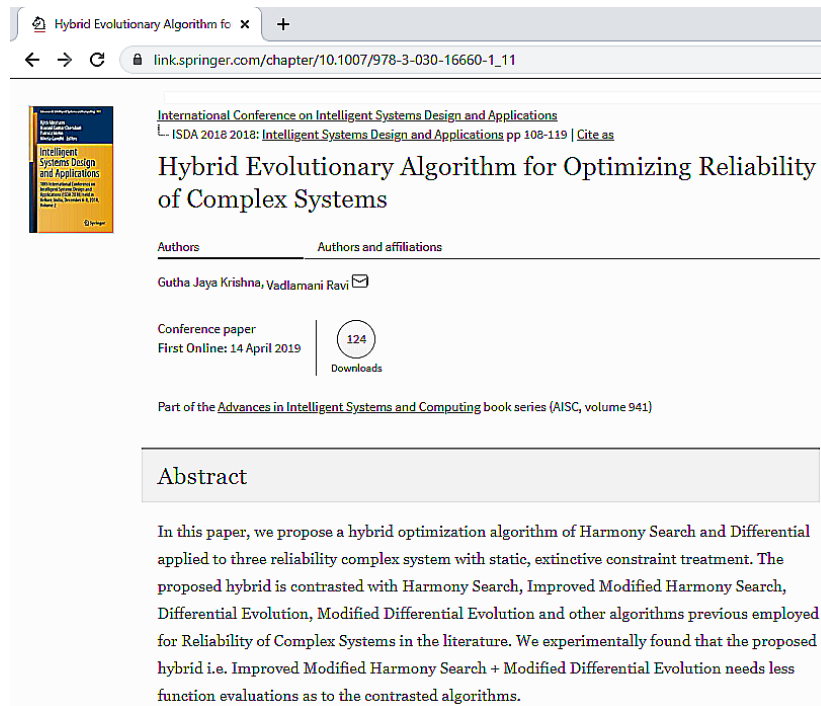


Figure D.6: Screenshot of ISDA-2018 Publication [170]

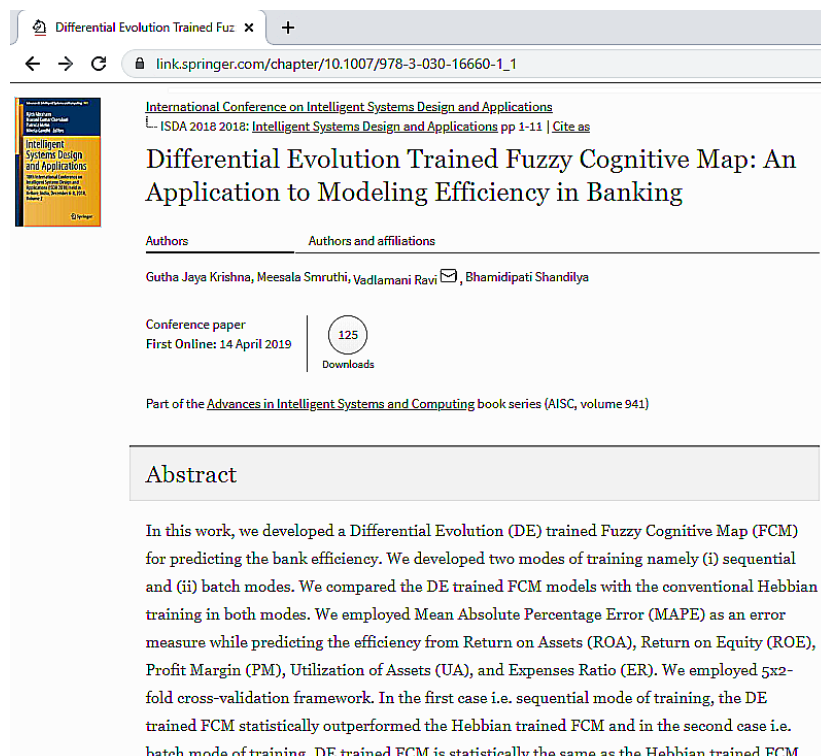


Figure D.7: Screenshot of ISDA-2018 Publication [175]

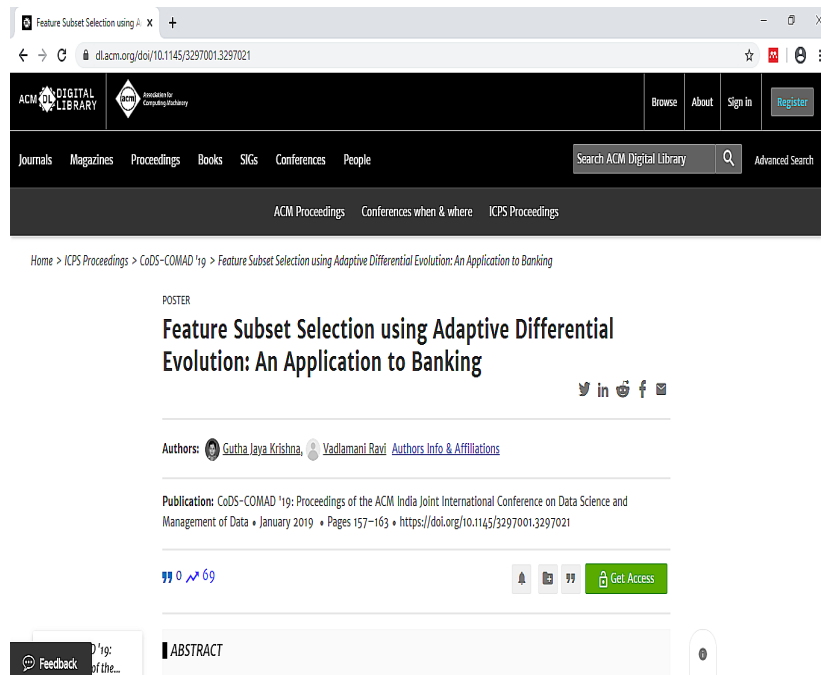


Figure D.8: Screenshot of CoDS COMAD-2019 publication [172]

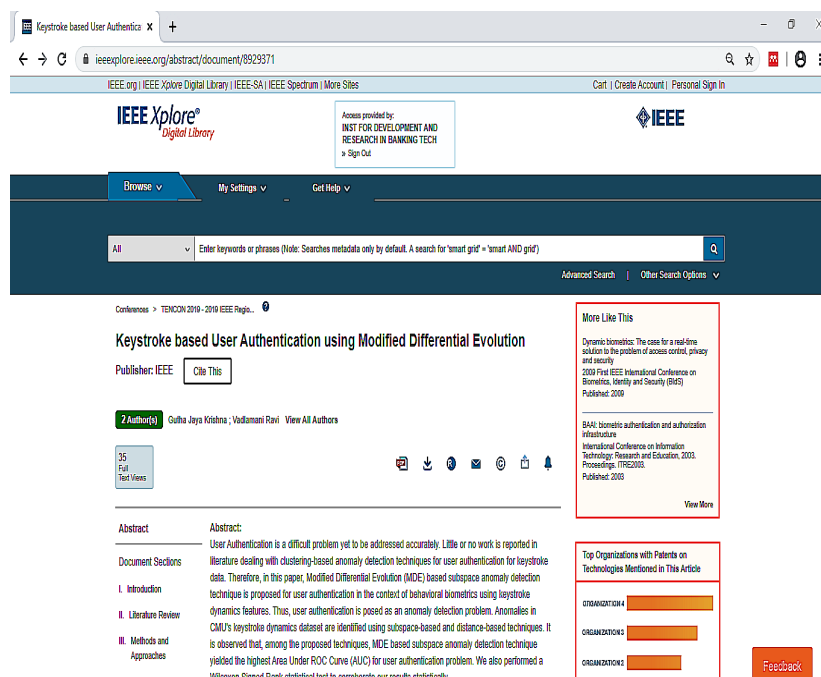


Figure D.9: Screenshot of IEEE TENCON 2019 publication [173]

Evolutionary Computing applied to solve some Analytical and Operational problems in Banking, Financial Services and Insurance

by Gutha Jaya Krishna

Submission date: 12-Mar-2020 12:12PM (UTC+0530)

Submission ID: 1274175572

File name: 14MCPC14_-Thesis_Without_Front_and_Back_Matter.pdf (3.73M)

Word count: 69810

Character count: 350227

Evolutionary Computing applied to solve some Analytical and Operational problems in Banking, Financial Services and Insurance

ORIGINALITY REPORT

39%

SIMILARITY INDEX

18%

INTERNET SOURCES

37%

PUBLICATIONS

13%

STUDENT PAPERS

PRIMARY SOURCES

1

Gutha Jaya Krishna, Vadlamani Ravi.
"Evolutionary computing applied to customer
relationship management: A survey",
Engineering Applications of Artificial Intelligence,
2016

Publication

9%

2

Gutha Jaya Krishna, Vadlamani Ravi, S.
Nagesh Bhattu. "Key generation for plain text in
stream cipher via bi-objective evolutionary
computing", Applied Soft Computing, 2018

Publication

6%

3

Gutha Jaya Krishna, Vadlamani Ravi. "Feature
Subset Selection using Adaptive Differential
Evolution", Proceedings of the ACM India Joint
International Conference on Data Science and
Management of Data - CoDS-COMAD '19, 2019

Publication

5%

4

link.springer.com

Internet Source

4%

5	www.dl-maghaleh.ir Internet Source	3%
6	Submitted to Assumption University Student Paper	2%
7	"Intelligent Systems Design and Applications", Springer Science and Business Media LLC, 2020 Publication	2%
8	Gutha Jaya Krishna, Vadlamani Ravi. "Keystroke based User Authentication using Modified Differential Evolution", TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON), 2019 Publication	2%
9	Submitted to Arab Open University Student Paper	1%
10	"Optimization in Industry", Springer Science and Business Media LLC, 2019 Publication	1%
11	Lecture Notes in Computer Science, 2015. Publication	1%
12	Gutha Jaya Krishna, Vadlamani Ravi. "Outlier Detection using Evolutionary Computing", Proceedings of the International Conference on Informatics and Analytics - ICIA-16, 2016 Publication	<1%

13

Communications in Computer and Information Science, 2015.

Publication

<1 %

14

Ameet Deshpande, Monisha Jegadeesan. "Leveraging Ontological Knowledge for Neural Language Models", Proceedings of the ACM India Joint International Conference on Data Science and Management of Data - CoDS-COMAD '19, 2019

Publication

<1 %

15

Ricardo P. Beausoleil. "Multi-start and path relinking methods to deal with multiobjective knapsack problems", Annals of Operations Research, 10/12/2007

Publication

<1 %

16

"High-Utility Pattern Mining", Springer Science and Business Media LLC, 2019

Publication

<1 %

17

Radka Poláková, Josef Tvrdík. "A COMBINED APPROACH TO ADAPTIVE DIFFERENTIAL EVOLUTION", Neural Network World, 2013

Publication

<1 %

18

"Handbook of Optimization", Springer Science and Business Media LLC, 2013

Publication

<1 %

19

www.lri.fr

Internet Source

<1 %

20	Maria Muntean, Corina Rotar, Ioan Ileana, Honoriu Valean. "Learning classification rules with genetic algorithm", 2010 8th International Conference on Communications, 2010 Publication	<1 %
21	arxiv.org Internet Source	<1 %
22	Ravisankar, P.. "Detection of financial statement fraud and feature selection using data mining techniques", Decision Support Systems, 201101 Publication	<1 %
23	Submitted to University of Strathclyde Student Paper	<1 %
24	public.cdl.uni-saarland.de Internet Source	<1 %
25	Gutha Jaya Krishna, Vadlamani Ravi. "Chapter 11 Hybrid Evolutionary Algorithm for Optimizing Reliability of Complex Systems", Springer Science and Business Media LLC, 2020 Publication	<1 %
26	hdl.handle.net Internet Source	<1 %
27	Hui Wang, Zhijian Wu, Shahryar Rahnamayan, Dazhi Jiang. "Sequential DE enhanced by neighborhood search for Large Scale Global	<1 %

Optimization", IEEE Congress on Evolutionary Computation, 2010

Publication

28

N. K. Sreelaja, G. A. Vijayalakshmi Pai. "Swarm intelligence based key generation for stream cipher", Security and Communication Networks, 2011

Publication

<1 %

29

Submitted to University of Hyderabad, Hyderabad

Student Paper

<1 %

30

Hooshmand, R.A.. "Emission, reserve and economic load dispatch problem with non-smooth and non-convex cost functions using the hybrid bacterial foraging-Nelder-Mead algorithm", Applied Energy, 201201

Publication

<1 %

31

Ali Wagdy Mohamed, Abdulaziz S. Almazyad. "Differential Evolution with Novel Mutation and Adaptive Crossover Strategies for Solving Large Scale Global Optimization Problems", Applied Computational Intelligence and Soft Computing, 2017

Publication

<1 %

32

Pattern Mining with Evolutionary Algorithms, 2016.

Publication

<1 %

33

Mirjam Sepesy Maučec, Janez Brest. "A review of the recent use of Differential Evolution for Large-Scale Global Optimization: An analysis of selected algorithms on the CEC 2013 LSGO benchmark suite", Swarm and Evolutionary Computation, 2019

Publication

<1 %

34

rsw.office.hiroshima-cu.ac.jp

Internet Source

<1 %

35

Zhao, Shi-Zheng, Ponnuthurai Nagaratnam Suganthan, and Swagatam Das. "Dynamic multi-swarm particle swarm optimizer with sub-regional harmony search", IEEE Congress on Evolutionary Computation, 2010.

Publication

<1 %

36

researchbank.rmit.edu.au

Internet Source

<1 %

37

Vy Huynh Trieu, Chau Truong Ngoc, Hai Le Quoc, Nguyen Nguyen Si. "Algorithm for Hiding High Utility Sensitive Association Rule Based on Intersection Lattice", 2018 1st International Conference on Multimedia Analysis and Pattern Recognition (MAPR), 2018

Publication

<1 %

38

worldwidescience.org

Internet Source

<1 %

39

www.tandfonline.com

Internet Source

<1 %

40

Wensheng Gan, Chun-Wei Lin, Philippe Fournier-Viger, Han-Chieh Chao, Vincent Tseng, Philip Yu. "A Survey of Utility-Oriented Pattern Mining", IEEE Transactions on Knowledge and Data Engineering, 2019

Publication

<1 %

41

Lecture Notes in Computer Science, 2013.

Publication

<1 %

42

Murat Aslan, Mesut Gunduz, Mustafa Servet Kiran. "JayaX: Jaya algorithm with xor operator for binary optimization", Applied Soft Computing, 2019

Publication

<1 %

43

Submitted to SASTRA University

Student Paper

<1 %

44

www.philippe-fournier-viger.com

Internet Source

<1 %

45

LaTorre, Antonio, Santiago Muelas, and Jose-Maria Pena. "Multiple Offspring Sampling in Large Scale Global Optimization", 2012 IEEE Congress on Evolutionary Computation, 2012.

Publication

<1 %

46

Gutha Jaya Krishna, Meesala Smruthi,

<1 %

Vadlamani Ravi, Bhamidipati Shandilya.
"Chapter 1 Differential Evolution Trained Fuzzy
Cognitive Map: An Application to Modeling
Efficiency in Banking", Springer Science and
Business Media LLC, 2020

Publication

47

Bahareh Rahmati, Mohammad Karim Sohrabi.
"A Systematic Survey on High Utility Itemset
Mining", International Journal of Information
Technology & Decision Making, 2019

Publication

<1 %

48

Yan Guo, Jing He, Libo Xu, Wei Liu. "A novel
multi-objective particle swarm optimization for
comprehensible credit scoring", Soft Computing,
2018

Publication

<1 %

49

simple.wikipedia.org

Internet Source

<1 %

50

cris.brighton.ac.uk

Internet Source

<1 %

51

Mohammad Nabi Omidvar, Xiaodong Li, Xin
Yao. "Cooperative Co-evolution with delta
grouping for large scale non-separable function
optimization", IEEE Congress on Evolutionary
Computation, 2010

Publication

<1 %

Issa Traore, Isaac Woungang, Mohammad S.

52

Obaidat, Youssef Nakkabi, Iris Lai. "Combining Mouse and Keystroke Dynamics Biometrics for Risk-Based Authentication in Web Environments", 2012 Fourth International Conference on Digital Home, 2012

Publication

<1 %

53

"Advances in Swarm Intelligence", Springer Science and Business Media LLC, 2013

Publication

<1 %

54

Huang, J.J.. "Two-stage genetic programming (2SGP) for the credit scoring model", Applied Mathematics and Computation, 20060315

Publication

<1 %

55

Submitted to National Institute of Technology, Rourkela

Student Paper

<1 %

56

Alberto Cano, Carlos García-Martínez, Sebastián Ventura. "Extremely high-dimensional optimization with MapReduce: Scaling functions and algorithm", Information Sciences, 2017

Publication

<1 %

57

"Nature-Inspired Computation and Machine Learning", Springer Science and Business Media LLC, 2014

Publication

<1 %

58

Jimmy Ming-Tai Wu, Justin Zhan, Jerry Chun-Wei Lin. "An ACO-based approach to mine high-

<1 %

utility itemsets", Knowledge-Based Systems, 2017

Publication

59

Lecture Notes in Computer Science, 2011.

Publication

<1 %

60

www.mdpi.com

Internet Source

<1 %

61

www.orb-academic.org

Internet Source

<1 %

62

Submitted to SUNY, Binghamton

Student Paper

<1 %

63

LaTorre, Antonio, Santiago Muelas, and José-María Peña. "A comprehensive comparison of large scale global optimizers", Information Sciences, 2015.

Publication

<1 %

64

Wang, Y.J.. "A dynamic clustering based differential evolution algorithm for global optimization", European Journal of Operational Research, 20071116

Publication

<1 %

65

"Cutting-Edge Research Topics on Multiple Criteria Decision Making", Springer Science and Business Media LLC, 2009

Publication

<1 %

Thang Mai, Bay Vo, Loan T.T. Nguyen. "A

66	lattice-based approach for mining high utility association rules", Information Sciences, 2017 Publication	<1 %
67	vdocuments.mx Internet Source	<1 %
68	"Proceedings of the International Conference on Soft Computing for Problem Solving (SocProS 2011) December 20-22, 2011", Springer Science and Business Media LLC, 2012 Publication	<1 %
69	Submitted to Indian Institute of Technology, Kharagpure Student Paper	<1 %
70	"Guide to Wireless Network Security", Springer Science and Business Media LLC, 2006 Publication	<1 %
71	Lecture Notes in Computer Science, 2012. Publication	<1 %
72	Chin-Tsai Lin, In-Fun Lee. "Building decision analysis model by the experts knowledge—applying to management fraud detection", Journal of Information and Optimization Sciences, 2010 Publication	<1 %
73	Yadav, P.. "An Intelligent Tuned Harmony Search algorithm for optimisation", Information	<1 %

74

Ke-Lin Du, M. N. S. Swamy. "Search and Optimization by Metaheuristics", Springer Science and Business Media LLC, 2016

Publication

<1 %

75

Xiaoling Jin, Jianguo Zhao, Ying Sun, Kejun Li, Boqin Zhang. "Distribution network reconfiguration for load balancing using binary particle swarm optimization", 2004 International Conference on Power System Technology, 2004. PowerCon 2004., 2004

Publication

<1 %

76

"Evolutionary Computation in Combinatorial Optimization", Springer Science and Business Media LLC, 2013

Publication

<1 %

77

Ganghishetti Pradeep, Vadlamani Ravi. "chapter 7 Fuzzy Multi-Objective Association Rule Mining Using Evolutionary Computation", IGI Global, 2017

Publication

<1 %

78

Israel Cruz-Vega, Hugo Jair Escalante, Carlos A. Reyes, Jesus A. Gonzalez, Alejandro Rosales. "Surrogate modeling based on an adaptive network and granular computing", Soft Computing, 2015

<1 %

79

Janez Brest, Borko Boskovic, Ales Zamuda, Iztok Fister, Mirjam Sepesy Maucec. "Self-adaptive differential evolution algorithm with a small and varying population size", 2012 IEEE Congress on Evolutionary Computation, 2012

Publication

<1 %

80

Submitted to University of Brighton

Student Paper

<1 %

81

courses.cs.ut.ee

Internet Source

<1 %

82

Josef Tvrdik. "Adaptive differential evolution and exponential crossover", 2008 International Multiconference on Computer Science and Information Technology, 2008

Publication

<1 %

83

Xingguang Peng, Yapei Wu. "Large-scale cooperative co-evolution using niching-based multi-modal optimization and adaptive fast clustering", Swarm and Evolutionary Computation, 2017

Publication

<1 %

84

B. S. N. Murty. "Reliability optimization of complex systems using modified random-to-pattern search (MRPS) algorithm", Quality and Reliability Engineering International, 05/1999

Publication

<1 %

85	Submitted to National Institute of Industrial Engineering Student Paper	<1 %
86	Submitted to KTH - The Royal Institute of Technology Student Paper	<1 %
87	Keurfon Luu, Mark Noble, Alexandrine Gesret, Philippe Thierry. "Towards large-scale stochastic refraction tomography: a comparison of three evolutionary algorithms", Geophysical Prospecting, 2019 Publication	<1 %
88	61.187.92.238:5008 Internet Source	<1 %
89	Proceedings in Adaptation Learning and Optimization, 2015. Publication	<1 %
90	espace.library.uq.edu.au Internet Source	<1 %
91	VADLAMANI RAVI. "OPTIMIZATION OF COMPLEX SYSTEM RELIABILITY BY A MODIFIED GREAT DELUGE ALGORITHM", Asia-Pacific Journal of Operational Research, 2012 Publication	<1 %

92

Internet Source

<1 %

93

pareto.uab.es

Internet Source

<1 %

94

Submitted to Universiti Teknikal Malaysia
Melaka

Student Paper

<1 %

95

"Advances in Swarm Intelligence", Springer
Science and Business Media LLC, 2016

Publication

<1 %

96

Kaibo Zhang, Bin Li. "Cooperative Coevolution
with global search for large scale global
optimization", 2012 IEEE Congress on
Evolutionary Computation, 2012

Publication

<1 %

97

Wang, Yu, Jin Huang, Wei Shan Dong, Jun Chi
Yan, Chun Hua Tian, Min Li, and Wen Ting Mo.
"Two-stage based ensemble optimization
framework for large-scale global optimization",
European Journal of Operational Research,
2013.

Publication

<1 %

Exclude quotes On

Exclude bibliography On

Exclude matches

< 14 words

Evolutionary Computing applied to solve some Analytical and Operational problems in Banking, Financial Services and Insurance

by Gutha Jaya Krishna

Submission date: 12-Mar-2020 12:12PM (UTC+0530)

Submission ID: 1274175572

File name: 14MCPC14_-Thesis_Without_Front_and_Back_Matter.pdf (3.73M)

Word count: 69810

Character count: 350227

Similarity Screening Done @ IGM Library

Librarian: DLYAL
IGM Library (UOH)

Evolutionary Computing applied to solve some Analytical and Operational problems in Banking, Financial Services and Insurance

ORIGINALITY REPORT

Similarity Screening Done @ IGM Library
SUB ID: 1274175372
Sub date: 12.02.2020
Librarian / DL / AL
IGM Library UOH

39%

SIMILARITY INDEX

18%

INTERNET SOURCES

37%

PUBLICATIONS

13%

STUDENT PAPERS

PRIMARY SOURCES

Note: Samples 1 to 10, 12, 25 & 46 belong to student works VR

1

Gutha Jaya Krishna, Vadlamani Ravi.

"Evolutionary computing applied to customer relationship management: A survey",
Engineering Applications of Artificial Intelligence,
2016

Publication

9%

2

Gutha Jaya Krishna, Vadlamani Ravi, S.
Nagesh Bhattu. "Key generation for plain text in
stream cipher via bi-objective evolutionary
computing", Applied Soft Computing, 2018

Publication

6%

3

Gutha Jaya Krishna, Vadlamani Ravi. "Feature
Subset Selection using Adaptive Differential
Evolution", Proceedings of the ACM India Joint
International Conference on Data Science and
Management of Data - CoDS-COMAD '19, 2019

Publication

5%

4

link.springer.com

Internet Source

my other samples pertain to
same source, pls exclude
VR

Dr. A.S. RAMASASTRI
DIRECTOR

INSTITUTE FOR DEVELOPMENT AND
RESEARCH IN BANKING TECHNOLOGY
(Established by Reserve Bank of India)
Castle Hills Road No.1, Masab Tank,

Dr. V. Ravi, M.Sc., Ph.D.,
Professor

Head, Center of Excellence on CRM &
Analytics
INSTITUTE FOR DEVELOPMENT AND
RESEARCH IN BANKING TECHNOLOGY
(Established by Reserve Bank of India)
Castle Hills Road No.1, Masab Tank,

4%

5 www.dl-maghaleh.ir
Internet Source

6 Submitted to Assumption University
Student Paper

7 "Intelligent Systems Design and Applications",
Springer Science and Business Media LLC,
2020
Publication

8 Gutha Jaya Krishna, Vadlamani Ravi.
"Keystroke based User Authentication using
Modified Differential Evolution", TENCON 2019 -
2019 IEEE Region 10 Conference (TENCON),
2019
Publication

9 Submitted to Arab Open University
Student Paper

10 "Optimization in Industry", Springer Science and
Business Media LLC, 2019
Publication

11 Lecture Notes in Computer Science, 2015.
Publication

12 Gutha Jaya Krishna, Vadlamani Ravi. "Outlier
Detection using Evolutionary Computing",
Proceedings of the International Conference on
Informatics and Analytics - ICIA-16, 2016
Publication

13

Communications in Computer and Information Science, 2015.

Publication

<1 %

14

Ameet Deshpande, Monisha Jegadeesan. "Leveraging Ontological Knowledge for Neural Language Models", Proceedings of the ACM India Joint International Conference on Data Science and Management of Data - CoDS-COMAD '19, 2019

Publication

<1 %

15

Ricardo P. Beausoleil. "Multi-start and path relinking methods to deal with multiobjective knapsack problems", Annals of Operations Research, 10/12/2007

Publication

<1 %

16

"High-Utility Pattern Mining", Springer Science and Business Media LLC, 2019

Publication

<1 %

17

Radka Poláková, Josef Tvrdík. "A COMBINED APPROACH TO ADAPTIVE DIFFERENTIAL EVOLUTION", Neural Network World, 2013

Publication

<1 %

18

"Handbook of Optimization", Springer Science and Business Media LLC, 2013

Publication

<1 %

19

www.lri.fr

Internet Source

<1 %

20 Maria Muntean, Corina Rotar, Ioan Ileana, Honoriu Valean. "Learning classification rules with genetic algorithm", 2010 8th International Conference on Communications, 2010
Publication

<1%

28

21 arxiv.org
Internet Source

<1%

29

22 Ravisankar, P.. "Detection of financial statement fraud and feature selection using data mining techniques", Decision Support Systems, 201101
Publication

<1%

30

23 Submitted to University of Strathclyde
Student Paper

<1%

24 public.cdl.uni-saarland.de
Internet Source

<1%

31

25 Gutha Jaya Krishna, Vadlamani Ravi. "Chapter 11 Hybrid Evolutionary Algorithm for Optimizing Reliability of Complex Systems", Springer Science and Business Media LLC, 2020
Publication

<1%

26 hdl.handle.net
Internet Source

<1%

32

27 Hui Wang, Zhijian Wu, Shahryar Rahnamayan, Dazhi Jiang. "Sequential DE enhanced by neighborhood search for Large Scale Global

<1%

39

www.tandfonline.com

Internet Source

<1 %

40

Wensheng Gan, Chun-Wei Lin, Philippe Fournier-Viger, Han-Chieh Chao, Vincent Tseng, Philip Yu. "A Survey of Utility-Oriented Pattern Mining", IEEE Transactions on Knowledge and Data Engineering, 2019

Publication

<1 %

41

Lecture Notes in Computer Science, 2013.

Publication

<1 %

42

Murat Aslan, Mesut Gunduz, Mustafa Servet Kiran. "JayaX: Jaya algorithm with xor operator for binary optimization", Applied Soft Computing, 2019

Publication

<1 %

43

Submitted to SASTRA University

Student Paper

<1 %

44

www.philippe-fournier-viger.com

Internet Source

<1 %

45

LaTorre, Antonio, Santiago Muelas, and Jose-Maria Pena. "Multiple Offspring Sampling in Large Scale Global Optimization", 2012 IEEE Congress on Evolutionary Computation, 2012.

Publication

<1 %

46

Gutha Jaya Krishna, Meesala Smruthi,

<1 %

Vadlamani Ravi, Bhamidipati Shandilya.
"Chapter 1 Differential Evolution Trained Fuzzy
Cognitive Map: An Application to Modeling
Efficiency in Banking", Springer Science and
Business Media LLC, 2020

Publication

- 47 Bahareh Rahmati, Mohammad Karim Sohrabi.
"A Systematic Survey on High Utility Itemset
Mining", International Journal of Information
Technology & Decision Making, 2019

Publication

- 48 Yan Guo, Jing He, Libo Xu, Wei Liu. "A novel
multi-objective particle swarm optimization for
comprehensible credit scoring", Soft Computing,
2018

Publication

- 49 simple.wikipedia.org
Internet Source

- 50 cris.brighton.ac.uk
Internet Source

- 51 Mohammad Nabi Omidvar, Xiaodong Li, Xin
Yao. "Cooperative Co-evolution with delta
grouping for large scale non-separable function
optimization", IEEE Congress on Evolutionary
Computation, 2010

Publication

Issa Traore, Isaac Woungang, Mohammad S.