

**Dissertation for the Degree of Doctor of Philosophy**

**ACCURATE CLASSIFIER SELECTION  
METHODOLOGY USING MULTI-  
CRITERIA DECISION MAKING AND  
META-LEARNING**

**Rahman Ali**

**Department of Computer Science and Engineering  
Graduate School  
Kyung Hee University  
South Korea  
August 2016**

**Dissertation for the Degree of Doctor of Philosophy**

**ACCURATE CLASSIFIER SELECTION  
METHODOLOGY USING MULTI-  
CRITERIA DECISION MAKING AND  
META-LEARNING**

**Rahman Ali**

**Department of Computer Science and Engineering  
Graduate School  
Kyung Hee University  
South Korea  
August 2016**

This thesis is dedicated to my beloved mother (may ALLAH rest her soul in peace), father, brothers and sisters and to my beloved wife and sweet sons for their endless love, support and encouragement.

# ACCURATE CLASSIFIER SELECTION METHODOLOGY USING MULTI-CRITERIA DECISION MAKING AND META-LEARNING

by

Rahman Ali

Supervised by

Prof. Sungyoung Lee

Submitted to the Department of Computer Science and Engineering and the Faculty of Graduate  
School of Kyung Hee University in partial fulfillment of the requirements of the degree of  
Doctor of Philosophy

Dissertation Committee:

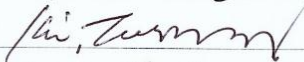
Prof. Oksam Chae



Prof. TaeChoong Chung



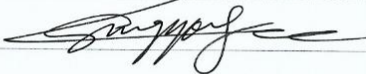
Prof. Tae-Seong Kim



Prof. Il-Kon Kim



Prof. Sungyoung Lee



One of the important tasks in data mining applications is to find suitable classifier(s), for user's classification problems, and designing the classifier accurately to meet their application's requirements. The design of an accurate methodology for evaluating the performance of these algorithms and selecting the best one has recently gained an immense interest of the research community due to the rapid shift of data mining processes and the use of classification algorithms from academics to the real-world application domains. If these tasks are not carefully accomplished, the evaluation of algorithms performance and consequently the selection of a best classifier may result in invalid recommendations of a statistically incorrect classifier(s). Subsequently, incorrect decisions will be made by the applications, which are based on these recommended classifiers. In practical data mining application scenarios, this is a subjective decision making process that not only takes experts' preferences and interests into account but also considers a number of other factors into account, such as data characteristic (e.g., meta-features), classifiers characteristics (e.g., performance metrics) and domain specific data mining processes and their associated domain constraints. For example, some domains require interpretable classification model, while other requires classifiers with reasonable training and/or testing time, or have the capacity to classify binary class problem or multi-class problem or have consistent performance results. These obligations make the processes of classifiers evaluation, selection, and design more challenging, especially in situations where the evaluation and selection are based on multiple characteristics of the classifier (i.e., performance metrics, called criteria), data characteristics (i.e., meta-features) and the associated constraints, all taken into account simultaneously. This thesis is focused on multi-criteria evaluation of classifiers, meta-learning based decision tree classifier selection and design of some accurate classifiers for real-world applications scenarios. The design of accurate rough-set and hybrid case-based reasoning (hybrid-CBR) classifiers are discussed along with their associated issues, such as domain-

specific data acquisition for real-world dataset and case-base creation, semantics-preserving discretization and accurate and efficient case matching and retrieval functions for case-based reasoning. In case of classifiers performance evaluation, there is no universally acceptable classifier that outperforms all other classifiers on every kind of domain data, given a single evaluation criterion or multi-metrics evaluation criteria. Similarly, there is no universally acceptable guidelines or rules for the selection of suitable evaluation metric(s) to evaluate the classifiers. Other related issues regarding classifier evaluation include: the experts' preferences (*i.e.*, weights on the criteria) are normally defined using absolute values that lack the consistency check for insuring that the assigned weight are correct, global and local constraints of the domain and evaluation metrics which sometimes impose restrictions on the classifiers performance evaluation process and must need to be satisfied. Moreover, there is lack of a universally acceptable classifier evaluation strategy, which includes almost all the required multiple-criteria including consistency measure to insure the selection of optimum performance consistent classifier. Apart from the issues highlighted in state-of-the-art classifiers performance evaluation methods, the automatic classifiers selection using meta-learning also suffers from a number of challenging issues. These include: the extraction and selection of a suitable set of meta-characteristics of the data to best represent the intrinsic behaviors of the dataset from all aspects and thus help in automatic recommendation of best classifier and enabling multi-views multi-level meta-learning and reasoning for accurately selecting classifiers based on data and classifiers characteristics.

This thesis establishes the problem statement and proposes a number of theoretical and systematic empirical methods and meta-learning based methods to provide solutions to the problem of accurate classifier selection and the associated issues, mentioned above. Similarly, for the issues highlighted in real-world application scenarios, novel methods are proposed to improve performance of the traditional rough-set and case-based reasoning classifiers.

The problem of best classifier selection and design can be approached either using automatic evaluation, ranking and selection methods or using the expert's heuristic knowledge about the domain problem and the candidate classifiers. Under the automatic classifier selection approach, two types of novel methodologies are proposed. In the first methodology, a unique accurate multi-criteria decision making (AMD) method is proposed that evaluates the classifiers performance on the basis of multiple performance metrics (constituting a composite criterion) satisfying the domain constraints and ranks the final score to select the top-ranked classifier as the best one. In this method, based on the motivation from experts' consensus-based nominal group technique (NGT), an experts' group-based decision making method is proposed that accurately selects suitable performance metrics satisfying the domain constraints. The experts' preferences on the evaluation metrics are realized and quantified using the experts' group decision making with relative consistent weighting scheme using analytical hierarchy process (AHP). For ranking performance of the classification algorithms, relative closeness values, with respect to the ideal classifier, are computed for all the classifiers using multi-criteria decision making Technique for Order Performance by Similarity to Ideal Solution (TOPSIS). Moreover, this thesis contributes in the selection of a significant performance consistent classifier by introducing an additional consistency measure in the evaluation criteria and using only statistically significant classifiers in the evaluation process. The statistical significance test is enhanced by encompassing a fitness evaluation function that excludes the algorithms that perform significantly poor on all the considered evaluation criteria. In the second methodology of classifier selection, a novel CBR-based meta-learning and reasoning (CBR-MLR) framework is proposed and implemented that utilizes data and classifiers meta-characteristics during multi-level multi-views case-based reasoning to accurately recommend best decision tree classifier for users' applications in-hand. In this method, 29 meta-characteristics are extracted from user data and 09 decision tree classifiers are empirically evaluated, using predictive accuracy and consistency, to design a Case-Base. Accurate case retrieval functions are defined and the CBR output is refined

with classifiers conflict resolution approach that uses weight sum score and AMD methods.

The heuristic-based evaluation and selection method is based on the experts' knowledge about the candidate classifiers' performance on a particular application. Under this approach, semantics-preserving accurate rough-set classifier, based on rough-set theory (RST), and precise hybrid-CBR classifiers, are proposed, designed and implemented in real-world application scenarios. In the design of these classifiers, standard data mining process flow is used with necessary modifications in order to fulfill the specific requirements of the domain applications. However, the methodologies are designed in generalized manner, without restricting to the specific domains for which they have been initially designed. For improving capability of the rough-set classifier, a new, semantics-preserving discretization scheme is introduced that keeps the data semantics intact after being transformed into decision rules. Similarly, the design of the standard CBR classifier is improved by efficiently integrating it with rule-based reasoning and defining accurate case similarity and retrieval function.



## Acknowledgement

---

First, I render my humble and sincere thanks to the *Almighty ALLAH* for blessings upon me. The *Almighty ALLAH* gave me the strength, courage, and patience during my doctoral studies. Special thanks to my advisor Prof. Sungyoung Lee who provided me guidance, strength, support, and courage in overcoming the difficult challenges throughout my PhD time. I learned a lot from him in becoming a productive person in diverse situations. He inspired me with his dynamic personality and his unreserved help and his guidance lead me to successfully finish my PhD studies. He has great role in polishing my skills, such as thinking, creativity and technical soundness, which are the key ingredients of a quality research. I am grateful to my dissertation evaluation committee for their comments and valuable suggestions they provided me during the defense of my thesis. Their valuable comments improved the presentation and contents of this dissertation.

I am extremely grateful to my colleagues who have always provided me time, expertise, and encouragement in my course of research. They always guide me in difficult situations of my PhD studies. I would like to thank Dr. Asad Masood Khattak, Dr. Zeeshan Pervaiz, Dr. Muhammad Fahim, Dr. Wajahat Ali Khan, Dr. Bilal Amin, Dr. Muhammad Hameed Siddiqi, Maqbool Hussain, Shujaat Hussain, and Muhammad Afzal. They have contributed enormously in successfully performing various academic and personal tasks that confronted me during my stay at the Republic of Korea. I am very thankful to all of my current and former Ubiquitous computing lab fellows and colleagues for their kind support to my personal and academic life at Kyung Hee University. I am highly obliged to brilliant researchers Mahmood Ahmad, Taqdir Ali, Jamil Hussain, Maqbool Ali, Bilal Ali, Muhammad Asif Razzaq, Imran Ali, Usman Akhtar, Muhammad Idris, Tae Ho Hur, Jae Hun Bang, Kifayat Ullah, and Saeed Ullah. This journey would not have been possible without their support. They contributed a lot in developing my personal and academic life. Furthermore, I appreciate the efforts of my roommates Dr. Rizwan

Ahmad, Dr. Muhammad Faheem and specially Mr. Amjad Ali who always refreshed my mind with delicious food during my busy days of research and thesis preparation. I am also thankful to all my Korean and international friends who worked with me as a team and developed my team work skills and provided me wonderful memories during my stay in Republic of Korea.

Last but not the least, I would like to express my sincere gratitude to my mother (may ALLAH rest her soul in peace), father, brothers, sisters, my beloved wife and in-laws for their endless love, support, prayers, and encouragement. Their support and encouragement has made this dissertation possible. They provided me the strength to work hard during my PhD studies. I would like to thank my cousins for their support and guidance in every difficult situation I faced during this hard time.

Rahman Ali  
August, 2016

## Table of Contents

ABSTRACT .....	I
ACKNOWLEDGEMENT .....	V
TABLE OF CONTENTS .....	VII
TABLE OF TABLES .....	XII
TABLE OF FIGURES .....	XV
CHAPTER 1 .....	1
INTRODUCTION.....	1
1.1.    BACKGROUND .....	1
1.2.    MOTIVATION .....	3
1.3.    PROBLEM STATEMENT .....	5
1.4.    PROPOSED CONCEPT .....	7
1.5.    CONTRIBUTIONS .....	8
1.6.    THESIS ORGANIZATION.....	12
CHAPTER 2 .....	15
RELATED WORK.....	15
2.1.    OVERVIEW.....	15
2.2.    AUTOMATIC CLASSIFIER SELECTION .....	16
2.2.1. <i>Multi-criteria decision making for classifiers ranking and selection</i> .....	16
2.2.2. <i>Meta-learning and reasoning for classifier selection</i> .....	19
2.3.    HEURISTICS-BASED CLASSIFIER SELECTION AND DESIGN FOR REAL-WORLD APPLICATIONS.....	20
2.3.1. <i>Rough Set Classifier Selection and Design for Real-world Application</i> .....	21
2.3.2. <i>Hybrid-CBR Classifier Selection and Design for Wellness</i> .....	23
2.3.3. <i>Trade-off criteria for evaluating heuristic approach for classifiers selection</i> 25	
2.4.    SUMMARY .....	26
CHAPTER 3 .....	27
MACHINE LEARNING AND CLASSIFICATION: TECHNICAL PRELIMINARIES .....	27
3.1.    OVERVIEW.....	27
3.2.    DATA MINING.....	27
3.2.1. <i>Technologies used in data mining</i> .....	27
3.3.    MACHINE LEARNING .....	27
3.3.1. <i>Supervised learning</i> .....	28
3.3.2. <i>Unsupervised learning</i> .....	28

3.4.	CLASSIFICATION .....	28
3.4.1.	<i>Classification techniques</i> .....	28
3.4.1.1.	Decision tree induction .....	28
3.4.1.2.	Bayes classification methods .....	28
3.4.1.3.	Rule-based classification .....	28
3.4.1.4.	Meta-learning or classifiers ensemble methods .....	29
3.4.1.5.	Case-based reasoning for classification .....	29
3.4.1.6.	Rough sets classification .....	29
3.4.2.	<i>Evaluation and selection of classifiers</i> .....	30
3.4.2.1.	Metrics for evaluating classifier performance .....	30
3.4.2.2.	Cross-validation .....	31
3.5.	BINARY AND MULTICLASS CLASSIFICATION .....	31
3.6.	DECISION MAKING PROCESS .....	31
3.6.1.	<i>Multi-criteria decision making</i> .....	32
3.6.1.1.	Analytic hierarchy process .....	32
3.6.1.2.	Technique for Order Preference by Similarity to Ideal Solution .....	32
3.7.	META-LEARNING FOR ALGORITHMS SELECTION .....	32
3.7.1.	<i>Meta-features of datasets and algorithms</i> .....	32
3.7.2.	<i>Meta-learner for algorithms selection</i> .....	33
3.7.3.	<i>Meta-reasoner for algorithms selection</i> .....	33
3.8.	SUMMARY .....	33
CHAPTER 4	.....	34
MULTI-CRITERIA DECISION MAKING FOR CLASSIFIER SELECTION	.....	34
4.1.	OVERVIEW .....	34
4.1.1.	<i>Key Contributions</i> .....	35
4.2.	ALGORITHM SELECTION: MULTI-METRIC DECISION MAKING PROCESS .....	36
4.3.	METHODOLOGY – MULTICRITERIA EVALUATION OF CLASSIFIERS .....	37
4.3.1.	<i>Guidelines for algorithms evaluation</i> .....	38
4.3.2.	<i>Multi-metric decision making for algorithm selection</i> .....	39
4.3.2.1.	Selecting Suitable Quality Meta-metrics .....	42
a.	Classifiers quality meta-metrics classification model .....	42
b.	Selecting suitable quality meta-metrics .....	46
4.3.2.2.	Selecting suitable evaluation-metrics .....	48
4.3.2.3.	Consistent relative criteria weighting .....	49
4.3.2.4.	Measuring algorithms performance .....	53
4.3.2.5.	Testing significance of performance results .....	54
4.3.2.6.	Algorithmic fitness evaluation .....	55
4.3.2.7.	Ranking algorithms .....	56
4.3.2.8.	Constraints satisfaction .....	57
4.4.	VALIDATION OF THE AMD METHODOLOGY - A SCENARIO .....	59
Step 1:	<i>Goal and objectives definition</i> .....	60
Step 2:	<i>Selecting suitable quality meta-metrics</i> .....	60

<i>Step 3: Selecting suitable evaluation metrics</i> .....	61
<i>Step 4: Weighting Metrics</i> .....	61
<i>Step 5: Measuring algorithms performance</i> .....	64
<i>Step 6: Testing significance of performance results</i> .....	64
<i>Step 7: Algorithmic fitness evaluation</i> .....	65
<i>Step 8: Ranking algorithms</i> .....	65
4.5. EXPERIMENTS AND EVALUATION.....	67
4.5.1. <i>Classifiers and datasets</i> .....	67
4.5.2. <i>Evaluation methodology and criteria</i> .....	67
4.5.3. <i>Experiments and analysis of the results</i> .....	71
4.5.3.1. Correctness: average Spearman’s rank correlation coefficient .....	71
4.5.3.2. Generalization of AMD: sensitivity and consistency analysis.....	73
4.5.3.3. Significance fitness evaluation .....	75
4.5.4. <i>Comparison with existing methods</i> .....	75
4.5.4.1. Statistical significance test for comparison of ranking methods.....	80
4.6. LIMITATIONS OF AMD METHOD FOR CLASSIFIER SELECTION .....	81
4.7. SUMMARY .....	83
CHAPTER 5 .....	84
CBR-BASED META-LEARNING AND REASONING FOR ACCURATE CLASSIFIER SELECTION .....	84
5.1. OVERVIEW .....	84
5.1.1. <i>Key Contributions</i> .....	85
5.2. CBR-BASED META-LEARNING AND REASONING (CBR-MLR) FRAMEWORK.....	86
5.2.1. <i>Definition of Algorithm Selection Problem</i> .....	87
5.2.2. <i>Architecture of CBR-MLR Framework</i> .....	87
5.2.2.1. Offline Phase: Creation of Classifier Selection Model.....	88
5.2.2.2. Online Phase: CBR-based Multi-level Multi-view Meta-Reasoning (CBR-MlvMr).....	89
5.2.3. <i>Methods of CBR-MLR Based Classifier Selection</i> .....	91
5.2.3.1. Multi-view Data Characterization - Meta-Features Extraction .....	91
5.2.3.2. Multi-view Classifiers Characterization – Multi-criteria Performance Analysis ..	94
5.2.3.3. Model Creation – Feature-vector (Propositional) Representation .....	96
5.2.3.4. CBR-based Multi-level Multi-views Meta-reasoning (CBR-MlvMr) .....	100
New Case Preparation: .....	101
CBR-based Multi-views Meta-reasoning (CBR-MvMr): .....	101
Classifiers Conflict Resolver: .....	104
5.2.4. <i>Implementation, experiments and evaluation</i> .....	105
5.2.4.1. Implementation .....	105
5.2.4.2. Experimental Setup.....	106
a. Classifiers used.....	106
b. Training and testing datasets.....	107
5.2.4.3. Evaluation methodology and criteria .....	107
5.2.4.4. Experiments and analysis of the results.....	108

5.3.	LIMITATIONS OF CBR-MLR FOR CLASSIFIERS SELECTION .....	110
5.4.	SUMMARY .....	111
CHAPTER 6 .....		112
SELECTION AND DESIGN OF ROUGH SET CLASSIFIER.....		112
6.1.	OVERVIEW .....	112
6.1.1.	<i>Key Contributions</i> .....	113
6.2.	HEURISTICS-BASED SELECTION OF THE ROUGH SETS CLASSIFIER .....	114
6.3.	DESIGN OF ROUGH SETS CLASSIFIER.....	116
6.4.	METHODOLOGY OF ROUGH SET CLASSIFIER DESIGN AND DEVELOPMENT .....	119
6.4.1.	<i>Patient Charts and Online Guidelines</i> .....	119
6.4.2.	<i>Charts and Guidelines Translation</i> .....	121
6.4.3.	<i>Rough Set-based Knowledge Acquisition</i> .....	123
6.4.3.1.	Preprocessing Phase .....	124
6.4.3.2.	Data Reduction Phase .....	126
6.4.3.3.	Rules Mining and Validation Phase .....	130
6.4.4.	<i>Hybrid Rule-based Reasoning</i> .....	131
6.4.5.	<i>Correlation-based trend analysis</i> .....	133
6.5.	<i>Experiments and Results</i> .....	135
6.5.1.	Evaluation Criteria .....	135
6.5.2.	Experimental Setup.....	136
6.5.3.	Results.....	136
6.5.4.	Comparison.....	140
6.6.	LIMITATIONS OF THE PROPOSED ROUGH SET CLASSIFIER.....	140
6.7.	SUMMARY .....	141
CHAPTER 7 .....		143
SELECTION AND DESIGN OF HYBRID-CBR CLASSIFIER .....		143
7.1.	OVERVIEW.....	143
7.1.1.	<i>Key Contributions of Hybrid-CBR Classifier</i> .....	144
7.2.	SELECTION AND DESIGN OF HYBRID-CBR CLASSIFIER .....	145
7.2.1.	<i>Rationales behind the selection of hybrid-CBR</i> .....	146
7.2.2.	<i>Design of hybrid-CBR classifier</i> .....	147
7.3.	A REAL-WORLD APPLICATION SCENARIO .....	148
7.4.	METHODOLOGY - DESIGN AND IMPLEMENTATION OF HRM AND HYBRID- CBR CLASSIFIER....	149
7.4.1.	<i>Architectural design and workflow</i> .....	152
7.4.2.	<i>Knowledge Acquisition</i> .....	155
7.4.2.1.	Rules Creation: Translating Guidelines .....	155
	Personal profile assessment .....	156
	Goal setting and plan management .....	156
	Physical activities assessment .....	158
	Rules creations .....	159

7.4.2.2.	Case Base Creation.....	161
7.4.3.	<i>Hybrid Reasoning and Recommendation</i> .....	162
7.4.3.1.	Rule-based reasoning and recommendation .....	162
	Level-1 RBR.....	163
	Level-2 RBR.....	164
	Level-3 RBR-CBR .....	167
7.4.4.	<i>Case-based Reasoning (CBR)</i> .....	168
7.4.4.1.	Retrieve and Reuse Steps.....	169
7.4.4.2.	Retain steps.....	173
7.4.5.	<i>Preference-based Reasoning (PBR)</i> .....	173
7.5.	EXPERIMENTS AND EVALUATION.....	175
7.5.1.	<i>Case-study: weight management</i> .....	175
7.5.2.	<i>Experimental setup</i> .....	176
7.5.2.1.	Environment .....	176
7.5.2.2.	Data and knowledge (rules/case base) .....	177
7.5.2.3.	Evaluation criteria .....	178
7.5.3.	<i>Experiments and Analysis of the Results</i> .....	178
7.5.3.1.	Experiment 1: Baseline-RBR system.....	180
7.5.3.2.	Experiment 2: Modified-RBR system .....	181
7.5.3.3.	Experiment 3: CBR system .....	182
7.5.4.	<i>Comparison of hybrid-CBR with jColibri</i> .....	186
7.6.	LIMITATION OF THE PROPOSED HYBRID-CBR CLASSIFIER.....	188
7.7.	SUMMARY .....	189
CHAPTER 8	.....	190
CONCLUSION AND FUTURE WORK	.....	190
8.1.	CONCLUSION.....	190
8.2.	FUTURE DIRECTIONS.....	192
8.2.1.	<i>Future perspective of AMD method</i> .....	192
8.2.2.	<i>Future perspective of CBR-MLR method</i> .....	193
8.2.3.	<i>Future perspective of rough-set classifier</i> .....	194
8.2.4.	<i>Future perspective of hybrid-CBR classifier</i> .....	194
BIBLIOGRAPHY	.....	196
APPENDIX	.....	210
LIST OF PUBLICATIONS	.....	210

## Table of Tables

---

TABLE 3.1. CONFUSION MATRIX OF THE CLASSIFIERS PERFORMANCE EVALUATION METRICS .....	31
TABLE 4.1. CATEGORIZATION OF CLASSIFIERS EVALUATION METRICS BASED ON QUALITY META-METRICS .....	46
TABLE 4.2. SAATY'S PREFERENCE SCALE FOR PAIR-WISE COMPARISON OF EVALUATION CRITERIA .....	50
TABLE 4.3. RANDOM CONSISTENCY INDICES (RI) FOR DIFFERENT NUMBER OF EVALUATION CRITERIA (N).....	52
TABLE 4.4. LIST OF WEKA WELL-KNOWN MULTI-CLASS CLASSIFIERS .....	53
TABLE 4.5. GENERAL CHARACTERISTICS OF UCI/OPENML REPOSITORIES DATASETS .....	59
TABLE 4.6. EXPERTS' GROUP-BASED RATING OF QUALITY METRICS FOR HETEROGENEOUS CLASSIFIERS .....	60
TABLE 4.7. EVALUATION METRICS FOR PERFORMANCE ANALYSIS OF HETEROGENEOUS MULTI-CLASS CLASSIFIERS.....	61
TABLE 4.8. ANALYTICAL HIERARCHY PROCESS (AHP) BASED RELATIVE CRITERIA WEIGHTING.....	62
TABLE 4.9. PARTIAL LIST OF AVERAGE STANDARD DEVIATION (AVERAGE CONSISTENCY) OF THE CLASSIFIERS.....	64
TABLE 4.10. CLASSIFIERS PERFORMANCE AND RANKING BASED ON RELATIVE DISTANCE FROM IDEAL ALGORITHM .....	65
TABLE 4.11. AVERAGE SPEARMAN'S RANK CORRELATION COEFFICIENT FOR 15 CLASSIFICATION DATASETS.....	71
TABLE 4.12. COMPUTATION OF SPEARMAN'S RANK CORRELATION COEFFICIENT ..	72
TABLE 4.13. SENSITIVITY ANALYSIS OF CLASSIFIERS WITH VARYING CRITERIA WEIGHTS.....	74
TABLE 4.14. ANALYSIS OF SIGNIFICANTLY POOR ALGORITHMS USING SIGNIFICANT FITNESS FUNCTION .....	75
TABLE 4.15. COMPARISON OF AMD METHOD WITH STATE-OF-THE-ART METHODS	77
TABLE 4.16. FRIEDMAN'S TEST STEPS TO COMPARE RANKING METHODS FOR STATISTICAL SIGNIFICANCE .....	82
TABLE 4.17. SUMMARY OF FRIEDMAN'S TEST RESULTS FOR COMPARING RANKING METHODS.....	82
TABLE 5.1. GENERAL VIEW (META-CHARACTERISTICS) OF CLASSIFICATION DATASET.....	92
TABLE 5.2. BASIC STATISTICAL VIEW (META-CHARACTERISTICS) OF CLASSIFICATION DATASET .....	92



TABLE 5.3. ADVANCED STATISTICAL VIEW (META-CHARACTERISTICS) OF CLASSIFICATION DATASET .....	93
TABLE 5.4. INFORMATION THEORITICS VIEW (META-CHARACTERISTICS) OF CLASSIFICATION DATASET .....	93
TABLE 5.5. LIST OF DECISION TREE CLASSIFIERS FROM WEKA LIBRARY .....	94
TABLE 5.6. CASE-BASE STRUCTURE AND FEATURE-VECTOR REPRESENTATION OF RESOLVED CASES .....	97
TABLE 5.7. DATASETS USED IN CASE-BASE CREATION WITH THEIR BREIF DESCRIPTIONS .....	97
TABLE 5.8. DECISION TREE CLASSIFIERS CHARACTERIZATION .....	104
TABLE 5.9. RESULTS OF META-LEARNING BASED METHOD FOR DECISION TREE CLASSIFIER SELECTION.....	108
TABLE 6.1. LIST OF GUIDELINES USED FOR MANAGING DIABETES MELLITUS .....	120
TABLE 6.2. SET OF REFERENCE RANGE RULES. ....	122
TABLE 6.3. MISSING VALUE TREATMENT, CRITERIA AND STRATEGIES, APPLIED TO THE DIABETES MELLITUS DATASET. ....	125
TABLE 6.4. CLINICAL CHARACTERISTICS OF THE DIABETES PATIENTS.....	126
TABLE 6.7. SET OF CUT-POINTS AND CORRESPONDING INTERVALS FOR DISCRETIZATION.....	128
TABLE 8.6. PARTIAL INFORMATION SYSTEM (TRAINING DATASET) IN INTERVAL FORMAT AFTER DISCRETIZATION. ....	129
TABLE 6.9. LIST OF ALL POSSIBLE REDUCTS AFTER APPLYING REDUCT OPERATION .....	129
TABLE 6.10. A PARTIAL RULES LIST EXTRACTED FROM DISCRETIZED INFORMATION SYSTEM USING LEM2 ALGORITHM .....	131
TABLE 6.11. EXPERIMENTAL SETUP FOR VALIDATION OF PREDICTION RULES IN ROSE 2 SYSTEM.....	136
TABLE 6.12. CONFUSION MATRIX DESCRIBING OVERALL OUTPUT OF THE VALIDATION PROCESS. ....	136
TABLE 6.13. AVERAGE ACCURACY (%) OF THE MODEL FOR INDIVIDUAL CLASS AND OVERALL MODEL.....	137
TABLE 6.14. PERCENT ACCURACY AND PERCENT ERROR FOR EACH TEST OF THE 10- FOLD.....	138
TABLE 6.15. EVALUATION PARAMETERS FOR COMPUTING BALANCED ACCURACY. .....	139
TABLE 6.16. COMPARISON OF THE ROUGH SET CLASSIFIER WITH STATE-OF-TEH- ART-CLASSIFIERS.....	140
TABLE 7.1. WEIGHT STATUS RULES (WSR) BASED ON BODY MASS INDEX (BMI).....	156
TABLE 7.2. GOALS AND WEIGHT MANAGEMENT PLAN RULES (GPR) .....	157
TABLE 7.3. DISTINCT-METS RULES FOR BASELINE-RBR.....	160

TABLE 7.4 RANGED-METs RULES FOR MODIFIED-RBR .....	161
TABLE 7.5. CASE BASE STRUCTURE .....	162
TABLE 7.6. LOCAL SIMILARITY MATRIX OF 'AGE GROUP' ATTRIBUTE .....	170
TABLE 7.7. PERSONAL PROFILE INFORMATION OF THE VOLUNTEERS FOR SYSTEM EVALUATION .....	175
TABLE 7.8. DISTRIBUTION OF THE PHYSICAL ACTIVITIES IN THE METs CASE BASE .....	177
TABLE 7.9. OUTPUT OF LEVEL-1- AND LEVEL-2 RULE-BASED REASONING MODELS .....	179
TABLE 7.10. RECOMMENDATION OF THE BASELINE RULE-BASED REASONING SYSTEM.....	180
TABLE 7.11. RECOMMENDATIONS GENERATED USING MODIFIED RULE-BASED REASONING SYSTEM.....	182
TABLE 7.12. RECOMMENDATIONS GENERATED USING CASE-BASED REASONING METHODOLOGY .....	183

## Table of Figures

---

FIGURE 1.1. MOTIVATION FOR CLASSIFIER PERFORMANCE EVALUATION AND SELECTION OF BEST CLASSIFIER .....	4
FIGURE 4.1. EVALUATION OF ALGORITHMS ON THE BASIS OF MULTIPLE EVALUATION CRITERIA .....	36
FIGURE 4.2. AMD METHODOLOGY FOR CLASSIFIERS PERFORMANCE EVALUATION	40
FIGURE 4.3. CLASSIFICATION MODEL OF THE CLASSIFIERS QUALITY META-METRICS .....	45
FIGURE 4.4. CATEGORIZATION OF CONSTRAINTS DEFINED OVER EVALUATION CRITERIA .....	58
FIGURE 4.5. CRITERIA RELATIVE WEIGHTS, ESTIMATED USING ANALYTIC HIERARCHY PROCESS.....	63
FIGURE 4.6. EVALUATION METHODOLOGY OF RECOMMENDED RANKING AGAINST IDEAL RANKING .....	68
FIGURE 4.7. COMPARISON OF RECOMMENDED RANKING (RR) AND IDEAL RANKING (IR).....	73
FIGURE 4.8. COMPARISON OF THE AMD METHOD WITH STATE-OF-THE ART METHODS.....	79
FIGURE 5.1. CBR-BASED META-LEARNING AND REASONING FRAMWORK FOR CLASSIFIER SELECTION.....	88
FIGURE 5.2. MULTI-VIEW REPRESENTATION OF CLASSIFICATION DATASET BASED ON META-CHARACTERISTICS.....	91
FIGURE 5.3. MULTI-CRITERIA BASED CLASSIFIERS PERFROMANCE EVALUATION METHOD.....	95
FIGURE 5.4. META-FEATURES EXTRACTION GOT NEW CASE CREATION .....	106
FIGURE 5.5. CBR-BASED REASONING FOR BEST CLASSIFIER SELECTION .....	106
FIGURE 6.1. ROUGH SET CLASSIFICATION AND PREDICTION.....	115
FIGURE 6.2. HYBRID ROUGH SET CLASSIFICATION MODEL (H2RM) FOR PREDICTION .....	117
FIGURE 6.3. CLINICAL ENCOUNTER OF PATIENT IN SOAP PROTOCOL FORMAT. ...	120
FIGURE 6.4. DISTRIBUTION OF PATIENT’S OBSERVATIONS IN CLINICAL CHART. ...	121
FIGURE 6.5. CORRELATION-BASED TREND ANALYSIS FOR PROGNOSIS .....	134
FIGURE 6.6. TEST RESULTS OF EACH PASS OF THE 10-FOLDS CROSS VALIDATION PROCESS. ....	138
FIGURE 7.1. HYBRID CASE-BASED REASONING CLASSIFIER.....	147
FIGURE 7.2. ABSTRACT VIEW OF THE REAL-WORLD APPLICATION MINING MINDS .....	148

FIGURE 7.3. DESIGN STRATEGIES OF THE PROPOSED HYBRID REASONING METHODOLOGY .....	150
FIGURE 7.4. FUNCTIONAL DIAGRAM OF PROPOSED MULTIMODAL HYBRID REASONING MODEL .....	151
FIGURE 7.5. DETAILED DATA FLOW DIAGRAM OF THE PROPOSED MULTIMODAL HYBRID REASONING MODEL .....	153
FIGURE 7.6. DISTRIBUTION OF SUBJECTS BASED ON AGE FACTOR.....	161
FIGURE 7.7. COMPARISON OF BASELINE-RBR, MODIFIED-RBR AND HYBRID-CBR SYSTEM.....	185
FIGURE 7.8. COMPARISON OF BASELINE-RBR, MODIFIED-RBR AND HYBRID-CBR .....	185
FIGURE 7.9. PERFORMANCE OF HYBRID-CBR FOR DIFFERENT THRESHOLDS .....	186
FIGURE 7.10. PERFORMANCE OF HYBRID-CBR AND JCOLLIBRI SIMILARITY FUNCTIONS .....	187

### 1.1. Background

In real-world domains, organizations try to build intelligent decision support systems and tools for automating their organizational processes and analyzing the available data for future predictions and strategic planning. For this purpose, the organizational experts and machine learning practitioners adopt the available decision making methods and algorithms and apply them for their problems in hand. These stakeholders pick the appropriate decision making method based on their heuristics knowledge about the domain problems and the available decision making methods. Once the algorithm is selected, the corresponding decision making model *i.e.*, classification or recommendation model is built for real-world services generation in the form of intelligent decisions. Each domain application has its own constraints and requirements, such as some applications need higher accuracy while others need lower computational complexity and robustness. Similarly, some of the domains need to have the classification models with higher accuracy, lower computational and space complexity and consistent and comprehensible results. Other criteria that can be used for evaluating and selecting classification models include scalability, integration, stability, and interestingness [1]. This shows that the selection of classification algorithm for the decision making process of an application is a challenging task and need a number of aspects to be considered. This makes the process difficult for experts and machine learning practitioners to heuristically pick an algorithm. This requires a proper methodology to evaluate the classifier from the perspective of the domain constraints imposed by the application scenario and the strengths and weaknesses of the classifiers itself. Historically, this process of the evaluation of classifiers has been done by estimating predictive accuracy via cross-validation tests and receiver operating curves (ROC) analysis. However, the features

greatly vary from domain to domain and has been shown that different evaluation methods are suitable for different domain problems. Furthermore, the evaluation of algorithms based on the combination of more than one suitable criteria results in good performance results [2, 3].

As a result of involvement of more than one criteria, for the classifiers evaluation and selection, the task of algorithm selection can be modeled as a multiple criteria decision making (MCDM) problems. Different MCDM methods evaluate classifiers from different aspects and produce different rankings results [4]. The literature of classifiers evaluation and selection can be categorized into the following three types, keeping in view the involvement of the human experts (*i.e.*, domain expert or machine learning expert or practitioner). Firstly, the expert uses his heuristic knowledge about the domain application and the available algorithms and pick the appropriate one. This approach is mainly applicable in real-world application scenarios, where the dataset need to be properly prepared and then used for model creation. Secondly, empirical performance evaluation approaches are used, which focus on the experimental results analysis of all the candidate algorithms and then applying some multi-criteria decision making method to rank the alternatives. This approach involves the selection of right evaluation criteria for comparing the results of these algorithms and then a proper methodology to rank them correctly. The third and the last way is to use automatic selection method using meta-learning approaches where meta-features of the dataset are exploited and accordingly an appropriate algorithm is recommend. However, this approach requires the creation of a machine learning model based on historical datasets which is a difficult task. In this method, to build an automatic classifier recommendation model, a training dataset is required whose features will come from the data meta-characteristics and the class labels from the empirical evaluation of the classifiers performances.

In first part of this thesis, focus is on the automatic selection of classifiers for classification data problem, while in the second part, focus is on the heuristic-based selection of classifier and designing accurate classifier meeting the domain

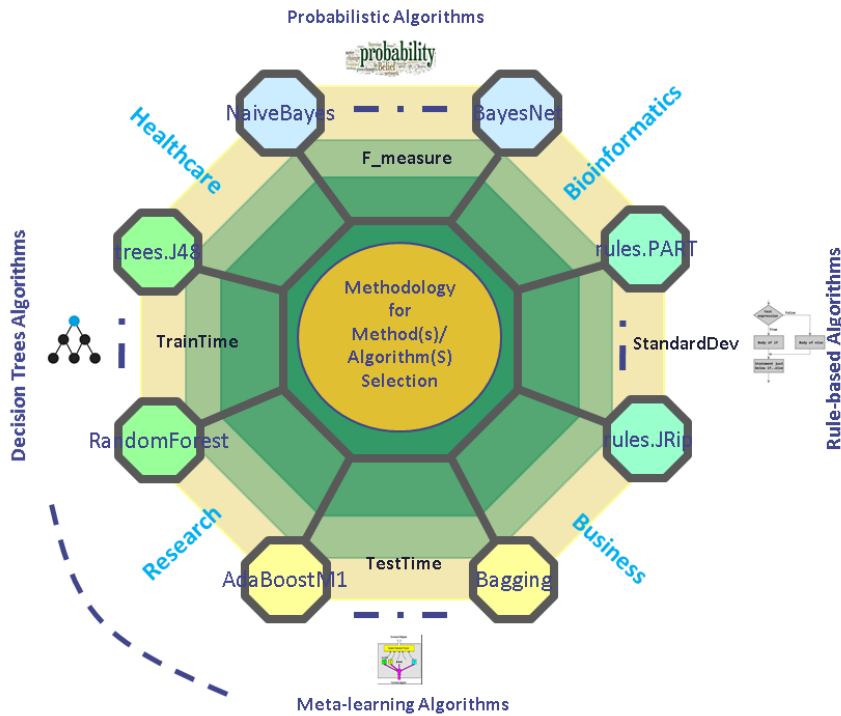
application's requirements. In the first part, a novel multi-criteria decision making (AMD) methodology is developed that consist of a set of intelligent methods for evaluating classifiers performances and ranking them to find the top-ranked classifier for learning the domain data in-hand. This helps the experts to easily pick the top classifier as the best one for their problems in hand and consequently design the corresponding classifier. Similarly, an automatic classifier selection framework, CBR-based meta-learning and reasoning (CBR-MLR), is proposed and implemented that utilizes the data and classifiers meta-characteristics to first build a classifier recommendation model and then automatically recommend the best classifier for a given new data problem (user's dataset).

In the second part, two classifiers are designed for real-world applications scenarios. The first classifier is designed using the rough-set classifier with enhancements in the data preparation and discretization steps for getting semantics-preserved accurate results. The second classifier is built for accurate and precise recommendation generation using hybrid case-based reasoning (hybrid-CBR) methodology. In the design of this classifier, first a training dataset (Case-Base) is prepared from the domain knowledge using the rule-based method and then the case retrieval step is enhanced with accurately defined similarity functions.

## **1.2. Motivation**

The advancement of ubiquitous technologies and its adoption in real domain applications, such as trade and business, healthcare and bioinformatics, various industries, and education and research (shown in Figure 1.1) has greatly increased the availability of data. The organizations in these domains are trying to analyze their data by building prediction models for knowing insights of the businesses operations and making long-term businesses strategies. This has gained the attention of researchers in the areas of machine learning and data mining to apply appropriate machine learning algorithms for generating real-world application services. However, a large number of classifiers are available and its number increasing day-by-day. Each classifier has its own set of qualities that bring certain strengths and

weaknesses, when they are applied in real domain applications for real services generation. Some of these qualities includes: correctness, robustness to noise, scalability, computational complexities (training speed), responsiveness (prediction speed), model comprehensibility and interpretability, robustness to noise and redundant features, robustness to numeric features, storage complexity, and others [5]. Moreover, various domains have their specific requirements in the form of domain constraints, such as in some domains accuracy is compromised over the computational complexity while in others computational complexity or storage complexity are compromised over the accuracy. The qualities of the classifiers need to be assessed prior to their application in the domain. This brings the attention of the machine learning experts into the well-known *no-free lunch theorem* [6], which states that no algorithm can perform well on all kinds of dataset and hence no algorithm is universally acceptable for all types of problems, given an evaluation criterion or multiple evaluation criteria.



**Figure 1.1.** Motivation for classifier performance evaluation and selection of best classifier



All these restrictions and the complexities make it very hard for the machine learning experts, domain experts and end users to accurately pick suitable classifier from the large set of available classifiers and build accurate prediction, and classification model(s) for user's problem in-hand. Besides overlooking specific qualities of the classifiers, domain data meta-characteristics, specific constraints and requirements, a common drawback in existing classifier selection methods is that they only consider predictive accuracy as the classification performance metric. However, it has been proved that it is insufficient in domains which suffer from the class imbalance problems. Therefore, an accurate methodology, which efficiently integrate different methods necessary for selection of best classifier is of interest. This will enormously reduce the time, effort and cost of the machine learning experts, practitioners and the business owners, and will results in accurate domain models developments for real-world applications.

In addition to the primary motivation presented above, this thesis provides some applications specific solutions, which can equally be applied on general data mining processes, such as generating an accurate dataset or training cases using domain knowledge, building semantics-preserving interpretable and incremental learning-based classification models.

### **1.3. Problem Statement**

Researchers have designed a variety of methods to select accurate classifier and build classification models for generating services in various real-world applications. The selection of best classifier is followed by the standard data mining process to first design the model and then develop it properly. These researchers have greatly contributed in research community, however some of the challenges still need to be overcome because they may vary from application to application and one learning algorithm to another learning algorithm. The key issues that occur in the selection and design of classification and recommendation model for real-world application scenarios include: evaluating classifiers heuristically based on multiple criteria and selecting the appropriate one, domain data acquisition from different sources,

representations of the instances and cases of the data in datasets or case-bases, and preserving semantics of the data during discretization of the continuous values, and ensuring efficient and accurate retrieval of cases from case-bases, during the case-based reasoning process. However, the selection of classifier using the practitioners' heuristic knowledge limits the evaluation process to a single quality or performance metric. This results in the selection of a sub-optimal performance classifier for decision making that may mislead the user in taking the recommended action. Similarly, a common limitation of the existing classifier evaluation methods is the use of only predictive accuracy as the classification performance metric, which has been proven insufficient in domains with class imbalance and many others problems.

In case of classifiers performance evaluation, there is no universally acceptable classifier that outperforms all other classifiers on every kind of domain data, given a single evaluation criterion or multi-metrics evaluation criteria. Similarly, there is no universally acceptable guidelines or rules for the selection of suitable evaluation metric(s) to evaluate the classifiers. Other related issues regarding classifier evaluation include: the experts' preferences (i.e., weights on the criteria) are normally defined using absolute values that lack the consistency check for insuring that the assigned weight are correct, global and local constraints of the domain and evaluation metrics which sometimes impose restrictions on the classifiers performance evaluation process and must need to be satisfied. Moreover, there is lack of a universally acceptable classifier evaluation strategy, which includes almost all the required multiple-criteria including consistency measure to insure the selection of optimum performance consistent classifier. Apart from the issues highlighted in state-of-the-art classifiers performance evaluation methods, the automatic classifiers selection using meta-learning also suffers from a number of challenging issues. These include: the extraction and selection of a suitable set of meta-characteristics of the data to best represent the intrinsic behaviors of the dataset from all aspects and thus help in automatic recommendation of best classifier and enabling multi-views multi-level meta-learning and reasoning for accurately selecting classifiers based on data and classifiers characteristics.

## 1.4. Proposed Concept

The proposed research work, presented in this thesis, is structured into two parts, each of which has several chapters. In Part I, the focus is on the development of accurate methods for the selection of right classifier based on multi-criteria decision making and meta-learning. Accurate methodologies are proposed to empirically evaluate classification algorithms on the basis of multiple performance metrics satisfying user's domain constraints and automatically select best classifiers based on the data and classifiers meta-characteristics using CBR-based approach. In the first solution, an accurate multi-criteria decision making (AMD) methodology is proposed, which integrates a series of novel methods for the selection of suitable performance metrics, relatively assigning consistent weights to each metric, satisfying the domain and experts' constraints, ranking algorithms with respect to an ideal algorithm and selecting the top-ranked classifier. The detail of this method is described in Chapter 4. In the second solution, a novel CBR-based meta-learning and reasoning (CBR-MLR) framework is proposed and implemented that utilizes data and classifiers meta-characteristics during multi-level multi-views case-based reasoning to accurately recommend best decision tree classifier for users' applications in-hand. In this method, 29 meta-characteristics are extracted from user data and 09 decision tree classifiers are empirically evaluated, using predictive accuracy and consistency, to design a Case-Base. Accurate case retrieval functions are defined and the CBR output is refined with classifiers conflict resolution approach that uses weight sum score and AMD methods. This method is described in detail in Chapter 5.

In Part II, the thesis is focused on classification and recommendation tasks and the related issues which may appear in real-world application scenarios, such as: domain data acquisition for real-world datasets and cases preparation, semantics-enabled discretization, accurate case similarity functions definitions and accurate classification and recommendation models creation. In this part of the thesis, the expert's heuristics based approach is applied for selection of classification and recommendation methods and building the associated models. Based on the heuristics selection, an accurate rough set-based classification model is proposed for a real-

world application scenario of diabetes mellitus where the data is composed of patients encounters structured in clinical charts. The rough set classifier's selection is made based on its capabilities of building a comprehensible and interpretable model and best learning the rough boundaries of different classes in the dataset tool [7, 8]. The discretization phase is enhanced by introducing a semantics-preserving discretization scheme that preserves the semantics in the transformed data, in the rules. The detail are in Chapter 6. Similarly, for another real-world wellness application scenario of physical activity recommendations, a hybrid case-based reasoning (hybrid-CBR) method is heuristically selected for generating accurate and precise wellness recommendations that closely match the users' requirements. A hybrid-CBR recommendation model is proposed with an enhanced rule-based case preparation methodology along with accurately defined similarity functions. The advantage of the proposed hybrid-CBR model, in comparison to the state-of-the-art rule-based models, is that it generates relevant recommendations even if there is no exact match of the input test case. The detail are in Chapter 7.

## **1.5. Contributions**

The goal defined for the thesis is accurate classifier selection for user' learning problem and designing classifiers for accurate decision making in real-world application scenarios. To achieve this goal, the objectives set are: (1) evaluation of classifiers performances and selection for accurate classifier for users' application in-hand. The achievement of this objective is based on the correct and consistent choice and weighting of the classifiers performance evaluation metrics for defining a general purpose aggregate metric to rank the classifiers and select the one with highest rank (2) design of accurate rough-set and hybrid-CBR classifiers for real-world applications with semantics-preserving data discretization and accurate case retrieval similarity functions definition.

The main challenges faced in successfully achieving the stated goal and the corresponding objectives includes: how to select suitable performance metrics for classifier evaluation (how much to select and how to aggregate) from the available

large set of metrics, how to quantify and estimate the user's preferences on the selected performance criteria in relative and consistent manner as compared to absolute mechanism, how to satisfy the users' local and global constraints, in the form of cost and benefit criteria, consistency performance measure and significance fitness evaluation function to lead into the selection of optimum performance consistent algorithm(s). Similarly, for automatic classifiers selection using meta-learning approach, the challenges faced are: how to extract data and classifiers meta-characteristics, how to know the extracted features are enough for accurate recommendation of the classifiers and how to establish relationship among the data and classifiers characteristics. In the same way, how to prepare real-world application dataset and cases from data sources, how to discretize the domain data so that the semantics remains intact, and how to accurately define case retrieval functions.

To resolve the highlighted challenges, this thesis presents the idea of empirical evaluation and ranking of classifiers using multi-criteria decision making, selection of right classifier using meta-learning and reasoning, and systematic analysis, design and enhancement of some of the standard data mining processes during the rough-set and hybrid-CBR classifiers design.

The main contributions made through this thesis are described as follows.

**Accurate classifier selection using AMD methodology:** According to the well-known "*no-free lunch theorem* [6]", no classifier can be found which best perform than all the others classifiers on every type of learning problem, based on certain given evaluation metric(s). Similarly, there are no generally accepted rules which specify the correct and suitable metric(s) and help in assigning consistent relative weight for prioritizing the individual metrics in the generalized aggregate evaluation criteria. There is also no method that help in specifying the specific domain context/constraints while evaluating the algorithms. In this thesis, an accurate multi-criteria decision making (AMD) methodology is proposed, which integrates a series of novel methods for the selection of suitable performance metrics, relatively assigning consistent weights to each metric, satisfying the domain and experts'

constraints and ranking algorithms with respect to an ideal algorithm. In this thesis, an extensive analysis of the most commonly used classifiers performance metrics is done and the suitability of each metric to a particular domain context is assessed. A classification model is built for describing these contexts and associating them to the domain constraints, which helps the expert in easily selecting the suitable metric. An experts' group-based method is proposed to accurately selecting the suitable metrics from the metrics classification model. A General purpose aggregate metric, comprising the accuracy, time complexity (comprising both training and testing time) and consistency measures, is proposed and the algorithms' performance are ranked with respect to ideal algorithm using the relative closeness concept of TOPSIS method. The AMD methodology is validated and extensively experimented on fifteen publically available classification datasets from UCI and OpenML repositories and thirty five freely available classification algorithms from the heterogeneous families of classifiers implemented in Weka tool. The empirical results and comparison with state-of-the-art methods have demonstrated that the proposed AMD method outperforms the existing methods. The AMD achieved an average Spearman's rank correlation coefficient of ninety seven ( $R_s$ , 0.97) with respect to the ideal ranking of these algorithms. The detail are in Chapter 4.

**CBR-based meta-learning and reasoning (CBR-MLR) methodology:** The key contributions made through this methodology are as follows. A flexible and incremental meta-learning and reasoning based framework is proposed which uses CBR-based methodology integrated with multi-criteria decision making, for classifier evaluation, and data characterization using multi-view meta-features extraction. Similarly, a new multi-metrics criteria is proposed for the evaluation of decision tree classifiers to select the best classifier as class label for the cases in training dataset (i.e., resolved cases in the proposed CBR methodology). Furthermore, classifiers are analyzed based on their predictive accuracy and standard deviation, called consistency to select the best classifier as class-label. The idea of multi-view learning is proposed to learn the data from multiple perspectives, with each perspective representing a set of similar meta-features that reflects one kind of behaviors of the

data. Each set of features is called a family that forms a view of dataset. Moreover, a multi-level multi-view meta-reasoning methodology is proposed with a flexible and incremental learning model integrating CBR with the classifiers conflict resolving (CCR) method to accurately recommend the most similar case as the suggested classifiers for a given new dataset. For the CBR retrieval phase, accurate similarity matching functions are defined, while for the CCR method, weighted sum score and AMD method (presented in Chapter 4) are proposed. This methodology is described in detail in chapter 5.

**Design of a semantics-preserving accurate rough set classifier:** real-world application data, exhibit the characteristic of variations or uncertainty and vagueness in their values [9]. The majority of classification algorithms have not been initially designed for dealing such kind of vague and ambiguous values within a dataset. In literature, some techniques, especially fuzzy approaches are available can solve the issue [9, 10], however they depend on several factors. So, we proposed a rough set classification model that is originally based on the classical rough sets theory [11], which needs no other factors and parameters except the dataset in the form of an information system. The rough set classifier's selection is made based on its capabilities of building a comprehensible and interpretable classification model and best approximation of the rough boundaries of different classes in the dataset. In the real-world diabetes scenario, the diabetes dataset (i.e., information system) is prepared from the semi-structured clinical notes using the subjective, objective, assessment, and plan (SOAP) protocol for the clinical notes data. Moreover, the discretization phase of the rough set classifier is replaced by a new semantics-preserving discretization scheme that preserves semantics in the transformed data from continuous values to discrete values in the knowledge rules. The existing discretization methods used in literature distort the original clinical semantics of the data when they are transformed to their discrete form. For both, the information system preparation and discretization, online guideline-enabled rules-based reasoning methodology is used. The proposed rough set classification model is evaluated on the real-world diabetes scenario, which produces highly accurate and

semantics preserved results of 0.959% on a dataset of 391 records and eight attributes. The detail are in Chapter 6.

**Design of an accurate and precise hybrid-CBR classifier:** There is no universally acceptable algorithm that can solve every type of domain problem, especially when there are a lot of variations in the values of the attributes, the number of instances in the dataset are minimum and the number of class are high or even one instance per class. In such cases, the traditional classifiers cannot perform well and suffer from the problem of over-fitting. The traditional classifiers work on the principal of generalization rather than specialization and exact matching strategy are used when decisions are made. Therefore, in this thesis, an incremental learning approach is proposed and implemented in the form of a case-based reasoning classifier. In CBR methodology, the essential part is the creation of accurate train and test cases. So, an accurate rule-based case preparation methodology is proposed with and accurate similarity functions for case retrieval during the recommendation generation process. The proposed hybrid-CBR model is tested and evaluated in a real-world application scenario of physical activity recommendation that has shown significant performance results with respect to state-of-the-art methods. The hybrid-CBR model is also evaluated in a real-world application scenario of physical activity recommendations' and compared with standard rule-based recommendation models. The evaluation results demonstrates that hybrid-CBR is significantly better that the state-of-the-art methods. The detail are in Chapter 7.

## **1.6. Thesis Organization**

This dissertation is organized into six chapters as follows.

**Chapter 1: Introduction.** Chapter 1 provides an overview of the research work in the area of data preparation for standard data mining process, selection of the appropriate classification and recommendation method/algorithm and creation of the corresponding model(s). The chapter describes the motivation behind the research thesis in the area of algorithm performance evaluation and classification and recommendation models selection and creation. Moreover, the research problem is



formulated, the overall concept of the proposed solution is highlighted and the contributions and uniqueness made are presented.

**Chapter 2: Related work.** A background detail of the related work is provided in this chapter. The state-of-the-art research work in the area of rough set classification in diverse domain is presented along with their comparative analysis. Similarly, the chapter also summarizes the relevant literature in the area of hybrid case-based reasoning for health and wellness applications. Furthermore, the chapter is started with an extensive literature review of the relevant work in the area of empirical performance evaluation of classifiers based on multi-criteria analysis techniques. The meta-learning based literature is also summarized in this chapter.

**Chapter 3: Machine learning and classification: technical preliminaries.** This chapter is focused on the preliminaries of machine learning techniques used for the classification problems. The classification task is discussed from the classifiers performance evaluation perspective. Multi-criteria decision making is highlighted, which assists in the process of evaluating and ranking classifiers with respect to ideal algorithms. Meta-learning and reasoning based terminologies are defined and described.

**Chapter 4: Accurate empirical evaluation of classifiers.** This chapter describes the proposed solution to the problem of selecting suitable classification algorithm from the set of available thirty five algorithms using multiple performance evaluation metrics. The proposed methodology, accurate multi-criteria decision making (AMD), is described from its initial step of goal setting to the final step of ranking and selection of best classifier.

**Chapter 5 CBR-based meta-learning and reasoning for accurate classifier selection.** A flexible and incremental meta-learning and reasoning based framework is proposed which uses CBR-based methodology integrated with multi-criteria decision making, for classifier evaluation, and data characterization using multi-view meta-features extraction. Similarly, a new multi-metrics criteria is proposed for the evaluation of decision tree classifiers to select the best classifier as class label for the

cases in training dataset. Furthermore, classifiers are analyzed based on their predictive accuracy and standard deviation, called consistency to select the best classifier as class-label. The idea of multi-view learning is proposed to learn the data from multiple perspectives, with each perspective representing a set of similar meta-features that reflects one kind of behaviors of the data. Each set of features is called a family that forms a view of dataset. Moreover, a multi-level multi-view meta-reasoning methodology is proposed with a flexible and incremental learning model integrating CBR with the classifiers conflict resolving (CCR) method to accurately recommend the most similar case as the suggested classifiers for a given new dataset. For the CBR retrieval phase, accurate similarity matching functions are defined, while for the CCR method, weighted sum score and AMD method are proposed.

**Chapter 6 selection and design of semantics-preserving accurate rough set classifier.** This chapter describes the proposed rough set classification methodology for generating semantically preserved accurate classification results. A rough set classification algorithm is presented and validated using a real-world application scenario from healthcare domain with diabetes dataset.

**Chapter 7: selection and design of an accurate hybrid case based reasoning classifier.** This chapter describes the proposed hybrid case-based reasoning (CBR) methodology for generating accurate and precise recommendation decisions. The integration of a rule-based reasoning (RBR) methodology is presented with the case-based reasoning approach to enable the process of accurate case preparation, at real-time, and suggestion of relevant recommendations. Guidelines-based rules creation process is highlighted in the real-world application scenario of physical activity recommendations and a case base of successful recommendations is prepared. Accurate similarity functions are defined for the correct retrieval of the relevant recommendation cases and providing as the final recommended decisions.

**Chapter 8: Conclusion and future work.** This chapter concludes the work done with the possible future directions, intended to be taken care in future.

#### 2.1. Overview

In real-world domain applications, suitable classifiers selection, their design and the associated methodologies have been widely used since long. The key challenge a machine learning practitioner face during designing a machine learning system is which classifier to use for building the proposed model. Similarly, to design an accurate classifier using the recommended algorithms, further tasks are required, such as the preparation of correct datasets using standard data mining process, i.e., preprocessing, discretization, training dataset preparation for building classifier and many others. To resolve these issues, comprehensive research have been taken and a large number of methods, techniques, frameworks and methodologies have been proposed in literature. This chapter first presents the relevant literature in the area of multi-criteria decision making for empirically evaluating the performance of classifiers and ranking them to select the top ranked algorithm. It also presents the related studies for suitable classifier selection using meta-learning approaches that consumes meta-characteristics of the data. Similarly, the chapter also presents the related work in the area of classifier design for real-world applications in medical and wellbeing area with specific focus on rough set classifiers and hybrid case-based reasoning classifier.

The choice of algorithm for classifier design can be either done automatically using algorithms performance analysis and ranking or using meta-learning approach or heuristic selection by the machine learning practitioner. In this chapter, first the automatic classifier selection literature is evaluated that uses multi-criteria decision making methods and meta-learning approaches and then the experts' heuristics based evaluation method are analyzed. The heuristic-based approach is studied in the real-

world application domains of prediction in diabetes mellitus and health and wellness applications of physical activities recommendations. The automatic and heuristics-based classifier selection and design define the flow of this chapter.

## **2.2. Automatic Classifier Selection**

Selection of a suitable classifier for a dataset or a user's problem in hand is a complex task and depends on many characteristics of the domain problem. Similarly, the process requires performance analysis of the candidate classifiers/algorithms to know which algorithm is best performing for certain type of data. The subsequent subsections summarize the related studies in the area of selection of best classifier based on multi-criteria decision making and meta-characteristics of the dataset in hand.

### **2.2.1. Multi-criteria decision making for classifiers ranking and selection**

Machine learning algorithm selection is a real-world problem in various domains, such as data mining business, knowledge acquisition and reasoning, research and many others areas [12]. Large business firms and research institutions hire machine learning experts, such as practitioners, data analysts and knowledge engineers to analyze the business data for different types of strategic planning. Usually, experts choose appropriate machine learning algorithm(s) using their heuristic knowledge about the domain and the available classification algorithms [13]. The heuristics-based algorithm(s) selection is a risky task and sometimes result in selection of a sub-optimal performance algorithm(s). The reasons may include lack of the complete knowledge about the domain application, i.e., the datasets have different intrinsic characteristics, and the candidate classifiers have different capabilities and strengths. This process become more challenging when the selection of best classifier is based on multiple-criteria under strict conditions and constraints. According to the well-known "no free lunch" theorem [6], no machine learning algorithm performs well on all kind of learning problems. However, it can be made possible to estimate the selection of a suitable machine learning algorithm for an application in hand [14]. This selection process of the classifiers is an application dependent task, because it

has been theoretically and empirically proved that no machine learning algorithm is universally superior on all datasets due to the different characteristics and features of the domain data [15].

In real-world applications, the requirements assessment of the applications and deciding which specific qualities need to be evaluated has great importance. Clear application's requirements easily clarify the ingredients of evaluation criteria and their individual contributions in the final decision making [3]. The evaluation methods for different domains are different due to different objectives of the domains. Some domains require single evaluation criteria, while others need multi-criteria evaluation. In classification problems, the most commonly used single criterion metric used for evaluations is the accuracy, which can be evaluated using the well-known metrics, such as area under the ROC curve [16], success rate, average accuracy, and balanced accuracy. However, the evaluation only on the basis of accuracy may misleads the selection of optimum performance algorithm [2]. To select optimum performance algorithm, multiple evaluation criteria, such as average accuracy, execution time, training time, consistency and many others need to be used. The objective of multi-criteria evaluation is to balance the trade-off between these criteria rather than maximizing a single criterion [3]. The main issue in multi-criteria evaluation is the selection and prioritization of suitable criteria and excluding those which have conflicting behaviors. This is a subjective issue and requires the involvement of stockholders, such as domain experts and machine learning practitioners and users [2]. In the criteria weight assignment, experts' preferences are quantified as weight scores and assigned to each metric of the evaluation criteria. The weights can be either assigned manually by experts or can be done using some semi-automatic weighting method, such as analytical hierarchy process (AHP) [17]. The manual weight assignment is a hard task, which has been realized by the simple and intuitive measure (SIM) [18], measure-based evaluation (MBE) [19] and application-oriented validation and evaluation (APPROVE) [2] approaches. Statistical methods [20, 21] have also been used for the evaluation of machine learning algorithms from different.

Apart from the criteria selection and prioritization issues, the non-uniformity of dimensionality of data for the evaluation metrics is another challenging issue [22]. To overcome this issue, a number of normalization techniques [2, 23] have been proposed in literature in which the unit or scale of measurements are transformed to a common compatible format to be fairly used in the evaluation process [24].

In literature, a number of studies can be found that evaluates classifiers on the basis of single evaluation criterion, such as accuracy [25-30]. The evaluation of classification algorithms on the basis of multiple criteria, such as accuracy and time, in non-simultaneous way, is presented in [31-33] and on the basis of sensitivity, precision, F-score, and area under the curve (AUC) is presented in [34]. Ali and Smith [35] performed evaluation among 8 classifiers with 100 different classification problems using extended measures of average accuracy (true positive rate, true negative rate and percent accuracy) and time complexity (training time and testing time). Similarly, for various real-world applications, the performance evaluation of various classifiers have been done, for examples, handwritten recognition [36], color prediction of rice paddy plant leaf [37], prediction of diabetes mellitus [38, 39]. The most commonly used criteria for algorithms evaluation are the adjusted ratio of ratio (ARR) [32] and performance of algorithm (PAI<sub>g</sub>) on dataset [40], which use accuracy and time. Reif et al., [41] used root mean squared error (RMSE) and Pearson product-moment correlation coefficient (PMCC) [42] for the evaluation and recommendation of the best classification algorithm. The methods discussed in literature use absolute or partial relative weights to prioritize evaluation criteria. However, recently, the focus of researchers has shifted to relative criteria weighting, using multi-criteria. In medical knowledge acquisition, relative criteria weighting has been proposed [43] that uses AHP process [17]. They used average training time, accuracy and memory usage as the criteria. Five multi-criteria decision making methods, including TOPSIS [44], elimination et choix traduisant la réalité III (ELECTRE III) [45], grey relational analysis, vlse kriterijumska optimizacija i kompromisno resenje (VIKOR), and preference ranking organization method for enrichment of evaluations II (PROMETHEE II) have been discussed in article [4].

### 2.2.2. Meta-learning and reasoning for classifier selection

In the area of machine learning applications, users are usually inexperienced with the details of the plethora of available classification algorithms and thus do not recognize which algorithm is appropriate for their problem at hand. The reason is that if algorithm  $A$  outperforms algorithm  $B$  on a specific dataset  $D1$  then  $B$  may outperform on other dataset, say  $D2$ , in which case  $A$  may fail. This gives us an idea that no single algorithm performs well on all types of datasets and thus validates the known theorem of “No Free Lunch” [46]. As the performance of a specific algorithm depends on the problem/dataset at hand, therefore an automatic recommendation system is needed to assist the users while picking an algorithm for learning the data. Automatic algorithms selection has been extensively studied since 1990s. At the start, cross-validation strategy was used but soon discouraged due to computational cost [26]. In parallel to cross-validation, some of the early work focused on meta-learning and empirical method to select appropriate learning algorithm [47]. Using meta-learning approach, meta-features of the datasets are calculated and the performance of a variety of learning algorithms is measured on these datasets. After this, mapping between problem features and algorithm performance is learned for recommending appropriate algorithm [48]. Problem and algorithm characterization, using meta-learning, and defining mapping function between problem features and algorithm performance is the most widely used approach to algorithm selection problem. Diverse machine learning approaches, such as C4.5 [49], rule-based classifier [35], linear regression [27] and k-NN [32] have been applied to learn the mapping function to select the algorithm. Some of the work, such as [50] has characterized complexities of the problems and performance of the algorithms and used for selecting appropriate algorithm. Recently, Q. Song et al. [40] has used a new dataset characterization method for computing datasets features and computed performance of seventeen classification algorithms over 84 UCI publically available datasets[51]. They have learned used k-NN to select the  $k$  nearest algorithms from the list of 17 algorithm and recommend to the user.

A large number of classifiers characteristics have been introduced in literatures to understand natures and intrinsic behaviors of the classification problems. These characteristics are categorized into a number of families, such as basic statistical characteristics, advanced statistical, information theoretic, complexity, landmarking and model-based [21] [40].

A meta-learning approach is an alternative to the AMD methodology, where the characteristics of a large number of classification datasets are extracted and mapped against the best classifier (computed using AMD methodology) to create a training dataset for building an automatic classifiers selection model.

The above mentioned methods map relationship between the problem characteristics and algorithms performance using single learner using single family of data characteristics and don't take into consideration the multi-view multi-level learning and reasoning using CBR approach to recommend the best closet classifier if there is no exact matching classifier available for a given dataset.

### **2.3. Heuristics-based classifier selection and design for real-world applications**

In real-world application scenarios, where the candidate algorithms' evaluation is really a hard problem due to the unavailability of suitable quantifiable criteria and one may not get an appropriate algorithm in an acceptable time, applying some arbitrary choices or educated guesses, then experts' heuristic-based choices are the best options to use [52]. In this approach, the expert uses his knowledge about the domain and the candidate algorithms and picks suitable algorithm for designing an accurate classifier. A heuristic is a kind of algorithm that does not explores all the possible aspects of the candidate algorithms and the domain application requirements, but still tries to explore the most likely ones. The heuristic approach defiantly excludes the obviously bad algorithms from the competition. In this thesis, the heuristic approach is applied in the specialized domains of diabetes mellitus and wellness application. The subsequent sub-sections describe studies used in these area



for selecting suitable classifiers and the way they are designed to generate classification results.

### **2.3.1. Rough Set Classifier Selection and Design for Real-world Application**

In medical diagnosis, it is quite difficult for physicians to take diagnosis decision by evaluating the current conditions of a patient without referring to the previous decisions with the similar symptoms. For the reason, a number of clinical decision support systems (CDSS) [53] [54] [55] [56] [57] have been developed that assist physicians [58] . Such systems have widely been applied for diagnosis, prediction, classification and risk forecasting of different diseases from EMR data. The area of risk forecasting of diabetes type-2 has been explored from EMR data with the use of machine learning techniques, such as Gaussian Naïve Bayes, Logistic Regression, K-nearest neighbor, CART, Random Forests and SVM [53]. Ensemble of SVM and BP NN is used over Pima Indian publically available UCI dataset to predict presence of diabetes [54] with the improved predictive accuracy than the traditional single learning method. Stahl [55] has proposed a Linear and Bayesian Ensemble Modeling technique to predict glucose level in DM patient data. They have evaluated their model with 47 patients' data and validated with 12 datasets. Similarly, a prototype diabetic decision support system, based on multi-layer perceptron neural network model has been developed [56] that predicts psychosocial well-being behavior, such as depression, anxiety, energy and positive well-being of patients. In this system, patient's biological or biographical variables, such as age, gender, weight and fasting plasma glucose are used as input predictors. In literature [57], an architecture of multi-stage DM prediction system, based on fuzzy logic, neural network and case based reasoning (CBR) is proposed that uses two stages for prediction. In the first stage, base classifiers are used, whose results are forwarded to the second level which uses a rule-based reasoner (RBR) for refinement of the results. Chen [58] have used fisher linear discriminate analysis (FLDA), support vector machine (SVM) and decision tree (DT) to predict type-2 diabetes based on several elements in blood and chemometrics of the diabetes patients. The elements considered in this research

includes: lithium, zinc, chromium, copper, iron, manganese, nickel and vanadium. Authors of this work constructed ensemble classifiers on the training set and selected the best one which is validated on independent test dataset. Likewise, prediction of T2DM, from the electronic health records (EHR) is done using ensemble of random forest and gradient boosting machine models [59]. In the same way, prediction of the onset of type-1 diabetes in juvenile subjects is examined using neural networks, decision trees and their ensembles [60]. In a recent research on prediction of T1DM and T2DM, boosting ensemble model is used that internally uses random committee classifier as the base classifier and enhance prediction accuracy to 81% [61].

Apart from the listed literature, rough sets theory (RST), a powerful mathematical tool [7, 8], has successfully been applied in medical diagnosis and prediction. For example, toxicity predictions [62], medical expert system rules creation [63], pneumonia patient's death prediction [64], chest pain prediction [65] and a lot others [66] are treated using RST. For diabetes prediction, RST is applied over Pima Indian dataset [67] that has produced 75% accuracy [68]. Similarly, for investigating relationship between psychosocial variables in Kuwaiti diabetic children, RST builds classifier function that correctly classifies patients [69]. RS-based data analysis of the genetic data of children with T1DM is performed [70] for rules extraction and prediction of children with genetic susceptibility to T1DM. This system recommends pre-diabetes therapy to patient, if they are susceptible to type-1 diabetes. A similar research for children with T1DM, in Poland, can also be found in literature [71].

Apart from prediction of diabetes into its types whether using traditional machine learning methods or rough sets techniques, future risk prediction is an important research issue and treated with different approaches. For example, risk prediction of T2DM using multivariate regression model [72], prediction of T2DM in elderly Spanish population with high cardiovascular risk, using multivariate cox regression model [73]. Other risk prediction models for type-2 diabetes can be found in the systematic review article [74]. A multivariate logistic regression equation has been developed and validated with non-diabetic Egyptian subjects data that has sensitivity of 62%, specificity 96%, and positive predictive value of 63% [75].

### **2.3.2. Hybrid-CBR Classifier Selection and Design for Wellness**

Human experts are limited in number and expensive in terms of healthcare and wellness services provided. Healthcare decision support systems play effective roles in overcoming the shortage of human experts and improving quality of life with better services [76]. Decision support systems rely on automatic reasoning methodology for their decisions. Most of these systems are based on a single methodology for reasoning, such as CBR or RBR [77], among others. Nevertheless, a few use multiple reasoning approaches with a certain integration strategy. The integration of multiple reasoning methodologies in a single system has attracted increased attention in the research community due to the improved performance with respect to accuracy. The analogy of integration of reasoning methodologies is adopted from the decisions made by domain experts, who rely on multiple knowledge sources rather than a single source. Domain experts use information from general guidelines, clinical trials, and past successful cases to arrive at a final decision. In automatic reasoning systems, the concept of multimodal reasoning methodology evolved from the use of heterogeneous knowledge sources to generate the final decision [77]. The knowledge source, such as guidelines and past successful cases are modeled as knowledge rules and case bases that require RBR and CBR for their executions.

The integration of reasoning approaches can follow any set of strategies, such as RBR followed by CBR, CBR followed by RBR and RBR and CBR in parallel [77, 78]. In the first strategy, RBR is used as the main methodology for making the decision. If RBR fails, CBR is used [79]. In the second strategy, CBR is used for the master reasoning process and RBR is used to refine the decision [80]. An example of this strategy is reasoning system for diabetes management [81]. The CBR refines the rules for the final outcome, specific to the patient's requirements. In other combinations, CBR and RBR are used in parallel, where either both outcomes are simply displayed or the best one is displayed based on some criteria. An example of parallel integration is the WHAT system [82, 83], which is used for training beginning sports medicine students to design exercise regimens for patients with cardiac or pulmonary disorders. The regimens are produced by RBR and CBR in parallel and presented to the experts

for choosing the best one. Other methodologies exist that closely cooperate with each other for generating final decisions [84, 85]. Apart from RBR and CBR, filtration-based approaches, such as content-based filtration [86] and collaborative filtration [87, 88] are also popular in the area of recommender systems for online shopping, product selection, and healthcare services. Preference-based recommender systems are used in e-applications such as e-commerce to offer alternative or cross-selling products to customers [89].

In the healthcare domain, hybrid reasoning approaches have been frequently used. In treatment planning for adolescent early intervention, hybrid CBR that uses RBR and fuzzy theory has been implemented [90]. For supporting physicians for the management of diabetes mellitus, integration of CBR, RBR and model-based reasoning (MBR) [91] and web-based CBR [92] has been proposed. For cancer decision support services, CBR has been integrated with RBR. The CBR part is used to adapt the production rules for decision making [85]. A recent research study [93] integrates rough set theory and correlation analysis in a hybrid model, called H2RM, that predicts the diabetes type and manages patient observations for future trend analyses. Other similar studies can be found that focus on heart disease [76] and oncology [77], among others.

In the wellness domain, the knowledge acquisition and reasoning engine (KARE) [94] is used in activity awareness for human-engaged wellness applications (ATHENA) [95] to promote active lifestyles. KARE uses the hybrid reasoning methodology by integrating the Random Forest, Naïve Bayes, and IB1 approaches. KARE generates food, physical activity, and music therapy recommendations for ATHENA users. For the elderly, an intelligent personalized exercise recommendations system is proposed [96] that utilizes the user's health status, goals and preference information. Similarly, a hybrid CBR/RBR approach has successfully been used for designing nutritional menus [97].

All of these methodologies have the common basis of being used in an exclusive manner. They do not guarantee a minimization of the shortcomings of RBR and CBR, which are discussed as follows:

- Conventional RBR systems lack the capability of specializing recommendations for individuals. In general, to deal with specific requirements of users and provide user-centric specialized recommendations, it is necessary to gradually increase the number of rules in the knowledge base. This approach not only results in knowledge base intractability problem, but also causes maintenance and combinatorial explosion issues [98].
- Standard CBR systems provide solutions for new problems using a large and unbiased case base as implicit knowledge. However, the requirement of a large case base is a difficult task and associated with a number of other issues, such as physical storage, proper indexing and computational complexities [99]. The preparation of the query cases to feed the CBR cycle for generating physical activity recommendations is a challenging task.
- There have been significant improvements in the integration of these methodologies in hybrid systems [100]; however, a number of challenging issues still need to be resolved for applying integration in the wellness domain.

### **2.3.3. Trade-off criteria for evaluating heuristic approach for classifiers selection**

To evaluate whether the heuristic-based approach adopted for the evaluation of classifiers and other recommendation methods and algorithms is efficient or not, the following set of criteria can be used [101].

- Optimality: When several algorithms exist for a given problem, does the heuristic guarantee that the best algorithm will be found? Is it actually necessary to find the best solution?
- Completeness: When several best algorithms exist for a given problem, can the heuristic find them all? Do we actually need all solutions?

- Accuracy and precision: Can the heuristic provide a confidence interval for the claimed algorithm? Is the error bar on the solution unreasonably large?
- Execution time: Is this the best known heuristic for solving this type of problem? Some heuristics converge faster than others. Some heuristics are only marginally quicker than classic methods.

By analyzing the evaluation criteria of the heuristic-based algorithms performance analysis, it tells that selecting the appropriate algorithm based on these criteria may not ensure the right algorithm.

## **2.4. Summary**

This chapter has summarized state-of-the-art techniques, methodologies, approaches, frameworks, tools and models that are used for the selection and design of accurate classifiers for users' applications in-hand. Firstly, the relevant literature on ranking of classifiers and selection of suitable one based on multiple performance criteria is presented. The algorithms' empirical performance evaluation and analysis methods, techniques and methodologies are critically analyzed and compared. Secondly, the literature on meta-learning based classifier selection methods is evaluated and described. Lastly, relevant literature on the classifiers used in medical and wellness applications is analyzed and described in detail that finally lead to the heuristic-based selection and design of two rough set and CBR classifiers for diabetes predictions and physical activity recommendations.

## Chapter 3

# Machine Learning and Classification: Technical Preliminaries

---

### 3.1. Overview

This chapter is about to describe the key concepts used in this thesis. The basic concepts of data mining, machine learning, classification, classifiers, performance evaluation, decision making, multi-criteria decision making and their techniques, meta-learning and reasoning are provided for easy understanding and grasping the idea presented in the subsequent chapters of this dissertation.

### 3.2. Data mining

The process of discovering interesting patterns and knowledge from large amounts of data is termed as data mining [102]. The sources for data can be databases, datasets in different formats (e.g., text file etc.), warehouses, Web, or streamed data.

#### 3.2.1. Technologies used in data mining

Data mining is an interdisciplinary research area that uses many techniques from statistics, machine learning, pattern recognition, database and data warehouse systems, information retrieval, visualization, algorithms, high-performance computing, and many application domains [103].

### 3.3. Machine learning

Machine learning is one of the important areas of research in artificial intelligence that tries to make computer programs intelligent to automatically learn from large volume of historical data and recognize complex patterns for making intelligent decisions [104].

**3.3.1. Supervised learning** is the type of machine learning in which the learning process from the training data is supported by the labelled examples. It is a synonym for classification.

**3.3.2. Unsupervised learning** is the learning process in which the input examples are not class labeled. It is a synonym for clustering.

### **3.4. Classification**

The task of machine learning process that finds a classification model (or function) for distinguishing data classes of the categorical or nominal types. The model is created on the basis of analysis of the training examples in the training data, which is used to predict the class label of new examples with unknown labels [105].

#### **3.4.1. Classification techniques**

##### **3.4.1.1. Decision tree induction**

Decision trees is a family of classification algorithms, which build flowchart-like trees models from a labeled training dataset [106]. The internal nodes of the tree represent conditions of a rules and the leaf nodes represents decisions. The most commonly used techniques for selecting the attributes for the node of the tree are: Information Gain, Gini Index, Minimum Description Length (MDL), and Multivariate Splits used [106]..

##### **3.4.1.2. Bayes classification methods**

Bayesian classifiers are statistical learners, based on the well-known Bayes' theorem, that learn prior probabilities and likelihoods from the training dataset to estimate the posterior probability and predict the class labels for unclassified test examples [103]..

##### **3.4.1.3. Rule-based classification**

Rule-based classifiers is the family of comprehensible and interpretable classifiers which learn training data using the sequential covering algorithm and the rules generated are represented in the form of IF-THEN rules [103].



**3.4.1.4. Meta-learning or classifiers ensemble methods**

Classifiers ensemble or meta-learning algorithms ensemble a series of  $k$  learned models, using some combination method with the aim of creating an improved composite classification model [107]. The individual model is termed as base classifiers. When a new unresolved case is given to the model for classification, the model collects decision from each base classifiers and combine them to a single unified decision. Some of the most popular methods are bagging, boosting, stacking, voting and random forests etc.

**3.4.1.5. Case-based reasoning for classification**

Case-based reasoning (CBR) classifiers use a set of resolved cases as the training dataset for solving new problems cases using the similarity measures approaches [108]. The similarity, among the new case and the resolved historical cases (called case base) is measured using Euclidean distance. When a new case is provided for classification, a case-based reasoner takes over the control and checks for identical cases, using the similarity function, in the case base. If exact match is found, the solution part of the matched case is provided as the classification or recommendation decision, otherwise the closest one to the input case is suggested as the class label.

**3.4.1.6. Rough sets classification**

Rough set theory is one of the most powerful tool used for classification to discover structural relationships within imprecise, vague, and noisy data that has rough classes boundaries [7]. Before applying the process of classification, it applies the discretization process to the continuous-valued attributes, because the theory works well on the discrete information. In real-world data, some of the classes cannot be differentiated based on the available attributes set. Rough sets theory is used to roughly estimate such classes by using the concepts of lower and upper approximation. The lower approximation consists of all those example of the training dataset which are certainly belonging to a particular class with no ambiguity [7]. Similarly, the upper approximation consists of all those instances that do not certainly

belonging to the class of upper approximation. Rough set theory approximates the classes that cannot be distinguished certainly based on the available condition attributes into rough sets. From these approximated classes, decision rules are generated which are then used during the online classification process.

### **3.4.2. Evaluation and selection of classifiers**

The situation, where more than one classifiers are available and we want to choose the “best” out of them, we need to perform classifiers evaluation process, which is referred as model selection or classification algorithm selection [103].

#### **3.4.2.1. Metrics for evaluating classifier performance**

To evaluate performance of classifiers, a set of evaluation criteria are used that are referred as performance evaluation metrics. The most commonly used metric is the predictive accuracy, which can be measured using a specific formula that consumes the following set of atomic evaluation metrics.

- True positives (TP): These refer to the positive instances correctly classified by a classifier.
- True negatives (TN): These are the negative instances correctly classified by a classifier.
- False positives (FP): These are the negative instances incorrectly classified by a classifier as positive
- False negatives (FN): These are the positive instances misclassified by the classifier as negative

**Table 3.1.** Confusion matrix of the classifiers performance evaluation metrics

		Predicted class		
		Yes	No	Total
Actual class	Yes	TP	FN	P
	No	FP	TN	N
	Total	P'	N'	P+N

In addition to the accuracy-based measures, classifiers can also be compared with respect to additional characteristics, such as speed, robustness, scalability, interpretability and space complexity etc.

#### 3.4.2.2. Cross-validation

Cross-validation is a rotation estimation process in which a model built is validated for assessing how the results will be get generalized for an independent test dataset [103]. In k-fold cross-validation, the data are randomly partitioned into k mutually exclusive datasets called folds i.e.,  $D_1, D_2, \dots, D_k$ , with approximately equal size. In first iteration  $i$ , dataset  $D_i$  is reserved as test dataset, and the rest datasets,  $D_2, \dots, D_k$ , are used as train datasets for the model creation. In the second iteration  $D_1, D_3, \dots, D_k$  are used as train datasets and  $D_2$  as the test dataset. This process is repeated for each fold/dataset and finally the average is taken as the collective result of the model.

### 3.5. Binary and multiclass classification

Classification algorithms that have the capabilities of classifying data only in two classes are referred to as binary classifiers, while those that can classify data into multiple classes are multiclass classifiers [103]. Support vector machines is an example of binary classifier while J48 is an example of multi-classifier.

### 3.6. Decision making process

The study of identifying and selecting alternative solutions/algorithm(s) based on the actual performance results of the alternatives/algorithms and the preferences of the

decision maker(s) is called decision making for algorithm selection. The objective of decision making is choosing the best out of the available alternative algorithms which best fits the goals, objectives, desires, values, and so on of the domain experts [109].

### **3.6.1. Multi-criteria decision making**

The decision making process made on the basis of multiple criteria to select the best option from the available multiple alternatives is referred as multi-criteria decision making. It is also termed as multi-attribute decision making.

#### **3.6.1.1. Analytic hierarchy process**

Analytic hierarchy process (AHP) [17] is a multi-criteria decision making approach used to convert subjective assessments of relative importance to a set of overall scores or weights and evaluate the alternatives. The methodology of AHP process follows the procedure of pairwise comparisons.

#### **3.6.1.2. Technique for Order Preference by Similarity to Ideal Solution**

Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) [110] is another multi-criteria decision making method that works on the idea of ranking alternatives based on the shortest distance from the ideal solution and farthest distance from the negative-ideal solution. The distance is computed using Euclidean distance.

### **3.7. Meta-learning for algorithms selection**

“Meta learning is a subfield of machine learning where automatic learning algorithms are applied on meta-data about machine learning experiments”[111].

#### **3.7.1. Meta-features of datasets and algorithms**

Meta-features of a dataset are the aggregate characteristics of that dataset, such as general, statistical, information-theoretic, complexity and landmarking that represents its global qualities. Similarly, characteristics of the learning algorithm, such as type of parameters, their settings, and various measures for evaluating

algorithms performance are all examples of algorithms' meta-features or meta-characteristics [111]

### **3.7.2. Meta-learner for algorithms selection**

A learning algorithm that learns meta-features or meta-characteristics of a large number of datasets and relate them to the meta-characteristic(s) of a set of candidate algorithms, e.g., predictive accuracy etc., is termed as meta-learner or meta-classifier [112].

### **3.7.3. Meta-reasoner for algorithms selection**

A algorithm or classifier that reasons over the already learned meta-characteristics for the meta-features of a given new learning problem (dataset) to predict the performance of the closest learning algorithms is called meta-reasoner. Hence, a meta-reasoner can correctly select the algorithm best suited for the new problem, if the induced relationship holds, i.e., the meta-learner has modeled the relationship well in advance [113, 114].

## **3.8. Summary**

This chapter has provided the basic concepts, terminologies, definitions, techniques, methodologies and tools, used in this thesis. Machine learning is described in terms of classification problem. The well-known families of classification algorithms are defined. The performance evaluation of classifiers is discussed and the associated multi-criteria decision making techniques, such as AHP and TOPSIS are defined. The concept of meta-learning, meta-characteristics and meta-reasoner are described which are used to select best classifiers for a new learning problems (dataset).

## Chapter 4

### **Multi-criteria Decision Making for Classifier Selection**

#### **4.1. Overview**

Manual evaluation of machine learning algorithms and selection of a suitable classifier from the list of available candidate classifiers, is highly time consuming and challenging task. If the selection is not carefully and accurately done, the resulting classification model will not be able to produce the expected performance results. In this chapter, we present an accurate multi-criteria decision making methodology (AMD) which empirically evaluates and ranks classifiers' and allow end users or experts to choose the top ranked classifier for their applications to learn and build classification models for them. Existing classifiers performance analysis and recommendation methodologies lack (a) appropriate method for suitable evaluation criteria selection, (b) relative consistent weighting mechanism, (c) fitness assessment of the classifiers' performances, and (d) satisfaction of various constraints during the analysis process. To assist machine learning practitioners in the selection of suitable classifier(s), AMD methodology is proposed that presents an expert group-based criteria selection method, relative consistent weighting scheme, a new ranking method, called optimum performance ranking criteria, based on multiple evaluation metrics, statistical significance and fitness assessment functions, and implicit and explicit constraints satisfaction at the time of analysis. For ranking the classifiers performance, the proposed ranking method integrates Wgt.Avg.F-score, CPUTimeTesting, CPUTimeTraining, and Consistency measures using the technique for order performance by similarity to ideal solution (TOPSIS). The final relative closeness score produced by TOPSIS, is ranked and the practitioners select the best performance (top-ranked) classifier for their problems in-hand. Based on the extensive experiments performed on 15 publically available UCI and OpenML datasets using 35 classification algorithms from heterogeneous families of classifiers, an average Spearman's rank correlation

coefficient of 0.98 is observed. Similarly, the AMD method has showed improved performance of 0.98 average Spearman's rank correlation coefficient as compared to 0.83 and 0.045 correlation coefficient of the state-of-the-art ranking methods, performance of algorithms (PAlg) and adjusted ratio of ratio (ARR). The evaluation, empirical analysis of results and comparison with state-of-the-art methods demonstrate the feasibility of AMD methodology, especially the selection and weighting of right evaluation criteria, accurate ranking and selection of optimum performance classifier(s) for the user's application's data in hand. AMD reduces expert's time and efforts and improves system performance by designing suitable classifier recommended by AMD methodology.

#### **4.1.1. Key Contributions**

The key contributions made through the proposed multi-criteria decision making methodology (AMD), for the objective of best classifier selection, are summarized as follows.

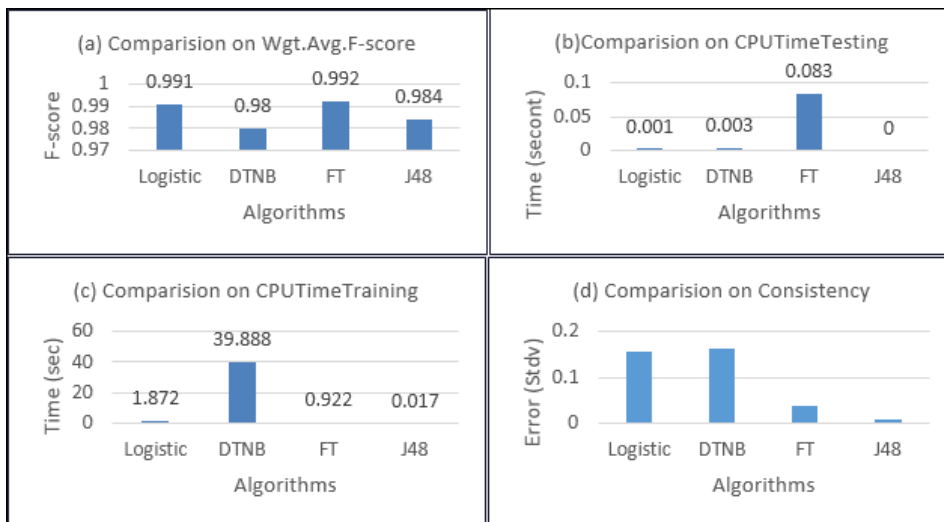
- A list of general guidelines are defined for performance evaluation of classifiers, based on extensive literature study of the classification algorithms.
- We analyzed and categorized classification algorithms' evaluation metrics and introduced the concept of classifiers quality meta-metrics (QMM) to construct QMM classification model, which is useful for non-experts of machine learning who need to make evaluation decision about classifiers selection. The QMM model further assists users in understanding physical meanings of the evaluation metrics.
- Proposed an experts' consensus-based group decision making method that assists experts to first select appropriate QMM and then select suitable evaluation criteria, satisfying interdependence and explicit global constraints, enforced by the objectives of the end user's application.
- An expert group-based relative criteria weighting technique is proposed, which can easily quantify and estimates experts' relative preferences about each evaluation criterion.

- A new ranking criteria, called optimum performance ranking (OPR) is proposed, which ranks classifiers based on Wgt.Avg.F-Score, CPUTimeTraining, CPUTimeTesting and Consistency performance metrics, integrated using TOPSIS method.
- Accurate statistical significance and fitness evaluation functions are defined, which inspect algorithms' fitness, prior to their inclusion in the final list of candidate algorithms for ranking.

Implicit and explicit constraints are defined at different levels of the evaluation process for accurate ranking of the classifiers.

## 4.2. Algorithm selection: multi-metric decision making process

Each machine learning algorithm performs differently on different datasets because of different features of the data. The evaluation of these algorithms on the basis of single criterion sometimes misleads the decision of selecting best algorithm from a list of available candidate algorithms. For example, consider the following scenario with four classification algorithms: multinomial logistic regression, decision table/naive Bayes hybrid classifier (DTNB), functional trees (FT) and J48 which are tested on anneal dataset [115] using 10x10-fold cross validation and evaluated using the criteria, Wgt.Avg.F-score, CPUTimeTesting, CPUTimeTraining and average consistency, as shown in Figure 4.1.



**Figure 4.1.** Evaluation of algorithms on the basis of multiple evaluation criteria



Figure 4.1(a) shows that FT algorithm performs well, measured in terms of weighted average f-score (0.992%) and is the winner amongst all algorithms. However, it performs poor from the CPUSpeed perspective (0.083 second). Similar interpretations can be made for CPUSpeed and the Consistency criteria. This analysis shows that no algorithm can be declared for all criteria.

From the empirical evidence, predictive accuracy is one of the traditional evaluation metric, estimated using cross-validation [116] that focuses on maximizing the accuracy, but ignores other criteria, such as comprehensibility, interestingness [117] and complexity. The formal measurement of comprehensibility and interestingness may not be possible like accuracy, but it more relevant than accuracy when the objective is discovering accurate knowledge [2] in medical domain for recommendation generation services. Similarly, time and space complexities are also the key criteria for evaluating algorithms and selecting the right algorithm for an application in hand. In situation, where the datasets are either large or the storage space or computational power is limited [118], the time and space complexities criteria need to be used for evaluation of the algorithms. Thus, in order to select appropriate classifiers or algorithms for such applications we must need to evaluate algorithms performance in terms of space and time complexities.

In light of the results shown in Figure 4.1 and the empirical evidences from the literature, the well-known no-free-lunch theorems [6] is confirmed. Hence, we conclude the discussion that no classification algorithms is superior on all problems and is therefore no single evaluation criterion is always superior for their evaluation. If one algorithm outperforms others on one criterion, it may underperforms on other criteria. As a consequence, the algorithm selection problem is a multiple criteria decision making problem which requires an accurate methodology to evaluate them properly. The rest of the study is focused to find a solution to this problem.

### **4.3. Methodology – multicriteria evaluation of classifiers**

In this section, first we define a set of general guidelines and then describe the methodology for evaluating classification algorithms on the basis of multiple evaluation criteria.

#### **4.3.1. Guidelines for algorithms evaluation**

For selecting suitable algorithm(s), a sequence of essential tasks need to be performed. To efficiently perform these tasks, a set of guidelines are presented as follows.

- 1** Define an unambiguous goal for which the algorithm(s) need to be selected
- 2** Analyze and specify goal as either single-objective or multi-objectives and specify the corresponding quality meta-metrics (QMM)
  - a.** Categorize objective(s) as cost and benefit criteria
  - b.** Define essential constraints on the objective(s), reflecting goal's constraints
- 3** Analyze the specified objective(s) and constraints against existing criteria
  - a.** If existing criteria work, then go to step 4.
  - b.** If existing criteria do not fit well, then go to step 5.
- 4** Evaluate the algorithms performances using the available criterion under the constraints, defined in step 2(b), and rank them for the best selection
- 5** Define a generic multi-metrics evaluation criteria using the following steps
  - a.** Analyze QMM for conflict among evaluation criteria (interdependence/fuzziness)
  - b.** Select suitable QMM, defining the objectives.
  - c.** Select suitable evaluation metrics for the selected QMM (objectives)
  - d.** Prioritize the selected evaluation metrics
  - e.** Rank algorithms based on the aggregate value of the weighted metrics
  - f.** Repeat step 5, if any of the constraints, defined in step 2(b), is not satisfied

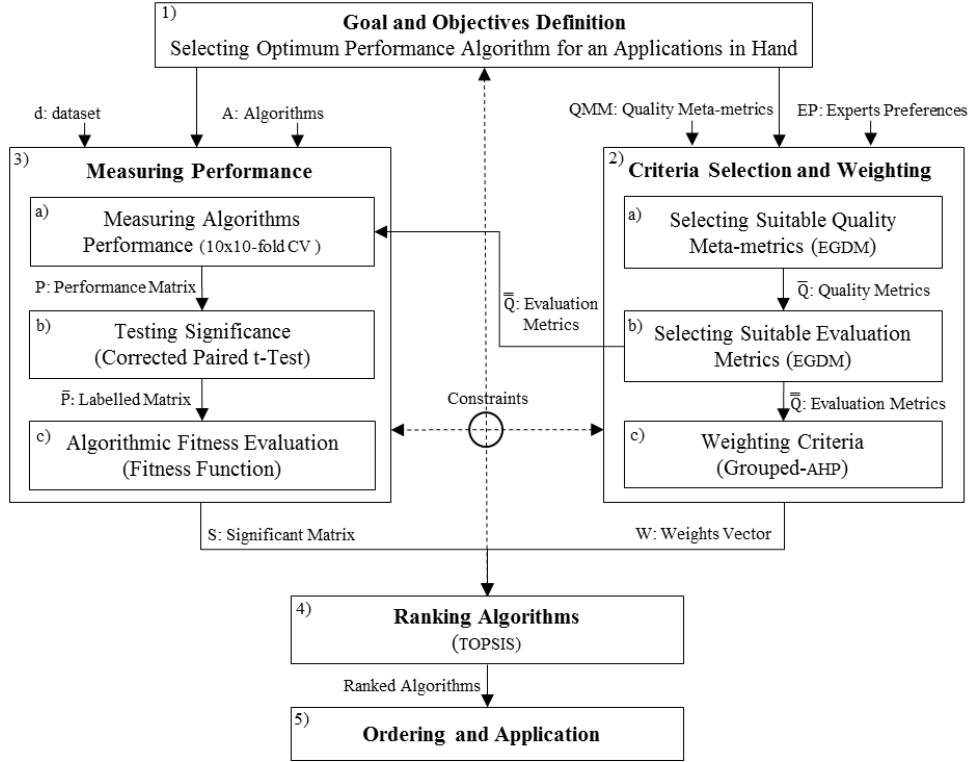
In the above guidelines, steps 1-4 are straightforward and can be easily followed. However step 5 is more challenging and needs technical contributions to accomplish the task of selecting suitable algorithm in the basis of multiple criteria. Generally, the outlined guidelines are generic, where only the domain specific parameters, such as the goal, objectives, evaluation criteria, and weights for each criterion need to be strictly followed while building a system. These guidelines are mainly focused on two essential aspects of the algorithms evaluation and recommendations systems. These aspects include (a) how to integrate multiple evaluation criteria and (b) what criteria should be integrated. To answer the first question, we designed and proposed a list of guidelines that were partially presented by [3] and [2]. Similarly, to extend answer of the first question and find solution to the second question, we have provided detail description in the next section.

#### **4.3.2. Multi-metric decision making for algorithm selection**

The proposed accurate multi-metric decision making methodology (AMD) consists of the following steps: goal and objectives definition, criteria selection and weighting, measuring algorithm performance, ranking algorithms, and ordering and application as shown in Figure 4.2.

Abstractly, the working methodology of AMD is described below.

- **Goal and objective definition:** describes the final goal, its corresponding objectives and the associated constraints to achieve the goal. For example, the selection of optimum performance classification algorithm for multi-class problems. In this statement, goal G is the “selection of optimum performance classification algorithm” and the global constraint C is “multi-class problems”. The corresponding objectives against this goal can be, e.g., ( $o_1$ ) accuracy, ( $o_2$ ) computational complexity, and ( $o_3$ ) consistency.



**Figure 4.2.** AMD methodology for classifiers performance evaluation

- **Criteria selection and weighting:** contains a set of methods to first select quality metrics for the objectives, then select suitable metric for each of the quality and finally assign consistent weight to each metric.
- **Measuring performance:** includes the tasks of generating performance results for the selected criteria using the candidate algorithms (considered in the study) on the datasets (one at a time) and performing significance and fitness tests. The purpose of this step is to generate significant matrix of the algorithms performance results for the selected evaluation criteria.
- **Ranking algorithms:** is used to rank the list of candidate algorithms by utilizing their performance results and the criteria weights.
- **Ordering and application:** consists of the trivial functions, such as sorting the ranked algorithms and selecting the top-k for the user's application in hand.
- **Constraints:** represent restrictions, i.e., for which family/families of problems the methodology should be activated (single class/multi-class),

how experts' preferences should be quantified (explicit criteria weights), introduction of special criteria as constraint i.e., consistency, which is measured in terms of standard deviation.

The proposed AMD methodology is algorithmically represented in algorithm 1.

---

**Algorithm 1.** Selection of optimum performance algorithm on the basis of multi-metric evaluation.

---

**Begin**

**inputs:**  $d$  – dataset

$A = \{a_1, a_2, \dots, a_n\}$  // list of  $n$  algorithms

**output:**  $R$  = top- $k$  algorithms; where,  $R \subseteq A$

**Let**  $QMM$  = Classifiers quality meta-metrics; // See section 4.3.2.1.

- 1 [Define Goal]  
 $G = \{o_1, o_2, \dots, o_n\}$ ; // where,  $n$  is the number of objectives, See section 4.3.2.
- 2 [Select Suitable Quality Meta-metrics]  
 $\bar{Q} = \text{selectSuitQuality}(QMM, G)$ ; // See section 4.3.2.1
- 3 [Select Suitable Evaluation Metrics]  
 $\bar{\bar{Q}} = \text{selectSuitEvalMetrics}(\bar{Q}, G)$ ; //where,  $\bar{\bar{Q}} \subseteq \bar{Q}$ . See section 4.3.2.2.
- 4 [Estimate Relative Weight of the Evaluation Metrics]  
 $W = \text{estimateRelativeWeights}(\bar{\bar{Q}})$ ; //where,  $W$  is weight vector. See section 4.3.2.3.
- 5 [Generate Performance Results of the Algorithms]  
**foreach** algorithm  $a$  in  $A$  perform 10x10-fold CV in Weka to produce an  $n*m$  performance matrix  $P$  for the evaluation metrics  $\bar{\bar{Q}}$ . See section 4.3.2.4.  
**end for**
- 6 [Perform Statistical Significance Test]  
 $\bar{P} = \text{performStatSigTest}(P)$ ; //where,  $\bar{P}$  is significance labelled matrix. See section 4.3.2.5.
- 7 [Perform Algorithm Fitness Test]  
 $S = \text{Perform Algorithm Fitness Test}$ ; See section 4.3.2.6, equation 8
- 8 [Compute Relative Closeness (RC) to Ideal Algorithm]  
 $RC = \text{rankAlgorithms}(S, W)$ ; See section 4.3.2.7.
- 9 [Rank the Algorithms]  
 $\text{RankedList} = \text{RANK.AVG}(RC_1, RC_1: RC_n, 1)$ ;
- 10 [Select Top-K Algorithms]  
 $R = \text{selectTopK}(\text{RankedList}, k)$ ;
- 11 apply  $R$  to learn  $d$

**End**

---

In algorithm 1, each step of the methodology is explicitly described in separate section except steps 9-11. In step 9, average ranking of the relative closeness scores  $RC$  of the algorithms are generated using the Microsoft Excel 2010 [119] built-in function  $\text{RANK.AVG}()$  with its generic form  $\text{RANK.AVG}(\text{number}, \text{ref}, [\text{order}])$ . In

step 10, the selectTopK() function is used to select top-k ranked algorithms while in step 11, the users build his/her model using the selected algorithms and deploy in their applications.

#### **4.3.2.1. Selecting Suitable Quality Meta-metrics**

To select an optimal performance algorithm, a machine learning (ML) user/expert must be aware of the physical meaning of the evaluation metrics. For understanding physical meaning of the evaluation metrics, we propose the idea to first abstract the evaluation metrics in the form of classifiers quality meta-metrics and then let the users know to select quality metrics compliant to their goal and objectives. This will help the users in identification of appropriate metrics and figuring out the conflicting (fuzzy) metrics, for example comprehensibility against correctness (accuracy) [120] and complexity [121]. The conflicting criteria are interdependent among each other and need special treatment during evaluation. The independent (crisp) criteria are simple to evaluate and result in unbiased decisions.

##### **a. Classifiers quality meta-metrics classification model**

Classifiers can be evaluated using a number of commonly used evaluation criteria, such as RMSE, predictive accuracy and ROC curves [16]. A general problem with users and domain experts is that they do not know physical meaning of the evaluation metrics. This creates difficulty for them to select suitable metric(s) for their evaluation. To resolve this problem, we define physical meaning of the classifiers evaluation metrics in terms of quality meta-metrics (QMM). We defined eight families of QMM for those evaluation metrics which are implemented in Weka library [122]. These include: responsiveness or computational efficiency, separability or coherency, robustness or sensitivity, consistency, correctness, complexity or simplicity, reliability and comprehensibility or interestingness or interpretability. The definitions of these qualities along with their evidences are given below.

- *Correctness.* It can be either measured directly from the correct cases or indirectly from the number of errors made. We categorize it into two sub-

groups of accuracy ('+'cor) and accuracy ('-'cor). This family contains metrics for binary class problems, multi-class problems and balanced and imbalanced data problems.

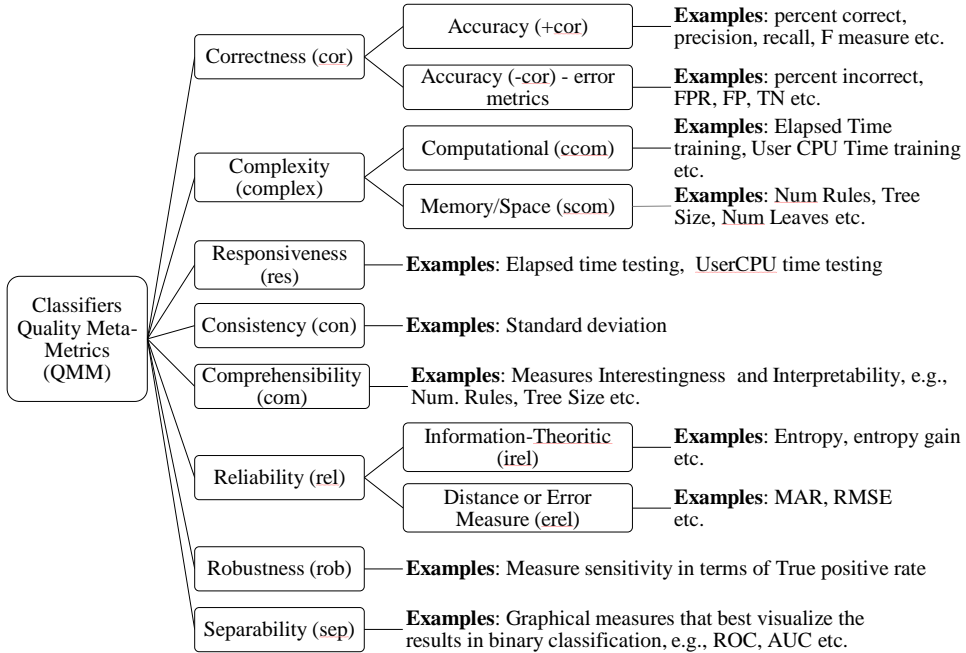
- *Complexity*. It can be measured either in terms of time spent in building the model, i.e., computational complexity (ccom) or the memory space consumed to complete the process of building and accommodating the model, i.e., memory/space complexity (scom).
- *Responsiveness*. It measures the computational efficiency of a classifier in terms of testing or execution time. We abbreviated it as *res* that stands for responsiveness of the model.
- *Consistency*. Consistency of a classifier, with respect to an evaluation metric, can be measured in terms of its standard deviation. If the classifier maintain a certain level of performance for a subsets of the main dataset then it will be consistent otherwise inconsistent one. For example, standard deviation of the accuracy measure of a classifier over the 10-fold of a test dataset measures its consistency in terms of predictive accuracy [2]. We abbreviated it as *con* in this study.
- *Comprehensibility, interestingness and interpretability*. It is combination of related subjective metrics that describes the nature of classifiers from the user's understanding and interpretation perspective. It measures the user oriented aspects, such as how well the classifier's output and the process of decision making be understood [2]. These metrics are favored in the knowledge acquisition scenario where understandability matters [120]. Comprehensibility may also results in model complexity. A complex model is intuitively more difficult to understand and interpret as compared to a simple model [121]. Similarly, for a recommender system, the interpretability criterion has great importance, where user needs to understand and verify the results of a trained model. This quality metric is abbreviated as *com*.
- *Reliability*. This family of metrics measures how much the user can trust on the quality of correctness of the performance results of a classifier. It

can be measured using error metrics, which are based on the probabilistic understanding of the errors that measures the deviation from the true probability, such as mean absolute error, mean squared error, LogLoss (cross-entropy), etc.[123]. Similarly, information-theoretic metrics, also estimate the reliability aspects of classifiers [124]. We abbreviated it as *rel* and categorized into distance or error measure (*erel*) and information-theoretic measure (*irel*).

- *Robustness*. It is a subjective measure used in diverse situations, such as ability of the classifier to make correct predictions on noisy dataset or a dataset with missing values [125] or have high sensitivity or true positive rate [3]. Sophisticated AUC measures have been reported recently for improving the quality of robustness of classifiers [126]. We abbreviated it as *rob* in our study.
- *Separability and coherency*. In the context of binary classification problems, area under the receiver operating curve (AUC) is closely related to the concept of separability [123]. AUC can best distinguish the positive and negatives classes of a dataset. We abbreviate it as *sep* in our study.

A partly similar concept of classifiers qualities can be found from [127] and [3] with limited scope and number of qualities defined. We have proposed and defined a classification model for these qualities, as shown in Figure 4.3.





**Figure 4.3.** Classification model of the classifiers quality meta-metrics

While selecting qualities from the QMM classification model, to evaluate classifiers, intensive care should be taken to select only those qualities which satisfy the properties of legibility (containing sufficiently small number of criteria), operational, exhaustiveness (containing all points of view), monotonicity and non-redundancy (each criterion should be counted only once). These properties were initially defined in article [128]. A mathematical representation of the proposed QMM is shown in equation 1.

$$\text{QMM} = \{\text{cor}, \text{complex}, \text{res}, \text{con}, \text{com}, \text{rel}, \text{rob}, \text{sep}\} \quad (1)$$

Based on QMM classification model, the list of Weka classifiers' evaluation metrics are categorized, as shown in Table 4.1.

**Table 4.1.** Categorization of classifiers evaluation metrics based on quality meta-metrics

Id	Evaluation Metric	QMM	Sub-QMM	Id	Metric	QMM	Sub-QMM
1	Number_correct	cor	+cor	27	Elapsed_Time_training	complex	ccom
2	Percent_correct	cor	+cor	28	UserCPU_Time_training	complex	ccom
3	Kappa_statistic	cor	+cor	29	measureNumRules	complex, com	scom
4	True_positive_rate	cor	+cor	30	measurePercentAttsUsedByDT	complex, com	scom
5	Num_true_positives	cor	+cor	31	measureTreeSize	complex, com	scom
6	False_negative_rate	cor	+cor	32	measureNumLeaves	complex, com	scom
7	Num_false_negatives	cor	+cor	33	measureNumPredictionLeaves	complex, com	scom
8	IR_precision	cor	+cor	34	measureNodesExpanded	complex, com	scom
9	IR_recall	cor	+cor	35	Elapsed_Time_testing	res	ures
10	F_measure	cor	+cor	36	UserCPU_Time_testing	res	sres
11	Weighted_avg_true_posi ve_rate	cor	+cor	37	SF_prior_entropy	rel	irel
12	Weighted_avg_false_negat ive_rate	cor	+cor	38	SF_scheme_entropy	rel	irel
13	Weighted_avg_IR_precisi on	cor	+cor	39	SF_entropy_gain	rel	irel
14	Weighted_avg_IR_recall	cor	+cor	40	SF_mean_prior_entropy	rel	irel
15	Weighted_avg_F_measure	cor	+cor	41	SF_mean_scheme_entropy	rel	irel
16	Number_incorrect	cor	-cor	42	SF_mean_entropy_gain	rel	irel
17	Number_unclassified	cor	-cor	43	KB_information	rel	irel
18	Percent_incorrect	cor	-cor	44	KB_mean_information	rel	irel
19	Percent_unclassified	cor	-cor	45	KB_relative_information	rel	irel
20	False_positive_rate	cor	-cor	46	Mean_absolute_error	rel	erel
21	Num_false_positives	cor	-cor	47	Root_mean_squared_error	rel	erel
22	True_negative_rate	cor	-cor	48	Relative_absolute_error	rel	erel
23	Num_true_negatives	cor	-cor	49	Root_relative_squared_error	rel	erel
24	Weighted_avg_false_posi ve_rate	cor	-cor	50	Area_under_ROC	sep, cor	-'
25	Weighted_avg_true_negati ve_rate	cor	-cor	51	Weighted_avg_area_under_RO C	sep, cor	-'
26	True_positive_rate	cor, rob	+cor	--	--	--	--

### b. Selecting suitable quality meta-metrics

In this section, we proposed a formal expert group-based quality meta-metrics selection method, where a group of experts participate in a closed discussion and rate the quality metrics. We are motivated to the experts' group-based decision making method due to the effectiveness of nominal group technique (NGT) [129]

that quantifies the experts' preferences in the form of quantitative score. The proposed experts' group-based QMM selection process is represented in procedure 1.

---

**Procedure 1.** selectSuitQuality
 

---

**Begin**
**inputs:** QMM – classifiers quality meta-metrics

G – goal

**output:** Q'' – highly rated/ranked quality meta-metrics

- 1 [Select key qualities by each expert]  
 Q = extractSalientQMM(QMM, G); //where,  $Q \subseteq QMM$
- 2 [Vote each quality by each expert]  
 Q' = preliminaryVoteAggQuality(Q'); //where, Q' is the initial list of selected QMM
  - a. **If** Q' contains *Consistent* qualities, then
    - i. Q'' = selectTopKQMM(Q', k); // where, k represents the number of qualities experts are interested in
    - ii. **goto** setp 3;
  - b. **Else**
    - i. **repeat** step 2;
- 3 **return** Q'';

**End**


---

In procedure 1, step 1, experts' panel uses extractSalientQMM() to extract those quality metrics from QMM classification model, which are essential for the evaluation of classifiers under the defined goal G. The salient qualities are collected by the head expert and presented for discussion, if needed, otherwise, preliminaryVoteAggQuality() is used (step 2,) to vote salient qualities by each expert. For voting salient qualities, rating or ranking methods can be used. The output of this function is to select top-k qualities, if they are consistent. A quality is said to be consistent if all the experts have uniformly rated/ranked it. For example, if  $\frac{3}{4}$  of the experts rate correctness as rank 1 and only one expert rates it negatively, then it may be due to the inconsistent rating by the experts. In this case, re-voting is done and the process is continued till consensus are made. The final output of procedure 1 is the list of most desirable qualities for the defined goal.

#### 4.3.2.2. Selecting suitable evaluation-metrics

Once suitable qualities,  $\bar{Q}$ , are selected, the next step is to select suitable evaluation metrics. However, in case of classification algorithms, for each  $\bar{Q}$ , a large number of metrics are available (a few are shown in Table 4.1).

The selection of suitable metrics (i.e., metrics to integrate) depends on the scope of the classifiers under analysis, which is defined in terms of the number of families of classifiers taken under consideration. A few of the commonly used families of classifiers, i.e., probabilistic family, lazy learners' family, function family, rule family, decision tree family and meta-learners family, are implemented in Weka [122], which are focused in this study. Apart from the scope of the classifiers, the domain/application requirements also influence the selection of suitable metrics. To resolve the metrics selection problem, we adopt the idea of experts group-based decision making, motivated by the NGT [129]. The methodology used is algorithmically represented in procedure 2.

---

##### Procedure 2. selectSuitEvalMetrics

---

**Begin**

**inputs:**  $\bar{Q}$  –highly rated/ranked quality meta-metrics

$G$  – goal

**output:**  $SM''$  –highly rated/ranked evaluation metrics

**Let specAlgEvlMetrics** = Specification of evaluation metrics. See Table 4.1.

- 1 [Select salient evaluation metrics (SM) from each quality metric]  
 $SM = \text{extractSalientMetrics}(\bar{Q}, G, \text{specAlgEvlMetrics});$
- 2 [Vote each evaluation metric by each expert]  
 $SM' = \text{preliminaryVoteAggMetrics}();$  //where,  $SM'$  is initial list of selected metrics
  - a. If  $SM'$  contains *Independent* metrics, then
    - i.  $SM'' = \text{selectTopKSuitMetrics}(SM', k);$  //where,  $SM'' \subset SM'$  and  $k$  is the number of metrics
    - ii. **goto** setp 3;
  - b. Else
    - i. **repeat** step 2;
- 3 **return**  $SM'';$

**End**

---

In procedure 2, step 1, experts' panel uses  $\text{extractSalientMetrics}()$  to extracts those quality metrics from  $\bar{Q}$ , which qualify the goal  $G$ . The salient evaluation

metrics from each quality are extracted by utilizing `specAlgEvlMetrics` (see Table 4.1). This process is completed in step 2 by using `preliminaryVoteAggMetrics()`. For voting the same method as described in previous section is used. The output of this function is to select top-k metrics, if they are crisp/independent. An evaluation metric is said to be independent if it is not duplicate with other metrics. For example, percent accuracy and percent incorrect/errors are interdependent evaluation metrics and both should not be included in the evaluation metrics. The final output of this procedure is the list of selected suitable evaluation metrics  $SM''$ , which are the main ingredients of the generic multi-metric criteria. Our focus is to select metrics that have the following features: (a) easily computable, (b) perform best on all types of datasets, (c) coherent with the final decision, (d) non-conflicting/independent of each other, (e) same representation with same scale, (f) quantifiable/measurable and (g) related with the algorithms evaluation. While selecting metrics, preference should be given to those metrics that qualify maximum of these qualities [130].

#### 4.3.2.3. Consistent relative criteria weighting

The selected evaluation metrics are the final ingredients of the evaluation criteria that play their corresponding roles in achieving the final goal. The roles define the preference or priority or weight of the metrics, which should be first estimated and then used during evaluation. State-of-the-art algorithm evaluation and recommendation studies, discussed in literature, follow absolute or partial relative weighting techniques that support limited number of criteria. The weights are assigned by experts, utilizing their own knowledge of the domain. In order to resolve shortcomings of the existing work, we proposed the idea of group decision making for consistent relative weights of the criteria. For this task, we are motivated by the AHP weighting method [131], which has the ability to quantify experts' preferences in the form of weight scores, using the pairwise-wise comparisons procedure utilizing Saaty's preference scale (SPS) [132], shown in Table 4.2.

**Table 4.2.** Saaty's preference scale for pair-wise comparison of evaluation criteria

Definition	Intensity of importance	Definition	Intensity of importance
<i>Equally important</i>	1	<i>Equally important</i>	1/1
<i>Equally or slightly more important</i>	2	<i>Equally or slightly less important</i>	1/2
<i>Slightly more important</i>	3	<i>Slightly less important</i>	1/3
<i>Slightly to much more important</i>	4	<i>Slightly to way less important</i>	1/4
<i>Much more important</i>	5	<i>Way less important</i>	1/5
<i>Much to far more important</i>	6	<i>Way to far less important</i>	1/6
<i>Far more important</i>	7	<i>Far less important</i>	1/7
<i>Far more important to extremely more important</i>	8	<i>Far less important to extremely less important</i>	1/8
<i>Extremely more important</i>	9	<i>Extremely less important</i>	1/9

According to the interpretation of this scale, if an evaluation metric  $e_1$  is extremely more important than evaluation metric  $e_2$ , it is rated as 9 and then  $e_2$  must be extremely less important than  $e_1$ , which is rated as 1/9. Table 4.2 has all the possible values of importance of evaluation criteria and its inverse along with their interpretations.

For weighting the evaluation criteria, the AHP expert group-based prioritization mechanism is followed in the sequence: prioritizing experts, creating a pairwise comparison matrix of the selected metrics ( $\bar{Q}$ ), assigning experts' relative priority weights, evaluating consistency of the individual weights and aggregating individual's weights into group weights. The process is described in procedure 3.

In step 1 of the procedure 3, an  $n \times n$  comparison matrix (DMM) is designed to estimate the decision power of each decision maker. These weights are assigned using function `estimatedDMWgt()` (step 2). The weights are estimated using the AHP pairwise comparison procedure. Each entry  $dm_{ij}$  of the matrix DMM is entered by the head expert, on the basis of his/her understanding about the expertise of other experts (DM). Each of these values represents the superiority of  $i^{\text{th}}$  DM relative to the  $j^{\text{th}}$  DM. If  $dm_{ij} > 1$ , then the  $i^{\text{th}}$  DM is more influential in decision making than the  $j^{\text{th}}$  DM, but if  $dm_{ij} < 1$ , then the  $i^{\text{th}}$  DM is less influential than

the  $j^{\text{th}}$  DM. However, if  $d_{mij} = 1$  both  $i^{\text{th}}$  and  $j^{\text{th}}$  DM have the same level of importance in the decision.

---

**Procedure 3.** `estimatRelativeWeight`

---

**Begin**

**inputs:**  $\bar{Q} = \{e_1, e_2, \dots, e_m\}$ ; // selected evaluation metrics

**output:**  $W$  – weights vector **Let**  $DM = \{dm_1, dm_2, \dots, dm_n\}$ ; // Group of experts

**SPS** = Saaty's preference scale (see Table 4.2)

**GDMM** =  $m \times n$  'group decision making matrix', where  $m$  represents metrics and  $n$  represents decision makers

- 1 [Design comparison matrix for decision makers]  
 $DMM = dm_{ij}$ ; //where,  $DMM$  is  $n \times n$  comparison matrix of decision makers with  $dm_{ij}$  is the decision weight of the  $i^{\text{th}}$  decision maker relative to the  $j^{\text{th}}$  decision maker
- 2 [Estimate decision makers decisions weight]
  - a.  $DMWeight = \text{estimateDMWgt}(SPS, DMM)$ ; //where,  $DMWeight$  is a single column weights vector containing preferences of decision makers. // See equations 2 and 3
  - b. Check *consistency* of  $DMWeight$ ; // See equations 4-7
- 3 [Estimate metrics weights]
 

**for**  $dm = 1$  to  $n$  do

  - a. [Design comparison matrix for evaluation metrics]  
 $EM = e_{ij}$ ; //where,  $EM$  is  $m \times m$  comparison matrix of the evaluation metrics with  $e_{ij}$  is the preference of  $i^{\text{th}}$  metric against the  $j^{\text{th}}$  metric
  - b.  $EMWeight_{dm} = \text{estimateEvalMetricsWgt}(SPS, EM)$ ; //where,  $EMWeight$  is single column weights vector for metrics  $\bar{Q}$ . // See equations 2 and 3
  - c. Check *consistency* of  $EMWeight_{dm}$ ; // See equations 4-7
  - d. Insert  $< EMWeight_{dm} >$  into  $GDMM$ ;

**End for**
- 4 [Aggregate weights of all decision makers using group decision making]
 

**foreach**  $e \in GDMM$

 $W = \sum (\prod_{dm=1}^n (DMWeight^T, EMWeight))$ ; //  $W$  is aggregate weights vector

**End for**
- 5 **return**  $W$ ;

**End**

---

For estimating the DM decision weights,  $DMM = dm_{ij}$  is first transformed to the normalized matrix,  $\overline{DMM} = \overline{dm}_{ij}$ , where each entry  $\overline{dm}_{ij}$  is computed using equation 2 and then a column weight vector  $W = w_j$  is produced using equation 3,

$$\overline{dm}_{ij} = dm_{ij} / \sum_{i=1}^n dm_{ij} \quad (2)$$

(51)

$$w_j = \sum_{i=1}^n \overline{dm}_{ij} / n = \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{pmatrix}, \text{ where } i \& j = 1, 2, \dots, n. \quad (3)$$

To verify correctness of the experts' judgment and preferences about the relative weights assigned to each DM, consistency is checked using the eigenvector method [17], which computes consistency ratio (CR) using equation 4-7,

$$CR = CI/RI, \quad (4)$$

where, RI is the random consistency index value from the random consistency table [132], shown in Table 4.3. Similarly, the value of CI measures the deviation which is computed using equation 5,

**Table 4.3.** Random consistency indices (RI) for different number of evaluation criteria (n).

Number of evaluation criteria (n)	1	2	3	4	5	6	7	8	9	10	11
Random consistency index (RI)	0.00	0.00	0.58	0.9	1.12	1.24	1.32	1.41	1.45	1.49	1.51

$$CI = (\lambda_{\max} - n) / (n - 1), \quad (5)$$

where,  $\lambda_{\max}$  is the principal eigenvalue, such as  $\lambda_{\max} \in \Re > n$ . The eigenvalue is computed by averaging values of the consistency vector Cv, as shown in equation 6,

$$\lambda_{\max} = (\sum_{i=1}^n Cv_{ij}) / n, \quad (6)$$

where, each value  $Cv_{ij}$  of the consistency vector, is computed by taking product of the pairwise comparison matrix DMM with the weight vector W. This relationship is shown in equation 7,



$$Cv_{ij} = E * W. \quad (7)$$

In step 3 (a),  $m$  comparison matrices (i.e., EM) are created, one for each decision maker to relatively weight all the evaluation metrics  $\bar{Q}$ . In step 3 (b), each decision maker (dm) uses the function `estimateEvalMetricsWgt()` to assign and estimate the weight for each evaluation metric. In step 3 (c) the consistency of metrics weights are checked using equations 4-7. In step 3 (d), the weight vector `EMWeight` is added to the group decision making matrix, `GDMM`. In step 4, the weights estimated for the evaluation metrics  $\bar{Q}$  by the  $n$  decision makers, `DM`, are aggregated using the group decision making process, which are return to the main algorithm 1 using step 5.

#### 4.3.2.4. Measuring algorithms performance

In this phase, first the candidate list of algorithms are selected from the pool of freely available classification algorithms. We selected 35 multiclass classification algorithms, from six heterogeneous families of the classifiers, implemented in Weka [122]. The list of these algorithms is shown in Table 4.4.

**Table 4.4.** List of Weka well-known multi-class classifiers

SNo	Abbreviation	Classifier	SNo	Abbreviation	Classifier
1	A1	bayes.BayesNet	19	A19	trees.J48
2	A2	bayes.NaiveBayes	20	A20	trees.J48graft
3	A3	bayes.NaiveBayesUpdateable	21	A21	trees.LADTree
4	A4	functions.Logistic	22	A22	trees.RandomForest
5	A5	functions.RBFNetwork	23	A23	trees.RandomTree
6	A6	functions.SMO	24	A24	trees.REPTree
7	A7	misc.HyperPipes	25	A25	trees.SimpleCart
8	A8	misc.VFI	26	A26	meta.AdaBoostM1
9	A9	rules.ConjunctiveRule	27	A27	meta.Bagging
10	A10	rules.DecisionTable	28	A28	meta.Dagging
11	A11	rules.DTNB	29	A29	meta.END
12	A12	rules.JRip	30	A30	meta.FilteredClassifier
13	A13	rules.OneR	31	A31	meta.LogitBoost
14	A24	rules.PART	32	A32	meta.RacedIncrementalLogitBoost
15	A15	rules.Ridor	33	A33	meta.RandomSubSpace
16	A26	rules.ZeroR	34	A34	meta.Stacking
17	A17	trees.BFTree	35	A35	meta.Vote
18	A18	trees.FT	---	--	---

To rank these algorithms,  $A$ , on a classification dataset,  $d$ , using the performance results of evaluation metrics  $\bar{Q}$ , all the algorithms ( $A$ ) are executed sequentially on

(53)

dataset  $d$  in Weka environment and the results are stored into the performance matrix  $P$  for later use.

#### 4.3.2.5. Testing significance of performance results

Unlike the traditional ranking methods that directly select top-rank algorithm (without considering significance tests of the results) for learning models, we propose the idea of checking the performance results for statistical significance. According to this idea, the performance results of the candidates algorithms  $A$  are first tested for statistical significance and then the for the significance fitness. The objective of significance test is to identify which algorithms perform significantly better, which perform significantly poor and which perform similar with respect to a reference algorithm. For this purpose, we adopted corrected paired t-test with significance of 0.05 [102] implemented in Weka [122], which checks the significance of the algorithms results and labels them either ‘v’ (for better performance), or ‘\*’ (for worst performance) or ‘’ for equal significance performance with respect to a baseline algorithm. In our case, the definition of the reference algorithm  $a \in A$  is the algorithm which performances best as compared to all the algorithms. The selection of the reference for each metric  $e \in \bar{Q}$  is done within its local scope rather than the global scope of all metrics  $\bar{Q}$ .

For a performance matrix  $P = p_{ij}$ , with  $p_{ij}$  as the performance value of  $i^{\text{th}}$  algorithm on the  $j^{\text{th}}$  evaluation metric, the process of corrected paired t-test and the production of final labelled performance matrix  $\bar{P} = \bar{p}_{ij}$  is described in procedure 5.

In procedure 5, the criteria for selecting reference algorithm is the maximum value for a benefit metric and minimum value for a cost metric, respectively. Benefit metric are those whose higher values are preferred, e.g., accuracy, while cost metrics are those whose lower value is preferred, e.g., training time. For labeling the algorithms as either significant, or poor or equal in performance, step 1(c) is used. For this purpose, Weka corrected paired t-test is used, which takes reference algorithm (referenceAlg), single evaluation metric ( $e$ ) and the performance matrix

(P) together as inputs and returns a labelled matrix ( $\bar{p} = \bar{p}_{ij}$ ) as output. Each value  $\bar{p}_{ij}$  of the labelled matrix is either labelled as (v) or, (\*) or (').

---

**Procedure 5.** performStatSigTest
 

---

**Begin**

**inputs:**  $P$  – performance matrix

**output:**  $\bar{P}$  –  $n \times m$  performance matrix, where  $n$  is the number of algorithms and  $m$  is the number of evaluation metrics;

**Let**  $d$  – given dataset

$A = \{a_1, a_2, \dots, a_n\}$  – set of classification algorithms

$\bar{Q} = \{e_1, e_2, \dots, e_m\}$  – set of evaluation metrics

```

1   foreach  $e \in \bar{Q}$  in performance matrix  $P$ 
    a. if  $e \in$  benefit metric
        i. referenceAlg = selectReferenceAlg(maxPerformValue( $e$ ));
    b. else
        i. referenceAlg = selectReferenceAlg(minPerformValue( $e$ ));
    c.  $\bar{P} =$  performCorrectedPairedtTest(referenceAlg,  $P$ ,  $e$ );
2   end for
3   return  $\bar{P}$ 

```

**End**

---

#### 4.3.2.6. Algorithmic fitness evaluation

In this step, the algorithms' fitness levels are evaluated for consideration in the next step of evaluation. The motivation for including the fitness evaluation as an additional step is to reduce the algorithm space by filtering out the algorithms that poorly perform on all evaluation metrics on a single dataset. This is reasonable and makes sense that not to allow poor performance algorithms to the next stage of evaluation. Furthermore, it reduces the chance of selection of bad algorithm.

To implement this idea, we proposed a fitness function that evaluates labels in the labeled performance matrix  $\bar{P} = \bar{p}_{ij}$ . This function can be defined as follows. Let  $\bar{Q} = \{e_1, e_2, \dots, e_m\}$  be the set of  $m$  evaluation metrics for evaluating performance of an algorithm  $a \in A$  on a classification dataset  $d$  and  $\bar{P} = \bar{p}_{ij}$  be the labeled performance matrix, obtained after significance test. The target significant matrix  $S$ , containing the list of significantly fit algorithms, can be generated using the fitness function,

$$S = \{\forall_{a \in A}: a \in \bar{P} | \forall_e: e \in \bar{Q}. \sim \text{nonSignificant}(e)\}, \quad (8)$$

where,  $\text{nonSignificant}(e)$  is the function that determines the significance level of each  $a \in A$  for each evaluation metric  $e \in \bar{Q}$  and returns true if it either performs significantly better or equal and add to the significant matrix  $S$ . The process is repeated for all algorithms  $A$  against all metrics  $\bar{Q}$  and the final results are accumulated in  $S$ , which is the reduced version of the original labelled matrix  $\bar{P}$ , in terms of number of candidate algorithms i.e.,  $\text{SizeOf}(S) < \text{SizeOf}(\bar{P})$ . Internally, the function  $\text{nonSignificant}(e)$  processes the labels, i.e., ‘v’, ‘\*’ and ‘,’ of the values of each metric  $e \in \bar{Q}$ , assigned by the corrected paired t-test of the procedure 4. In the significant matrix  $S$ , each value is represented by  $s_{ij}$ , where  $i$  represents the algorithm and  $j$  represents the evaluation metric.

#### 4.3.2.7. Ranking algorithms

State-of-the-art methods for ranking algorithms are based on the aggregate score of multiple evaluation metrics  $\bar{Q}$ , combined together in different ways, consuming absolute weights, which are assigned by domain experts and take appropriate normalization mechanism for the values of the criteria. These methods have minimal support for extension in terms of number of metrics to be added and lack support for implicit and explicit constraints satisfaction. Our idea is to evaluate the candidate algorithms and rank them according to their relative closeness score to the ideal algorithm with the consumption of relative consistent weights and different constraints. To achieve these objectives, we are motivated by the flexibility and ranking power of the TOPSIS multi-criteria decision making method [44, 133]. The TOPSIS steps used during algorithms ranking are shown in procedure 6.

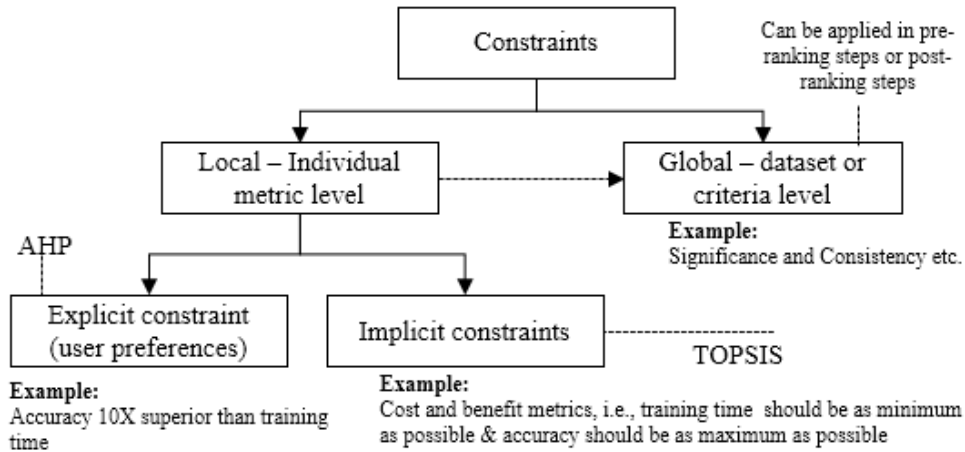
**Procedure 6.** rankAlgorithms**Begin****inputs:**  $S$  –  $n \times m$  matrix containing performance results of significant algorithms $W$  –  $1 \times m$  (single row) weight vector**output:**  $RC$  –  $n \times 1$  (single column) matrix of the relative closeness score**Let**  $d$  –dataset $A = \{a_1, a_2, \dots, a_n\}$  – set of classification algorithms $\bar{Q} = \{e_1, e_2, \dots, e_m\}$  – set of evaluation metrics**1** [create performance evaluation matrix from  $S$ ] $S = (s_{ij})_{n \times m}$ ; //where,  $s_{ij}$  represents value of algorithm  $i$  for evaluation metric  $j$ **2** Define local/implicit constraints on  $\bar{Q}$ ;**3** [normalize performance evaluation matrix  $S$ ] $\bar{S} = r_{ij} = s_{ij} / \sqrt{\sum_{i=1}^n s_{ij}^2}$ ; //where,  $i = 1, 2, \dots, n$  and  $j = 1, 2, \dots, m$ **4** [compute weighted normalized decision matrix  $V$ ] $V = (v_{ij})_{n \times m} = r_{ij} * W_j$ ; //where,  $W_j$  is the weight vector**5** [compute positive ideal solution (PIS) and negative ideal (NIS) solution]a.  $PIS = \left\{ \left( \max_i(v_{ij}) \mid j \in C_b \right), \left( \min_i(v_{ij}) \mid j \in C_c \right) \right\} = \{v_j^* \mid j = 1, 2, \dots, m\}$ b.  $NIS = \left\{ \left( \min_i(v_{ij}) \mid j \in C_b \right), \left( \max_i(v_{ij}) \mid j \in C_c \right) \right\} = \{v_j^- \mid j = 1, 2, \dots, m\}$ **6** [compute separation measures using  $m$ -dimensional Euclidean distance]a.  $PIS_i^+ = \sqrt{\sum_{j=1}^m (v_{ij} - v_j^*)^2}$ ,  $j = 1, 2, \dots, m$ b.  $NIS_i^- = \sqrt{\sum_{j=1}^m (v_{ij} - v_j^-)^2}$ ,  $j = 1, 2, \dots, m$ **7** [compute relative closeness (RC) of algorithms with respect to ideal algorithm] $RC = \frac{NIS_i^-}{PIS_i^+ + NIS_i^-}$ ,  $i = 1, 2, \dots, n$ ; where,  $RC$  is a  $n \times 1$  matrix**8** **return**  $RC$ ;**End**

The value  $RC$  lies between 0 and 1, i.e.,  $0 \leq RC \leq 1$ . If  $RC=1$ , the TOPSIS has the best condition of the top- $k$  algorithms selection; and if  $RC=0$ , the TOPSIS has the worst condition of algorithm selection. Any other value in-between these two values measures the appropriateness level of that algorithm.

**4.3.2.8. Constraints satisfaction**

The constraints used in our study can be categorized into individual level, limited to a single metric of the evaluation criteria, and global level, applicable to all the metrics in the evaluation criteria. Individual level constraints are satisfied in the pre-ranking and ranking steps of evaluation process. These are further categorized into explicit and implicit constraints. The explicit constraints are defined by the

users or experts, such as users' relative preferences on the metrics. An example can be, "the accuracy metric should be favored 10-times as compared to training time". The implicit constraints are the inherent characteristics of individual metrics, such as the value of cost criteria which should be as minimum as possible and the values of benefit criteria should be as higher as possible. Contrary to the local constraints, global constraints are the explicit constraints that are based on the local constraints and applicable to the overall criteria in the pre-ranking evaluation process. Examples of the global constraints are the consistency of estimated weights of the criteria, significance of the performance results of the algorithms and consistency in the performance results of the algorithms. Figure 4.4 shows different types of constraints with their examples that are applied at different levels of the algorithms evaluation and ranking process.



**Figure 4.4.** Categorization of constraints defined over evaluation criteria

In this study, for satisfaction of the local constraints, we proposed the idea of relative weighting using AHP process, and the idea of cost and benefits analysis of the metrics using the TOPSIS method. Similarly, for the satisfaction of global constraints, we adopted the AHP weights consistency check methods using eigenvector computation, and proposed the idea of paired t-test embedded in the algorithmic fitness evaluation function for checking the significance of the algorithms performance results. The local constraints can be satisfied through the configuration of AHP and TOPSIS methods, but the global constraints' satisfaction

need more advanced level user-defined functions. We measure the consistency of algorithms in terms of standard deviation of their results. The algorithm that has lowest standard deviation value is the consistent algorithm and vice versa.

#### 4.4. Validation of the AMD methodology - a scenario

In order to evaluate the effectiveness of AMD methodology, verify its potential use in real-world scenarios and allow other researchers to confirm our results, we perform step-by-step process in this section with the necessary experiments. First consider a scenario in which a user is interested in learning his dataset with a classification algorithm, which he does not really know. The key problem he faces is the selection of an optimum performance classification algorithm that fits well into his requirements and expectations, expressed in terms of goal and associated objectives. In this scenario, the user is given a choice to select the best algorithm from a list of most commonly used 35 multi-class classification algorithms, shown in Table 4.4 for the 15 classification datasets<sup>1</sup>, shown in Table 4.5. Due to the space issue, the AMD steps are described only for one dataset.

**Table 4.5.** General characteristics of UCI/OpenML repositories datasets

Datasets	Characteristics of Datasets						
	Attributes	Nominal Attributes	Numeric Attributes	Binary Attributes	Classes	Instance Count	Missing
abalone-3class	9	1	7	0	3	4177	0
rae-148	9	1	7	0	3	4177	0
acute-inflammations-neph	6	0	5	0	2	66	0
ADA_Agnostic	7	5	1	5	2	120	0
ADA_Prior	49	0	48	0	2	4562	0
adult-4000	15	8	6	1	2	4562	88
adult-8000	15	8	6	1	2	3983	0
aileron	15	8	6	1	2	8000	0
anacatdata-AIDS	41	0	40	0	2	5795	0
anacatdata-apnea2	5	2	2	0	2	50	0
anacatdata-apnea2	4	2	1	0	2	475	0
anacatdata-asbestos	4	2	1	0	2	475	0

<sup>1</sup> Some of the datasets are used with minor modifications by changing the type of the class label to nominal etc.

Datasets	Characteristics of Datasets						Missing
	Attributes	Nominal Attributes	Numeric Attributes	Binary Attributes	Classes	Instance Count	
analcatauthorship	4	2	1	1	2	83	0
analcatabankruptcy	71	0	70	0	4	841	0
analcatabirthday	7	1	5	0	2	50	0

A machine learning practitioner can use the proposed AMD methodology as follows.

### Step 1: Goal and objectives definition

The goal of the study is to select an optimum performance multiclass classification algorithm from the heterogeneous families of algorithms (see Table 4.4) for binary and multiclass problems (see Table 4.5) that has optimum performance.

### Step 2: Selecting suitable quality meta-metrics

For the goal in step 1, procedure 1 is used to select the suitable quality metrics. Four machine learning experts, i.e., machine learning and data mining expert (DM#1), a data and knowledge engineering expert (DM#2), a scientist, researcher and developer (DM#3) and an expert user of the classification algorithms in diverse application area (DM#4) were chosen to select the qualities. Using procedure 1, the experts selected correctness (accuracy), responsiveness, computational complexity and consistency (as shown in Table 4.6) as the relevant qualities that are compliant to the goal and satisfy the heterogeneity constraint of the classifiers.

**Table 4.6.** Experts' group-based rating of quality metrics for heterogeneous classifiers

Quality Metrics	DM#1	DM#2	DM#3	DM#4	Total
Correctness (cor)	60	50	55	70	235
Computational Complexity (ccom)	5	20	15	-	40
Responsiveness (res)	15	-	20	20	55
Consistency (con)	10	15	-	-	25
Comprehensibility (com)	-	15	-	7	23
Reliability (rel)	5	-	-	-	5
Robustness (rob)	-	-	10	3	13
Separability (sep)	5	-	-	-	5
Total	100	100	100	100	400

\*[Each expert distributes 100 points across the qualities metrics]



Table 4.6 shows the importance score of each quality metrics. The top 4 qualities are non-conflicting and reflect the general characteristics of all the classifiers, therefore they are selected. These qualities are represented in equation 9,

$$\bar{Q} = \{\text{cor}, \text{ccom}, \text{res}, \text{con}\}. \quad (9)$$

The physical meaning of equation 2, is that the optimum performance algorithm is the one that has high level of correctness in its results, low computational complexity, quick response time to users' requests, and high consistency in its results for a test dataset.

### Step 3: Selecting suitable evaluation metrics

Procedure 2 is used to assist expert in the selection of suitable evaluation metrics, shown by equation 10 and Table 4.7, respectively,

$$\bar{\bar{Q}} = \{\text{Wgt. Avg. F - score, CPUTimeTraining, CPUTimeTesting, Consistency}\}. \quad (10)$$

**Table 4.7.** Evaluation metrics for performance analysis of heterogeneous multi-class classifiers

Evaluation Metrics	(DM#1 - DM#5) Decision maker
Correctness (cor)	Wgt. Avg. F-score
Computational Complexity (ccom)	CPUTimeTraining
Responsiveness (res)	CPUTimeTesting
Consistency (con)	Consistency (Stdev.)

In Table 4.7, the consistency metric cannot be directly measured by any of the metric shown in Table 1. It is defined by the experts in their discussion of voting for metrics selection. It is a global explicit constraint that helps in selecting an algorithm that has consistent results.

### Step 4: Weighting Metrics

The estimation of evaluation metrics is done using procedure 3 and the results are shown in Table 4.8 and Figure 4.5. Weights of the decision power of each decision

maker is shown in Table 4.8(a). The relative weights, for each metric, estimated by each decision maker, are shown in Table 4.8(b-e). The final, experts' group-based weights are shown in Table 4.8(f).

**Table 4.8.** Analytical hierarchy process (AHP) based relative criteria weighting

(a). Experts' (decision makers') decisions' prioritization

DM/DM	DM#1	DM#2	DM#3	DM#4	DM Decision Weights
DM#1	1	3	2	5	0.49
DM#2	0.33	1	1	3	0.21
DM#3	0.50	1.00	1	3	0.23
DM#4	0.20	0.33	0.33	1	0.08
<b>CI: 0.009</b>					<b>1.00</b>

(b) DM#1 relative weighting

Criteria	F-score*	TestTime*	TrainTime*	Consistency	Weights
F-score	1	8	9	7	0.70
TestTime	0.13	1	3	1/2	0.09
TrainTime	0.11	0.33	1	1/5	0.04
Consistency	0.14	2.00	5	1	0.16
<b>CI:0.050</b>					<b>1.00</b>

(c) DM#2 relative weighting

Criteria	F-score	TestTime	TrainTime	Consistency	Weights
F-score	1	7	9	5	0.68
TestTime	0.14	1	2	1	0.12
TrainTime	0.11	0.50	1	1/3	0.06
Consistency	0.2	1.00	3	1	0.14
<b>CI:0.012</b>					<b>1.00</b>

(d) DM#3 relative weighting

Criteria	F-score	TestTime	TrainTime	Consistency	Weights
F-score	1	7	8	6	0.68
TestTime	0.14	1	2	1/2	0.10
TrainTime	0.13	0.50	1	1/3	0.06
Consistency	0.17	2.00	3.00	1	0.16
<b>CI:0.021</b>					<b>1.00</b>

(e) DM#4 relative weighting

Criteria	F-score	TestTime	TrainTime	Consistency	Weights
F-score	1	8	9	8	0.71
TestTime	0.13	1	4	1	0.12
TrainTime	0.11	0.25	1	1/6	0.04
Consistency	0.13	1.00	6	1	0.13
<b>CI:0.073</b>					<b>1.00</b>

(f) Criteria weights based on group decision making

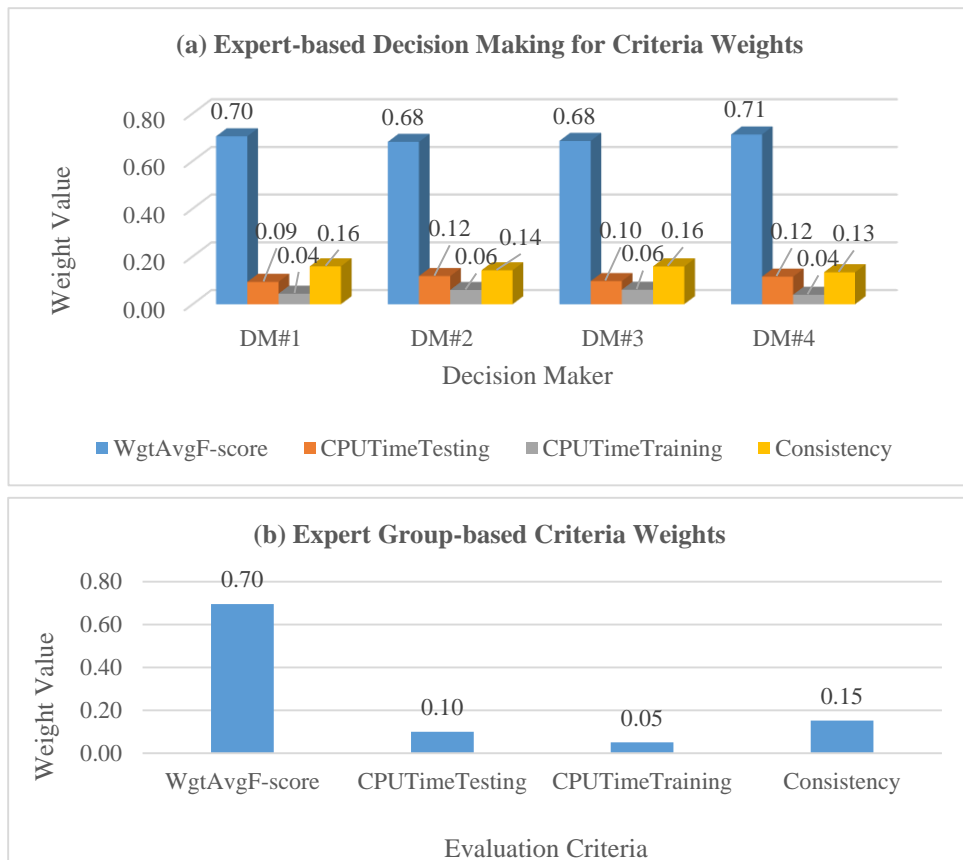
DM Decision Weights	0.49	0.21	0.23	0.08	
Criteria\DM	DM#1	DM#2	DM#3	DM#4	Weight
F-score	0.70	0.68	0.68	0.71	0.70
TestTime	0.09	0.12	0.10	0.12	0.10
TrainTime	0.04	0.06	0.06	0.04	0.05
Consistency	0.16	0.14	0.16	0.13	0.15
					<b>1.00</b>

\*F-score: WgtAvgF-score

\*TestTime: CPUTimeTesting

\*TrainTime: CPUTimeTraining

According to the weight scores of these metrics, Wgt. Avg. F-score is the most preferable, followed by consistency, followed by CPUTimeTesting followed by CPUTimeTraining.



**Figure 4.5.** Criteria relative weights, estimated using analytic hierarchy process

### Step 5: Measuring algorithms performance

For generating real performance results of the 35 classification algorithms, Weka environment is used. Table 4.10, column 2-5, shows the results for ADA\_Agnostic dataset [134]. The consistency column 5 of Table 4.10 is not directly measurable using the existing evaluation metrics, therefore we compute it by averaging standard deviations of the first three evaluation metrics, using equation 11,

$$\text{Consistency}_{a \in A} = \frac{\sum_{i=1}^m \text{Stdev}_i}{m} \quad (11)$$

where,  $a$  represents an algorithm belonging to the algorithm space  $A$  and  $m$  represents the number of measurable metrics (3 in this case). For simplicity purpose, in this chapter, we use the concept consistency instead of the average consistency. The consistency scores for a partial list of the algorithms are shown in Table 4.9 (last column).

**Table 4.9.** Partial list of average standard deviation (average consistency) of the classifiers

Algorithms	F-score* (Stdev)	TestTime* (Stdev)	TrainTime* (Stdev)	Average (Stdev) - Consistency
<b>bayes.BayesNet</b>	0.018	0.015	0.005	0.013
<b>bayes.NaiveBayes</b>	0.017	0.006	0.008	0.010
<b>bayes.NaiveBayesUpdateable</b>	0.017	0.007	0.008	0.011
<b>functions.Logistic</b>	0.015	0.019	0.002	0.012
...	...	...	...	...
<b>meta.Vote</b>	0.017	0.010	0.000	0.009

\*F-score: WgtAvgF-score

\*TestTime: CPUTimeTesting

\*TrainTime: CPUTimeTraining

### Step 6: Testing significance of performance results

For checking the statistical significance of the algorithms performance results, procedure 4 is used, whose results are shown in Table 4.10, column 2-4. In this table, the reference classifiers are marked by bold faced keyword “ref” and the statistically poor results are marked with symbol “\*”. The results, in these three

columns, with no symbol mentioned, are either same in performance or cannot be decided surely.

### Step 7: Algorithmic fitness evaluation

The fitness function is performed on the labelled significant matrix of the algorithms results, which are marked as significant, non-significant and equally significant. In our proposed fitness evaluation function, described by equation 8, the non-significant algorithms are identified and are either filter out and dropped from the next step of evaluation or leaved as they are but not considered, when final selection is made from the ranked list of algorithms. Applying the fitness function, the algorithms bayes.NaiveBayes, bayes.NaiveBayesUpdateable, and meta.Dagging are identified as significantly poor on ADA\_Agnostic dataset (see Table 4.10). The results of equation 8, for all the datasets, are summarized in Table 4.14.

### Step 8: Ranking algorithms

To generate recommended ranking, procedure 5 is applied on the performance matrix, Table 4.10, columns 2-5) with the specification of local constraints (i.e., Max and Min) and global constraints (i.e., consistency).

**Table 4.10.** Classifiers performance and ranking based on relative distance from ideal algorithm

Algorithms*	Constraints				PIS <sup>=</sup>	NIS <sup>-</sup>	RC	Ranking
	Max	Min	Min	Min				
	F-score	TestTime	TrainTime	Consistency				
A1	0.78*	0.027*	0.002	0.013	0.00906	0.03830	<b>0.80874</b>	26
A2	0.825*	0.013*	0.008*	0.010	0.00264	0.04180	<b>0.94068</b>	19
A3	0.825*	0.011*	0.01*	0.011	0.00272	0.04171	<b>0.93882</b>	20
A4	0.836	0.229*	0.000	0.012	0.00088	0.04317	<b>0.97995</b>	4
A5	0.733*	0.232*	0.004	0.043	0.01593	0.03492	<b>0.68672</b>	29
A6	0.830	1.99*	(ref) <b>0.000</b>	0.041	0.00181	0.04239	<b>0.95905</b>	12
A7	0.66*	(ref) <b>0.001</b>	0.000	0.005	0.02658	0.03309	<b>0.55457</b>	32
A8	0.716*	0.008*	0.004	0.012	0.01841	0.03433	<b>0.65097</b>	31
A9	0.645*	0.043*	0.000	0.006	0.02877	0.03301	<b>0.53432</b>	35
A10	0.829	1.086*	0.000	0.043	0.00195	0.04231	<b>0.95597</b>	14

Algorithms*	Constraints				PIS=	NIS=	RC	Ranking
	Max	Min	Min	Min				
	F-score	TestTime	TrainTime	Consistency				
A11	0.832	88.16*	0.004	2.611	0.02792	0.03234	<b>0.53668</b>	33
A12	0.825*	0.648*	0.000	0.067	0.00257	0.04180	<b>0.94203</b>	18
A13	0.739*	0.014*	0.000	0.007	0.01504	0.03574	<b>0.70380</b>	28
A14	0.819*	1.161*	0.001	0.057	0.00341	0.04126	<b>0.92367</b>	23
A15	0.795*	0.453*	0.000	0.034	0.00687	0.03942	<b>0.85156</b>	24
A16	0.645*	0.000	0.000	0.001	0.02877	0.03305	<b>0.53463</b>	34
A17	0.838	0.79*	0.000	0.024	0.00063	0.04328	<b>0.98557</b>	<b>2</b>
A18	0.827	1.38*	0.161*	0.044	0.01790	0.03819	<b>0.68088</b>	30
A19	0.828	0.221*	0.000	0.014	0.00205	0.04241	<b>0.95392</b>	15
A20	0.829	0.29*	0.000	0.014	0.00190	0.04251	<b>0.95715</b>	13
A21	0.833	1.676*	0.000	0.020	0.00134	0.04281	<b>0.96967</b>	10
A22	0.837	2.304*	0.022*	0.022	0.00255	0.04223	<b>0.94299</b>	17
A23	0.791*	0.028*	0.001	0.009	0.00745	0.03923	<b>0.84041</b>	25
A24	0.835	0.084*	0.000	0.012	0.00103	0.04308	<b>0.97669</b>	7
A25	0.836	0.713*	0.000	0.021	0.00090	0.04311	<b>0.97950</b>	5
A26	0.822*	1.074*	0.001	0.021	0.00293	0.04176	<b>0.93440</b>	21
A27	<b>(ref) 0.842</b>	0.753*	0.000	0.013	0.00014	0.04373	<b>0.99681</b>	<b>1</b>
A28	0.824*	0.013*	0.107*	0.010	0.01209	0.03861	<b>0.76154</b>	27
A29	0.828	0.215*	0.003	0.013	0.00207	0.04228	<b>0.95323</b>	16
A30	0.832	0.065*	0.000	0.009	0.00146	0.04282	<b>0.96697</b>	11
A31	0.835	1.948*	0.002	0.058	0.00121	0.04267	<b>0.97245</b>	9
A32	0.82*	0.062*	0.001	0.012	0.00322	0.04166	<b>0.92833</b>	22
A33	0.837	0.412*	0.001	0.012	0.00075	0.04322	<b>0.98299</b>	<b>3</b>
A34	0.834	0.724*	0.001	0.014	0.00118	0.04292	<b>0.97318</b>	8
A35	0.835	0.076*	0.000	0.009	0.00103	0.04310	<b>0.97676</b>	6
<b>RW</b>	<b>0.69520</b>	<b>0.05067</b>	<b>0.10097</b>	<b>0.15315</b>				
<b>PIS</b>	<b>0.12296</b>	<b>0.00874</b>	<b>0.01776</b>	<b>0.02647</b>				
<b>NIS</b>	<b>0.09419</b>	<b>0.00000</b>	<b>0.00000</b>	<b>0.00000</b>				

\*F-score: WgtAvgF-score

\*RW: relative weights

\*TestTime: CPUTimeTesting

\*PIS: Positive Ideal Solution

\*TrainTime: CPUTimeTraining

\*NIS: Negative Ideal Solution

\*Algorithms: See Table 4.4

The relative closeness score (RC) (8th column) is produced for which the corresponding ranking is generated in the 9th column. This column is the recommended ranking for the algorithms. According to this ranking, meta.Dagging, trees.BFTree and meta.RandomSubSpace are ranked first, second, and third, respectively, on the ADA\_Agnostic dataset. For evaluation of these

results, an evaluation criteria and methodology is used, which is described in the next section.

## 4.5. Experiments and evaluation

### 4.5.1. Classifiers and datasets

We performed the experiments on 35 most commonly used multi-class classification algorithms, shown in Table 4.4, which are implemented in Weka machine learning library [122]. These algorithms belong to six heterogeneous families' of classifiers including: probabilistic learners, functions-based learners, decision trees learners, rules-based learners, meta-learners, and miscellaneous learners. The meta-classifiers, i.e., Adaboost M1, Randomspace, and Voting are used with REPTree as the base classifier. Similarly, Dagging and Stacking are used with Naïve Bayes as the base classifier. The rest of algorithms are used with Weka default parameters. Similarly, 15 classification datasets<sup>2</sup>, shown in Table 4.5, from UCI machine learning repository [115] and OpenML repositories [134] are used.

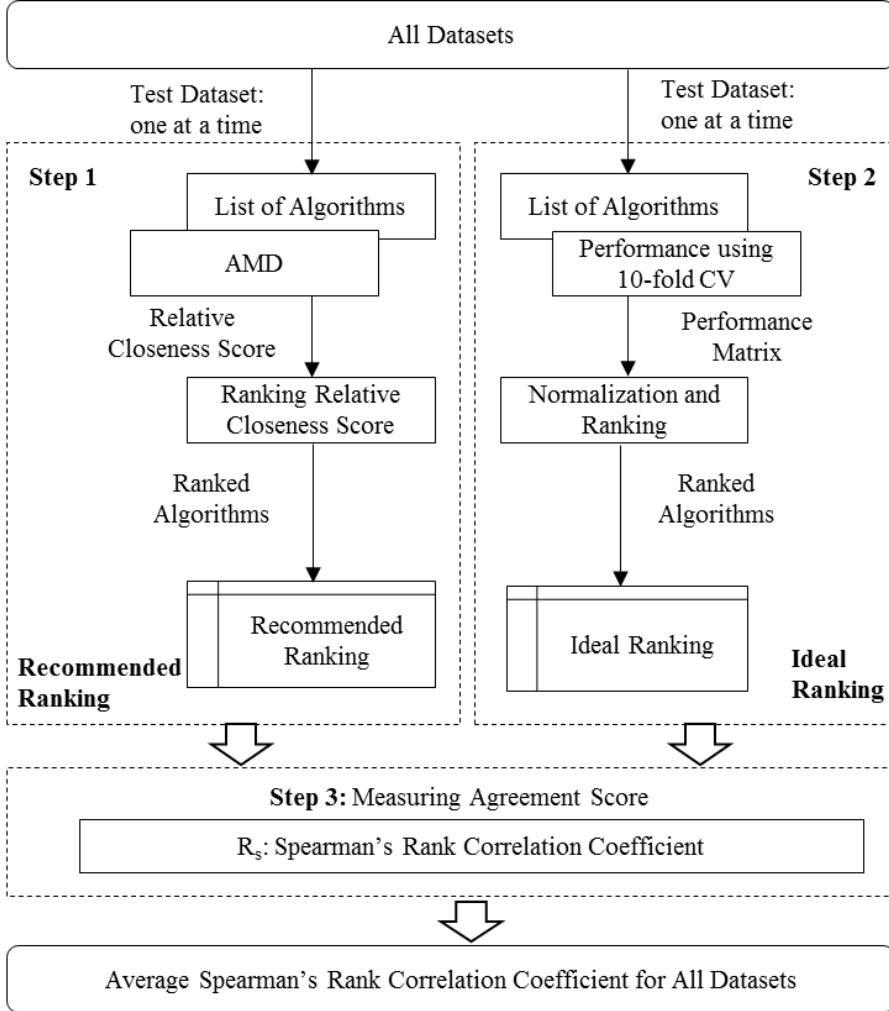
### 4.5.2. Evaluation methodology and criteria

To empirically evaluate the recommended ranking, the follows three steps methodology [32] is used, which is pictorially depicted in Figure 4.6.

- i. build a recommended ranking for a dataset  $d$  using the proposed AMD method
- ii. build an ideal ranking for dataset  $d$ , and
- iii. measure the agreement score between the two rankings

---

<sup>2</sup> Some of the datasets are used with minor modifications by changing the type of the class label to nominal etc.



**Figure 4.6.** Evaluation methodology of recommended ranking against ideal ranking

In step (i), the recommended ranking is obtained from the relative closeness score, which is computed using the proposed AMD method. In step (ii), the ideal ranking (IR) are constructed by applying ranking operation to the average score of algorithms performances, obtained by taking average of the weighted sum of normalized performance results of all the algorithms,  $A$ , on dataset  $d$ . We proposed the weighted sum average multi-criteria ideal ranking method (WAMR), described in equation 12 and 13, where the steps performed follow the sequence: (a) performance results for each metric are estimated (i.e.,  $s_{ij}$  is produced) using 10x10-fold CV, (b) normalized performance (i.e.,  $NS_{ij}$ ) is estimated using equation 13, (c) weighted performance, i.e.,  $W_j * NS_{ij}$  is computed, (d) weighted sum, i.e.,

(68)



$\sum_{j=1}^m (W_j * NS_{ij})$ , results are generated for all the metrics, (e) average of the weighted sum score is taken, and finally (f) ranks are generated. This process is described as follow,

$$IR = \text{rank} \left( \frac{\sum_{j=1}^m (W_j * NS_{ij})}{m} \right), \quad (12)$$

where,  $W_j$  is weight vector of evaluation metrics,  $E$ ,  $m$  is the number of evaluation metrics and  $NS_{ij}$  is the normalized performance value of the  $i^{th}$  algorithm for  $j^{th}$  evaluation metric, computed using equation 13,

$$NS_{ij} = \frac{s_{ij}}{\sqrt{\sum_{i=1}^n s_{ij}^2}}, \quad (13)$$

where,  $i = 1, 2, \dots, n$  and  $j = 1, 2, \dots, m$ .

The rank operation of equation 12 is described in algorithm 1. Similarly, in equation 13, the value  $s_{ij}$  is the performance of  $i^{th}$  algorithm for  $j^{th}$  evaluation metric, obtained using 10x10-fold cross-validation strategy (CV). Moreover the variables  $n$  and  $m$  represents number of algorithms and number of evaluation metrics, respectively.

In literature, different methods are used to compute ideal ranking, such as N-orderings, average correlation (AC) and average weighted correlation (AWC) [135, 136]. In  $N$  orderings method [32], first 10-fold CV results are generated for all the algorithms on a single dataset and a pair-wise comparison using statistical significance tests is performed. The algorithms are ordered based on their significance results score. In the average correlation method, ranks are computed for each fold of the 10-fold CV results which are then averaged to get the ideal rank. All the algorithms are arranged based on their average correlation score. Similarly, in the AWC method, weights are assigned to the ranks of individual folds and are then averaged together for get the final ranks.

The motivation for proposing the new ideal ranking generation method, so called WAMR, is that it is designed for multiple-criteria rather than single criterion, where the following essential steps take place prior to ideal ranks generation, such as normalizations of the criteria values, weighting the normalized value for uniformity with the AMD method, aggregating the weighted performance of all the criteria and taking average to get global performance results.

In step (iii), the agreement score, which is the mean agreement between the recommended ranking and the ideal ranking, is measured using the Spearman's ranked correlation coefficient [137, 138]. The final value of the agreement is a measure of the quality of the recommended ranking and proves the level of correctness of the proposed AMD method. The formula for Spearman's rank correlation coefficient is shown in equation 14.

$$R_s = 1 - \frac{6 * \sum_{i=1}^n (IR_i - RR_i)^2}{n^3 - n}, \quad (14)$$

where,  $IR_i$  and  $RR_i$  are the ideal and recommended ranking of algorithm  $i$ , respectively, and  $n$  is the number of algorithms to compare. If the value of  $R_s = 1$ , it represents a perfect agreement and if  $R_s = -1$ , it represents a perfect disagreement. If  $R_s = 0$ , then both the ranks are not related. Significance of Spearman rank correlation can be determined by looking in the table of critical values for  $R_s$  with different levels of significance, i.e.,  $\alpha$  value [139]. Similarly, the overall result for all the datasets is evaluated using the average Spearman rank correlation coefficient ( $AvgR_s$ ). This is shown by equation 15,

$$AvgR_s = \frac{\sum_{i=1}^d R_s(d_i)}{d}, \quad (15)$$

where,  $R_s(d_i)$  is the Spearman's rank correlation coefficient for dataset  $d_i$  and  $d$  is the total number of datasets.

### 4.5.3. Experiments and analysis of the results

In this section we perform a set of experiments and analyze the results from diverse perspective to validate the proposed AMD methodology. The set of experiments includes: (a) correctness check using average Spearman's correlation coefficient, (b) generalization power check using sensitivity and consistency, and (c) significance fitness evaluation.

#### 4.5.3.1. Correctness: average Spearman's rank correlation coefficient

To estimate correctness level of the proposed AMD, average Spearman's rank correlation coefficient is computed for all the datasets, using the proposed AMD methodology. The average of recommended rankings for all the datasets is shown in Table 4.11. The weights used for generating the recommended ranking are: Wgt.Avg.F-score (0.69520), CPUTimeTraining (0.05067), CPUTimeTesting (0.10097), and Consistency (0.15315). In the second step, ideal rankings for all the datasets are generated by taking average of the weighted sum of the normalized values of these evaluation metrics. Finally, the  $R_s$  is computed using equation 14 and the  $\text{Avg}R_s$  is calculated using equation 15.

**Table 4.11.** Average Spearman's rank correlation coefficient for 15 classification datasets

Dataset ID	Dataset Name	$R_s$
1	abalone-3class	<b>0.988</b>
2	rabe-148	0.985
3	acute-inflammations-nephro	0.994
4	ADA_Agnostic	0.990
5	ADA_Prior	0.991
6	adult-4000	0.983
7	adult-8000	0.975
8	aileron	0.979
9	analcata-AIDS	0.983
10	analcata-apnea2	0.932
11	analcata-apnea2	0.963
12	analcata-asbestos	0.973
13	analcata-authorship	0.999
14	analcata-bankruptcy	0.983
15	analcata-birthday	0.969
<b>Avg<math>R_s</math></b>		<b>0.979</b>

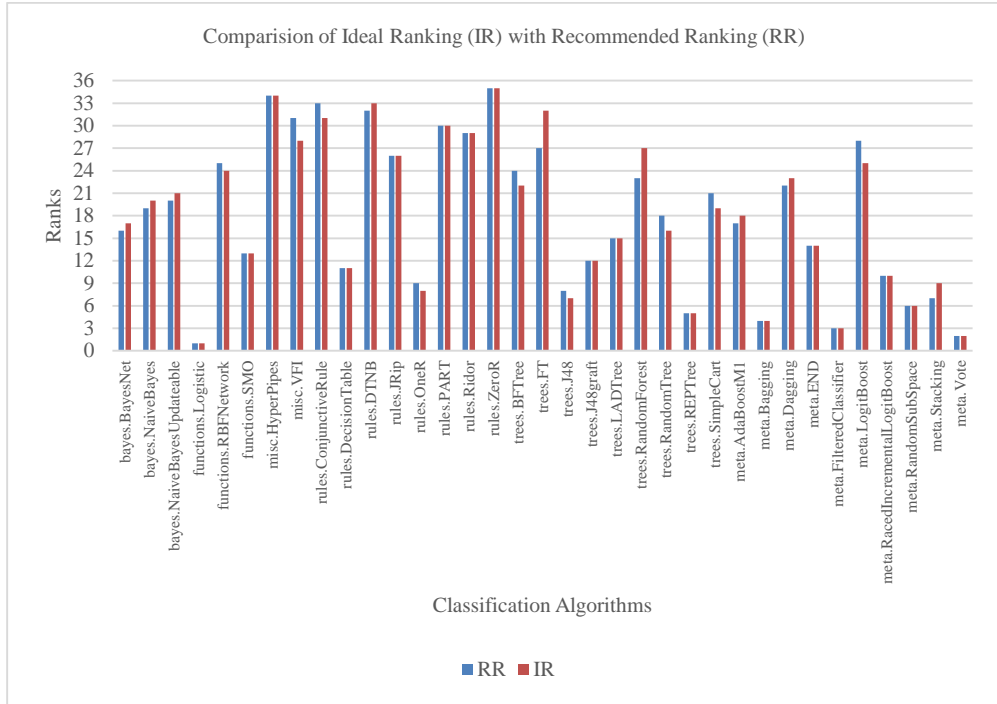
The  $\text{Avg}R_s$  value is very close to 1, which demonstrates correctness of the proposed AMD methodology. It accurately ranks the algorithms and thus assists experts in

the selection of accurate algorithms under the specified criteria. The statistical significance test of Spearman's rank correlation coefficient shows that the value 0.979 is statistically significant at the level of 0.001, with (35-2=33) degree of freedom (df), because the average correlation value 0.979 is far greater than the critical value of the correlation, i.e., 0.554. To show the process of calculating  $R_s$ , results for the abalone-3class dataset are shown in Table 4.12.

**Table 4.12.** Computation of Spearman's rank correlation coefficient

Algorithms	RR	IR	(IR-RR)	(IR-RR) <sup>2</sup>
bayes.BayesNet	16	17	1	1
bayes.NaiveBayes	19	20	1	1
bayes.NaiveBayesUpdateable	20	21	1	1
functions.Logistic	1	1	0	0
functions.RBFNetwork	25	24	-1	1
functions.SMO	13	13	0	0
misc.HyperPipes	34	34	0	0
misc.VFI	31	28	-3	9
rules.ConjunctiveRule	33	31	-2	4
rules.DecisionTable	11	11	0	0
rules.DTNB	32	33	1	1
rules.JRip	26	26	0	0
rules.OneR	9	8	-1	1
rules.PART	30	30	0	0
rules.Ridor	29	29	0	0
rules.ZeroR	35	35	0	0
trees.BFTree	24	22	-2	4
trees.FT	27	32	5	25
trees.J48	8	7	-1	1
trees.J48graft	12	12	0	0
trees.LADTree	15	15	0	0
trees.RandomForest	23	27	4	16
trees.RandomTree	18	16	-2	4
trees.REPTree	5	5	0	0
trees.SimpleCart	21	19	-2	4
meta.AdaBoostM1	17	18	1	1
meta.Bagging	4	4	0	0
meta.Dagging	22	23	1	1
meta.END	14	14	0	0
meta.FilteredClassifier	3	3	0	0
meta.LogitBoost	28	25	-3	9
meta.RacedIncrementalLogitBoost	10	10	0	0
meta.RandomSubSpace	6	6	0	0
meta.Stacking	7	9	2	4
meta.Vote	2	2	0	0
$\sum_{i=1}^n (IR_i - RR_i)^2$				<b>88</b>
$R_s = 1 - \frac{6 * \sum_{i=1}^n (IR_i - RR_i)^2}{n^3 - n}$				<b>0.988</b>

The interpretation of  $R_s$  result is the same as we did for the  $AvgR_s$ . A pictorial view of the results of recommended and ideal ranking for the abalone-3class dataset is shown in Figure 4.7.



**Figure 4.7.** Comparison of recommended ranking (RR) and ideal ranking (IR).

This figure shows that the recommended ranking of AMD is closed to the ideal ranking.

#### 4.5.3.2. Generalization of AMD: sensitivity and consistency analysis

In multi-criteria decision making, the choice and number or weights of the criteria affect the final recommended ranking [22, 140-142]. It has been demonstrated that the choice of criteria or the change in weights transforms the final recommended ranking [22, 140]. In majority of the algorithms ranking cases, it is hard for the decision makers to agree on the final ranks generated by a ranking method and is therefore required to perform sensitivity analysis [143, 144]. The significant results of the ranking method under varying parameters demonstrates generalization power of a ranking method. In our case, the scope of sensitivity analysis is limited

to the change in relative weights of criteria. We change the weight of each criterion, i.e., Wgt.Avg.F-score, CPUTimeTesting, CPUTimeTraining and Consistency, one at a time, and compute the Spearman's rank correlation coefficient value to see how the proposed AMD behaves with the changed weights. For the criteria Wgt.Avg.F-score, CPUTimeTesting, CPUTimeTraining and Consistency, the  $R_s$  results generated by the proposed AMD methodology using weights (0.70,0.05,0.10,0.15), (0.05,0.70,0.10,0.15), (0.05,0.10,0.70, 0.15) and (0.05,0.10,0.15,0.70) are shown in Table 4.13.

**Table 4.13.** Sensitivity analysis of classifiers with varying criteria weights

ID	Dataset Name	Sensitivity Analysis			
		$R_s$ - F-score (0.70,0.05,0.1 0,0.15)	$R_s$ - TestTime (0.05,0.70,0.10,0.15)	$R_s$ - TrainTime (0.05,0.10,0.70, 0.15)	$R_s$ - Consistency (0.05,0.10,0.15, 0.70)
1	abalone-3class	0.454	0.913	0.523	0.999
2	rabe-148	0.904	0.758	0.500	0.992
3	acute- inflammations-neph	0.858	0.798	0.501	0.979
4	ADA_Agnostic	0.880	0.368	0.819	0.433
5	ADA_Prior	0.295	0.943	0.565	0.985
6	adult-4000	0.276	0.890	0.599	0.979
7	adult-8000	0.488	0.792	0.670	0.943
8	aileron	0.946	0.223	0.806	0.563
9	analcata-AIDS	0.654	0.766	0.500	0.995
10	analcata-apnea2	0.107	0.844	0.652	0.986
11	analcata-apnea2	0.158	0.936	0.618	0.972
12	analcata-asbestos	0.508	0.838	0.500	0.999
13	analcata- authorship	0.880	-0.265	0.738	-0.074
14	analcata- bankruptcy	0.945	0.863	0.543	0.998
15	analcata-birthday	-0.506	0.777	0.618	0.990
<b>Avg<math>R_s</math></b>		<b>0.523</b>	<b>0.696</b>	<b>0.610</b>	<b>0.849</b>

\*F-score: WgtAvgF-score

\*TestTime: CPUTimeTesting

\*TrainTime: CPUTimeTraining

In Table 4.13, the  $R_s$  value for each set of the weights of the evaluation criteria is computed (using equation 14) and evaluated in the same way as in previous section. However, in this case, the ideal ranking is computed for the individual criteria and compared with the recommended ranking. In each set of the weights, more preference, i.e., weight 0.70, is given to only one criterion and thus algorithms are

preferred with respect to that criterion, which is natural. In Table 4.14, the  $R_s$  values shown in bold demonstrate negative/weak correlation with respect to the ideal ranking. The  $\text{Avg}R_s$  (for all datasets, computed using equation 15) in all the cases are positively correlated to ideal ranking, which demonstrate that the AMD is a generalized and consistent methodology that performs well in varying conditions. The statistical significance test of Spearman's rank correlation coefficient for the Wgt.Avg.F-score shows that the correlation value 0.523 is statistically significant at the level of 0.005-0.002, with (35-2=33) degree of freedom (df), because it is greater than the critical value 0.482 for  $R_s$ . Similar interpretations can be made for the rest of criteria.

#### 4.5.3.3. Significance fitness evaluation

The results of equation 8, which identifies significantly poor algorithms for the datasets are shown in Table 4.14.

**Table 4.14.** Analysis of significantly poor algorithms using significant fitness function

Algorithm	ADA_Agnostic (rank)	ADA_Prior (rank)	adult- 4000 (rank)	adult- 8000 (rank)	aileron (rank)	analcata- authorship (rank)
bayes.BayesNet*	26	4	2	7	27	<b>4</b>
bayes.NaiveBayes*	<b>19</b>	11	12	21	<b>30</b>	<b>7</b>
bayes.NaiveBayesUpdateable*	<b>20</b>	10	15	20	<b>31</b>	8
trees.FT*	30	<b>32</b>	<b>32</b>	<b>32</b>	25	2
trees.RandomForest*	17	<b>25</b>	23	<b>24</b>	<b>17</b>	6
meta.Dagging*	<b>27</b>	18	21	26	<b>32</b>	<b>30</b>

These results show that the classification algorithms bayes.BayesNet and bayes.NaiveBayes get higher ranks (4 and 7) on the analcata-authorship, however their performance on this dataset does not remain significant for all the criteria. Hence, prior applying the ranking process, the significance fitness function is required to execute to filter out insignificant algorithms from the competition. The values presented in bold represent the rank of algorithms on the dataset shown in the columns.

#### 4.5.4. Comparison with existing methods

In this section, we compare the results of AMD methodology with two well-known methods: adjusted ratio of ratios (ARR) [32] and automatic recommendation of classification algorithms based on data set characteristics, abbreviated as PAlg [40]. These methods evaluate and rank classification algorithms on the basis of accuracy and time.

The equation of ARR ranking methodology [32] is shown in equation 16,

$$ARR = \frac{\frac{SR_{ap}^{di}}{SR_{aq}^{di}}}{1 + \alpha * \log\left(\frac{T_{ap}^{di}}{T_{aq}^{di}}\right)}. \quad (16)$$

The accuracy is represented as the ratio of success rates of algorithm ap to algorithm aq on a dataset d as the numerator of the ARR. The time, which is the total of training and execution times, which is represented as a ratio of times is used as the denominator. To enforce preferences on the criteria, parameter  $\alpha$  is introduced with its value  $\alpha = 0.1, 1$ , and  $10$  to specify 10% preference of the accuracy on time, equal preferences of both the accuracy and time and 10% preference of time over the accuracy, respectively.

In the algorithm selection article [40], the performances of algorithms are evaluated using equation 17, where accuracy and total time are directly used instead of their ratios. The setting for  $\alpha$  is the same as that of the ARR method.

$$P_{Alg} = \frac{Accuracy_{Alg,D}}{1 + \alpha * \log(RTime_{Alg,D})} \quad (17)$$

As these two methods are only based on accuracy and execution and training time (T/RTime), therefore to create a fair comparison, we formulate our proposed criteria accordingly. We picked Wgt.Avg.F-score, CPUTimeTraining and CPUTimeTesting and omitted the Consistency criterion. The values of CPUTimeTraining and CPUTimeTesting are averaged to get the uniform value for T/RTime, used in equation 16 and 17, respectively. For simplicity, we performed



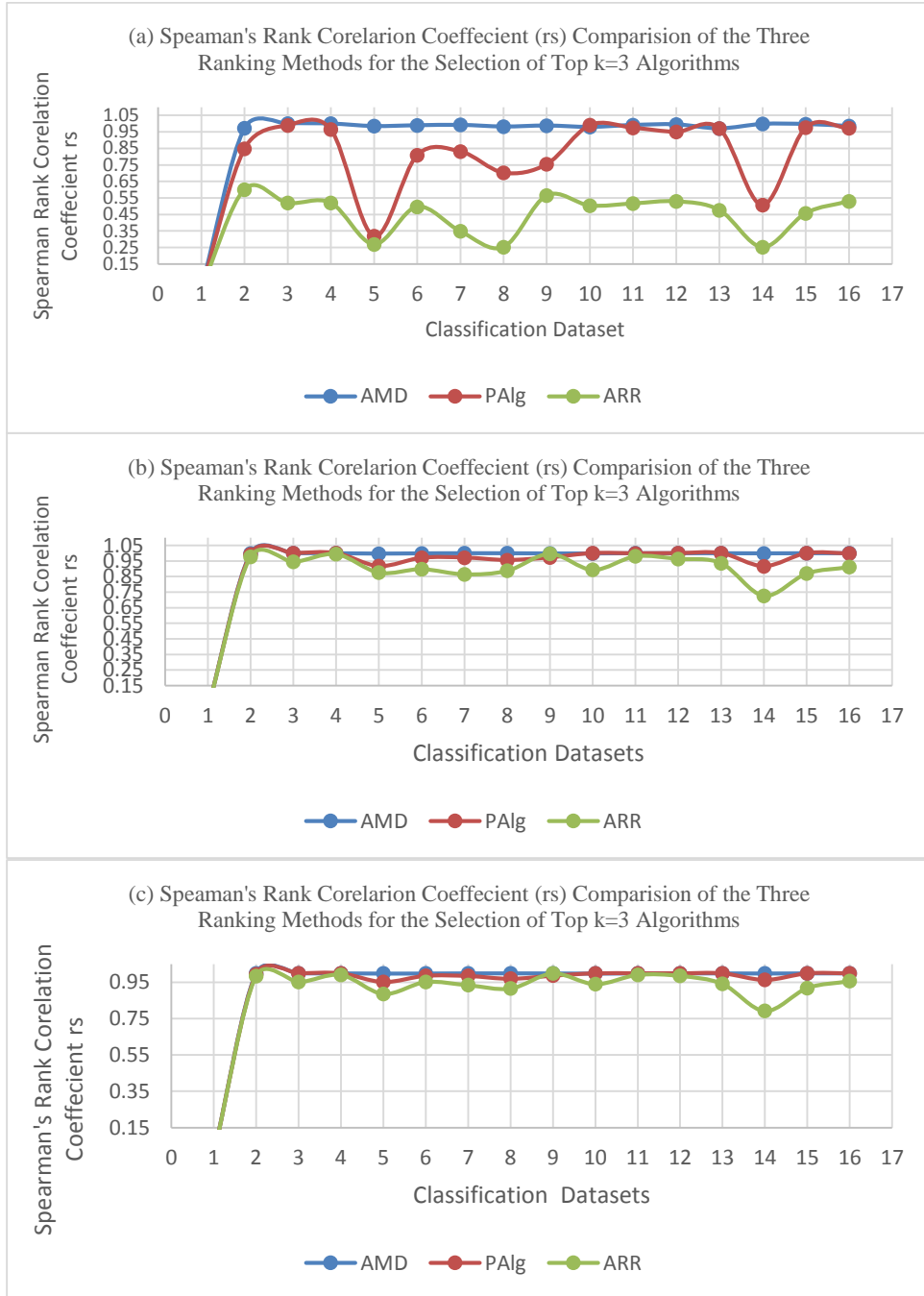
experiments only for  $\alpha = 0.1$  with three different sittings, such as ranking for all 35 algorithms (k=35), ranking for only top 5 algorithms (k=5) and ranking for top 3 algorithms (k=3). The weight for accuracy (Wgt.Avg.F-score) and T/RTime, in our proposed AMD method, were taken as 0.55 and 0.45, which are compliant to  $\alpha = 0.1$ .

We performed comparison experiments on the same 15 datasets and the results generated are shown in Table 4.15 and Figure 4.8(a-c).

**Table 4.15.** Comparison of AMD method with state-of-the-art methods

Id	Dataset	AMD			PAIlg			ARR		
		$R_s$ with $\alpha=0.1$ (Wgt.F-Score=0.55, Rtime=0.45)			$R_s$ with $\alpha=0.1$ (Wgt.F-Score=0.55, Rtime=0.45)			$R_s$ with $\alpha=0.1$ (Wgt.F-Score=0.55, Rtime=0.45)		
		k=35	k=5	k=3	k=35	k=5	k=3	k=35	k=5	k=3
1	abalone-3class	0.9720	0.9978	1.0000	0.8473	0.9926	0.9944	0.6012	0.9769	0.9842
2	rabe-148	1.0000	1.0000	1.0000	0.9900	1.0000	1.0000	0.5200	0.9450	0.9520
3	acute-inflammation-nephros	1.0000	1.0000	1.0000	0.9641	1.0000	1.0000	0.5199	0.9940	0.9908
4	ADA_Agnostic	0.9852	0.9974	0.9989	<b>0.3187*</b>	0.9171	0.9521	<b>0.2696*</b>	0.8752	0.8865
5	ADA_Prior	0.9899	0.9992	0.9993	0.8081	0.9699	0.9863	0.4966	0.8975	0.9515
6	adult-4000	0.9922	1.0000	1.0000	0.8314	0.9715	0.9851	<b>0.3482*</b>	0.8641	0.9342
7	adult-8000	0.9824	0.9997	1.0000	0.7028	0.9556	0.9697	<b>0.2529*</b>	0.8871	0.9158
8	aileron	0.9882	0.9986	0.9997	0.7541	0.9724	0.9869	0.5646	0.9956	0.9987
9	analcatdata-AIDS	0.9801	0.9985	0.9987	0.9908	1.0000	1.0000	0.5039	0.8929	0.9399
10	analcatdata-apnea2	0.9916	1.0000	1.0000	0.9748	0.9987	1.0000	0.5162	0.9799	0.9910
11	analcatdata-apnea2	0.9955	1.0000	1.0000	0.9501	1.0000	1.0000	0.5292	0.9636	0.9854
12	analcatdata-asbestos	0.9711	1.0000	1.0000	0.9706	1.0000	1.0000	0.4764	0.9359	0.9410
13	analcatdata-authorship	0.9980	0.9992	0.9993	0.5070	0.9164	0.9637	<b>0.2524*</b>	0.7271	0.7921
14	analcatdata-bankruptcy	0.9975	1.0000	1.0000	0.9756	0.9997	1.0000	0.4574	0.8694	0.9185
15	analcatdata-birthday	0.9854	1.0000	1.0000	0.9728	0.9977	1.0000	0.5298	0.9107	0.9567
AvgR <sub>s</sub>		<b>0.9886</b>	<b>0.9993</b>	<b>0.9997</b>	<b>0.8372</b>	<b>0.9794</b>	<b>0.9892</b>	<b>0.4559</b>	<b>0.9143</b>	<b>0.9426</b>

The performance results of the proposed AMD method are significantly better than the results of the PAlg and ARR under the three different setup: all ( $k=35$ ) algorithms, top  $k=5$  algorithms and top  $k=3$  algorithms. For the proposed method, the statistical significance test of Spearman's rank correlation coefficient shows that the correlation values,  $R_s = 0.9886$ ,  $R_s = 0.9993$ , and  $R_s = 0.9997$ , for  $k=35$ ,  $k=5$  and  $k=3$ , respectively, are statistically significant at the level of 0.001, with  $(35-2=33)$  degree of freedom (df). Similar interpretation can be made for PAlg method. However, this method produces ranks for the algorithms (with  $k=35$ ) on the ADA\_Agnostic dataset, which is statistically insignificant with respect to the ideal ranking. Similarly, the results of ARR method are significantly poor as compared to the proposed methods under all the conditions of  $k=35$ ,  $k=5$  and  $k=3$ . Under the setting,  $k=35$ , the ARR results are significant with respect to the critical value of  $R_s$  at the level of 0.01-0.005 with 33 degree of freedom. Using this method, four datasets, represented with '\*' has the ranks which are significantly poor and not correlated to the ideal ranking.



**Figure 4.8.** Comparison of the AMD method with state-of-the art methods

Figure 4.8 shows that AMD performs significantly better as compared to the state-of-the art methods under all the settings of top k=35, top k=5 and top k=3 algorithms.

#### 4.5.4.1. Statistical significance test for comparison of ranking methods

To test whether the results produced by AMD methodology are statistically significant or not as compared to the comparing methods, we performed Friedman's test [145]. First we set the following hypotheses:

- H0: There is no difference in the mean average correlation coefficients,  $\text{AvgR}_s$ , for the three ranking methods (AMD, ARR and PAIlg with all the datasets).
- H1: There are some differences in the mean average correlation coefficients,  $\text{AvgR}_s$ , for the three ranking methods.

For illustrating Friedman's test process and the corresponding results, we compare the three ranking methods (i.e.,  $j = 1, 2, 3$ ) on the 15 datasets. All the steps are shown in Table 4.16(a-c). The steps are performed as follows: (a) rank the correlation coefficients for each dataset, i.e., RR, (b) calculate the mean rank for each method, i.e.,  $\text{RR}_j = \sum_j \text{RR}_j / n$ , where  $n$  is the number of datasets (15 in this case), (c) calculate the overall mean rank ( $mR$ ) across all the methods, i.e.,  $mR = (m + 1)/2 = 2$ , where  $m$  is the number of methods to compare ( $m=3$  in this case), (d) calculate sum of the squared differences of mean rank for each method and the overall mean rank, i.e.,  $S = \sum_j (\text{RR}_j - mR)^2$ , and (e) calculate Friedman's statistic,  $M = (12nS)/(k(k + 1))$ .

The calculation of these steps is shown in Table 4.16(a-c), for all the fifteen datasets, and the results are summarized in Table 4.17. In the example of Table 4.16, where  $n = 15$  and  $m = 3$ , the critical value  $C$  is 10.99 for a confidence level of 95%. The Friedman's test values ( $M$ ) for  $k=35$ ,  $k=5$  and  $k=3 > C(10.99)$  is true, therefore the null hypothesis is rejected, which means 0.083 second that the average performance of the three methods is not similar and hence AMD is significantly better than state-of-the-art methods in comparison.

#### **4.6. Limitations of AMD method for Classifier Selection**

The proposed AMD method performs well as compared to the existing state-of-the-art methods, described in comparison section, however it comes with the following shortcomings that need proper research in future to overcome.

1. As described earlier, for ranking classifiers, correct criteria, based on suitable metrics is required. In the proposed AMD methodology, the criteria selection is depended on end user's goal, user level and system level constraints and specially the experts' knowledge about the domain and the available candidate algorithms. If the information are not available the proposed methodology will not be well-exploited for suitable classifier selection.
2. The proposed method has provided minimum support for the automatic criteria selection. A partially automatic solution, in the form of classifiers quality meta-metric classification model, is provided, however it is not enough to reduce the experts' efforts and time. To resolve this issue an advanced method is required to minimize the experts' time and efforts by introducing a semi-automatic analysis method for analyzing the classifiers performance metrics against the goal and constraints defined by the end user for his/her application.
3. The AMD methodology uses relative criteria weighting mechanism which is a semi-automatic way requiring experts' preferences for quantifying their opinion in the form of weights. However, experts' availability is not always be guaranteed, therefore some other mechanism need to be designed to estimate criteria weight.
4. The proposed method is based on exhaustive search mechanism to rank algorithms and finally select a single one for the application in hand. A hierarchical searching mechanism is required to filter-out the most unfit algorithms from the competition and reduce the search scope for recommending suitable algorithm.

**Table 4.16.** Friedman's test steps to compare ranking methods for statistical significance

(a) Friedman's test steps for comparing ranking methods with k=35

Dataset	d <sub>1</sub>		d <sub>2</sub>		d <sub>3</sub>		d <sub>4</sub>		d <sub>5</sub>		d <sub>6</sub>		d <sub>7</sub>		d <sub>8</sub>		d <sub>9</sub>		d <sub>10</sub>		d <sub>11</sub>		d <sub>12</sub>		d <sub>13</sub>		d <sub>14</sub>		d <sub>15</sub>			
Method\Rs	R <sub>s</sub>	RR	R <sub>s</sub>	RR	R <sub>s</sub>	RR	R <sub>s</sub>	RR	R <sub>s</sub>	RR	R <sub>s</sub>	RR	R <sub>s</sub>	RR	R <sub>s</sub>	RR	R <sub>s</sub>	RR	R <sub>s</sub>	RR	R <sub>s</sub>	RR	R <sub>s</sub>	RR	R <sub>s</sub>	RR	R <sub>s</sub>	RR	R <sub>s</sub>	RR	RR <sub>j</sub>	(RR <sub>j</sub> -mR) <sup>2</sup>
AMD	0.9720	1.0	1.0000	1.0	1.0000	1.0	0.9852	1.0	0.9899	1.0	0.9922	1.0	0.9824	1.0	0.9882	1.0	0.9801	2.0	0.9916	1.0	0.9955	1.0	0.9711	1.0	0.9980	1.0	0.9975	1.0	0.9854	1.0	1.1	0.871111111
PAIlg	0.8473	2.0	0.9900	2.0	0.9641	2.0	0.3187	2.0	0.8081	2.0	0.8314	2.0	0.7028	2.0	0.7541	2.0	0.9908	1.0	0.9748	2.0	0.9501	2.0	0.9706	2.0	0.5070	2.0	0.9756	2.0	0.9728	2.0	1.9	0.004444444
ARR	0.6012	3.0	0.5200	3.0	0.5199	3.0	0.2696	3.0	0.4966	3.0	0.3482	3.0	0.2529	3.0	0.5646	3.0	0.5039	3.0	0.5162	3.0	0.5292	3.0	0.4764	3.0	0.2524	3.0	0.4574	3.0	0.5298	3.0	3.0	1
																										S	1.875555556					

(b) Friedman's test for comparing ranking methods with k=5

Dataset	d <sub>1</sub>		d <sub>2</sub>		d <sub>3</sub>		d <sub>4</sub>		d <sub>5</sub>		d <sub>6</sub>		d <sub>7</sub>		d <sub>8</sub>		d <sub>9</sub>		d <sub>10</sub>		d <sub>11</sub>		d <sub>12</sub>		d <sub>13</sub>		d <sub>14</sub>		d <sub>15</sub>			
Method\Rs	R <sub>s</sub>	RR	R <sub>s</sub>	RR	R <sub>s</sub>	RR	R <sub>s</sub>	RR	R <sub>s</sub>	RR	R <sub>s</sub>	RR	R <sub>s</sub>	RR	R <sub>s</sub>	RR	R <sub>s</sub>	RR	R <sub>s</sub>	RR	R <sub>s</sub>	RR	R <sub>s</sub>	RR	R <sub>s</sub>	RR	R <sub>s</sub>	RR	R <sub>s</sub>	RR	RR <sub>j</sub>	(RR <sub>j</sub> -mR) <sup>2</sup>
AMD	0.9978	1.0	1.0000	1.5	1.0000	1.5	0.9974	1.0	0.9992	1.0	1.0000	1.0	0.9997	1.0	0.9986	1.0	0.9985	2.0	1.0000	1.0	1.0000	1.5	1.0000	1.5	0.9992	1.0	1.0000	1.0	1.0000	1.0	1.2	0.64
PAIlg	0.9926	2.0	1.0000	1.5	1.0000	1.5	0.9171	2.0	0.9699	2.0	0.9715	2.0	0.9556	2.0	0.9724	3.0	1.0000	1.0	0.9987	2.0	1.0000	1.5	1.0000	1.5	0.9164	2.0	0.9997	2.0	0.9977	2.0	1.9	0.017777778
ARR	0.9769	3.0	0.9450	3.0	0.9940	3.0	0.8752	3.0	0.8975	3.0	0.8641	3.0	0.8871	3.0	0.9956	2.0	0.8929	3.0	0.9799	3.0	0.9636	3.0	0.9359	3.0	0.7271	3.0	0.8694	3.0	0.9107	3.0	2.9	0.871111111
																										S	1.528888889					

(c) Friedman's test for comparing ranking methods with k=3

Dataset	d <sub>1</sub>		d <sub>2</sub>		d <sub>3</sub>		d <sub>4</sub>		d <sub>5</sub>		d <sub>6</sub>		d <sub>7</sub>		d <sub>8</sub>		d <sub>9</sub>		d <sub>10</sub>		d <sub>11</sub>		d <sub>12</sub>		d <sub>13</sub>		d <sub>14</sub>		d <sub>15</sub>			
Method\Rs	R <sub>s</sub>	RR	R <sub>s</sub>	RR	R <sub>s</sub>	RR	R <sub>s</sub>	RR	R <sub>s</sub>	RR	R <sub>s</sub>	RR	R <sub>s</sub>	RR	R <sub>s</sub>	RR	R <sub>s</sub>	RR	R <sub>s</sub>	RR	R <sub>s</sub>	RR	R <sub>s</sub>	RR	R <sub>s</sub>	RR	R <sub>s</sub>	RR	R <sub>s</sub>	RR	RR <sub>j</sub>	(RR <sub>j</sub> -mR) <sup>2</sup>
AMD	1.0000	1.0	1.0000	1.5	1.0000	1.5	0.9989	1.0	0.9993	1.0	1.0000	1.0	1.0000	1.0	0.9997	1.0	0.9987	2.0	1.0000	1.5	1.0000	1.5	1.0000	1.5	0.9993	1.0	1.0000	1.5	1.0000	1.5	1.3	0.49
PAIlg	0.9944	2.0	1.0000	1.5	1.0000	1.5	0.9521	2.0	0.9863	2.0	0.9851	2.0	0.9697	2.0	0.9869	3.0	1.0000	1.0	1.0000	1.5	1.0000	1.5	1.0000	1.5	0.9637	2.0	1.0000	1.5	1.0000	1.5	1.8	0.054444444
ARR	0.9842	3.0	0.9520	3.0	0.9908	3.0	0.8865	3.0	0.9515	3.0	0.9342	3.0	0.9158	3.0	0.9987	2.0	0.9399	3.0	0.9910	3.0	0.9854	3.0	0.9410	3.0	0.7921	3.0	0.9185	3.0	0.9567	3.0	2.9	0.871111111
S																														1.415555556		

**Table 4.17.** Summary of Friedman's test results for comparing ranking methods

Friedman's Test	S	M	C	M vs. C	Interpretation
Top-K=35	1.876	28.133	10.99	M > C	M(28.13) > C(10.99) → null hypothesis is rejected at the confidence level $\alpha = 0.001$
Top-K=5	1.529	22.933	10.99	M > C	M(22.93) > C(10.99) → null hypothesis is rejected at the confidence level $\alpha = 0.001$
Top-K=3	1.416	21.233	10.99	M > C	M(21.23) > C(10.99) → null hypothesis is rejected at the confidence level $\alpha = 0.001$

## 4.7. Summary

In this chapter, firstly, we introduced the concepts of algorithms' quality meta-metrics (QMM), describing physical meaning of the evaluation criteria, and developed a classification model with the help of extensive literature study to assist experts in the selection of suitable evaluation criteria for comparison of the classifiers. Motivated from the experts' consensus-based nominal grouped technique, we proposed an experts group-based method for the selection of suitable evaluation metrics from a large set of evaluation metrics and satisfying the constraints defined by the users/experts at the goal and objectives definition time.

Secondly, we estimated consistent relative weights for the evaluation metrics using the expert group-based decision making using the analytical hierarchy process. The experts' preferences on the criteria are quantified effectively and the weights are checked for consistency. We have analyzed performance of classification algorithm using statistical significance test and our proposed fitness function to filter out algorithms, which are statistically insignificant on all the evaluation criteria. For ranking the algorithms, we computed the relative closeness value of all the algorithms with respect to the ideal ranking, using the AHP-based estimated weights and local and global constraints on the evaluation criteria. The local constraints on criteria are used to encourage and discourage some of the criteria based on the categorization as cost and benefit criteria. The global constraints are imposed in the form of consistency measure that takes the standard deviation of all the criteria and consider an aggregate value to evaluate the quality of the selected/recommended algorithm.

Finally, we evaluated the AMD methodology by conducting a series of experiments on 15 different classification datasets using 35 classification algorithms. We compared the results of AMD with two stat-of-the-art methods. Results shows that the proposed AMD methodology performing significantly better than state-of-the-art methods and produce good results.

## Chapter 5

### CBR-based Meta-learning and Reasoning for Accurate Classifier Selection

---

#### 5.1. Overview

In machine learning area, a large number of classification algorithms are available that can be used for solving the problems of prediction and classification in different domains. These classifiers perform differently on different learning problems. For example, if one algorithm perform better on one dataset, the same algorithm may perform badly on different dataset. The reason is that each dataset has different nature in terms of its local and global characteristics. Similarly, the number of candidate algorithms are also large in number and it is very hard, even for a machine learning practitioner or expert, to know the intrinsic behaviors of different algorithms on different datasets and are therefore unable to select a right algorithm for his problem in-hand. One way of determining the behavior of each algorithm on different datasets is to perform algorithms performance analysis on the results generated using cross-validation strategy. Once the results are generated, ranking is performed on the final score and the top ranker is selected is the applicable algorithm. In Chapter 4, AMD methodology is proposed that performs the same task, however this proposed idea is complex due to the exhaustive search and analysis process of the results. To support the AMD methodology with some automatic search mechanism, an automatic classifier selection methodology is required. This automatic selection of suitable classifier, for building a data mining application for a user's problem in-hand, is one of the most important tasks in machine learning applications development since the applied algorithm (classifier) has great impact on the overall performances of the resulting classifier. However, this automatic selection of classifier is a challenging task in computer science, because the algorithms exploit the structure of the input data problem under consideration. This makes the problem of algorithm selection as



a domain and application dependent task that requires knowing the characteristics of data and the objective of the user. Thus the automatic algorithm selection task is basically a three-fold process model, as described below.

- (i) Definition of the application specific goal and objectives by the user of the algorithm for his learning problem
- (ii) Determining a representative suitable set of characteristics of the available data in the form of aggregated global features, also termed as meta-features
- (iii) Designing an efficient and accurate integration method to correctly map the user goal and characteristics of data and hence recommend right algorithm for the given data

The goal is the meta-characteristic(s) of the classifiers in which the user is interested, e.g., the selected algorithm should be accurate and consistent as compared to the candidate list of algorithms. Chapter 5 has discussed this issue of determining the goal and analyzing the algorithms performances based on that goal using multi-criteria decision. The characteristics are meta-features of the dataset that represents different behaviors of the data. Each dataset can be viewed as multi-dimensional based on type of characteristic they own. The integration of these meta-features or characteristic of the data with the goal of the user can be represented by building a meta-learning model. The rest of the chapter describes the whole process of analyzing the classifier performance based on the user's goal, extracting suitable set of meta-features, building a Case-Base for case-based reasoning (CBR) methodology and recommending classifier for a new dataset.

In this chapter, we are presenting the idea of automatic classifier selection using CBR-based meta-learning methodology that automatically selects a right decision tree classifier from a set of nine candidate classifiers implemented in Weka library.

### **5.1.1. Key Contributions**

As discussed in Chapter 3, a large number of methods, models, frameworks and methodologies have been proposed for automatic algorithms selections, however they have their limitations that have been analyzed in Chapter 3. The key contributions made through the proposed CBR-based meta-learning approach are enlisted below.

- (i) Proposed a flexible and incremental meta-learning and reasoning based framework using CBR-based methodology integrated with multi-criteria decision making, for classifier evaluation, and data characterization using multi-view meta-features extraction.
- (ii) A new multi-metrics criteria is proposed for the evaluation of decision tree classifiers to select the best classifier as class label for the cases in training dataset (i.e., resolved cases in the proposed CBR methodology). Classifiers are analyzed based on their predictive accuracy and standard deviation, called consistency to select the best classifier as class-label.
- (iii) The idea of multi-view learning is proposed to learn the data from multiple perspectives, with each perspective representing a set of similar meta-features that reflects one kind of behaviors of the data. Each set of features is called a family that forms a view of dataset.
- (iv) Proposed a multi-level multi-view meta-reasoning methodology with a flexible and incremental learning model integrating CBR with the classifiers conflict resolving (CCR) method to accurately recommend the most similar case as the suggested classifiers for a given new dataset. For the CBR retrieval phase, accurate similarity matching functions are defined, while for the CCR method, weighted sum score and AMD method (presented in Chapter 4) are proposed.

## **5.2. CBR-based Meta-learning and Reasoning (CBR-MLR) Framework**

In this section, the architectural view of the proposed framework, shown in Figure 5.1, is focused and each module is described with the rationales behind its use in the framework. This framework is motivated from the Rice framework [146] initially

designed for the algorithm selection problem based on the data and algorithm characterization.

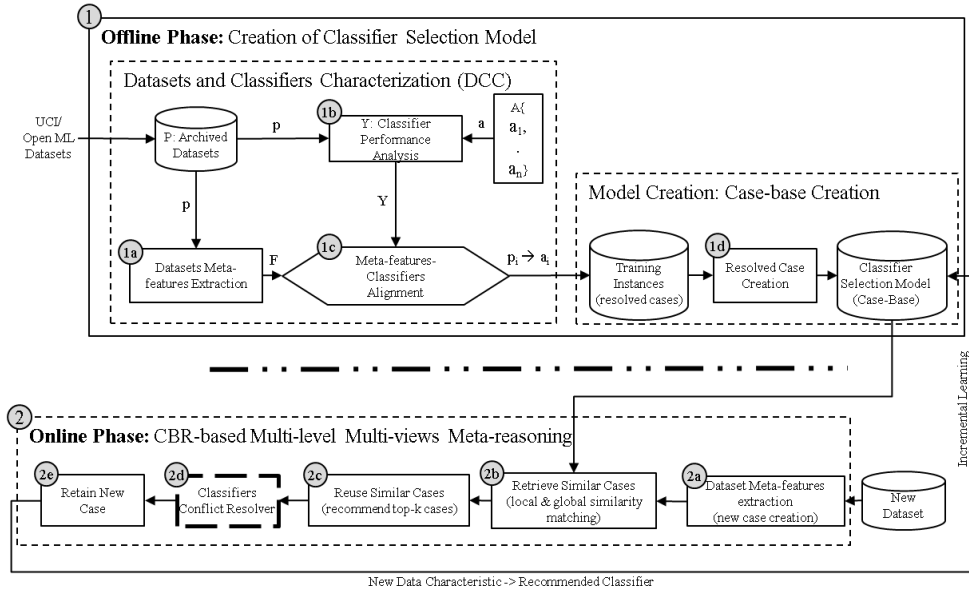
### **5.2.1. Definition of Algorithm Selection Problem**

Based on the Rice model [146], given a problem  $p$  as input, a set of candidate decision tree classifiers  $A$  that can learn the same  $p$  with different performance  $Y$ , find and select a decision tree classifier  $a \in A$  that can learn  $p$  with best possible performance. Now, we formally define the algorithm selection problem and introduce notation that we will use throughout this paper. Let  $P$  denotes a set of historical problems (i.e., classification datasets; in this case) with  $F$  as the features vector for representing the meta-features of each problem  $p \in P$  and  $A$  is a set of classification algorithms that can solve  $P$  with some performance  $Y$ .

### **5.2.2. Architecture of CBR-MLR Framework**

An abstract architecture of the proposed CBR- MLR framework is shown in Figure 5.1. As outlined in the overview, the problem of algorithm selection is a decision making problem with three main processes, the corresponding framework also consists of three modules. These includes:

- (i) Dataset and classifiers characterization (DCC)
- (ii) Algorithms selection model creation
- (iii) Multi-level Multi-view Meta-reasoning (MlvMr)



**Figure 5.1.** CBR-based meta-learning and reasoning framework for classifier selection

In the high level abstracted view (Figure 5.1), the proposed framework for the automatic classifier selection based on multi-view meta-learning consists of two main phases, offline phase and online phase, as described below.

#### 5.2.2.1. Offline Phase: Creation of Classifier Selection Model

This is the offline phase of the process of automatic classifier selection, where a model is developed that works as a knowledge model for real-world recommendation of a suitable classifier for a given new learning problem. It further consists of datasets and classifiers characterization and model creation processes, as described below.

- Datasets and Classifiers Characterization (DCC):** is the process of characterizing historical data problems  $P$  and classifiers  $A$  and mapping them against each other in way that the best classifier  $a$  is assigned the feature vector  $F$ . This produces resolved cases/instances for training purpose that are used in later step of model creation. This component is responsible for

extraction of meta-features  $F$  for each dataset  $d$  and relating/aligning the feature vector against the best classifier  $a \in A$ . The best classifier  $a$  in this case is computed using the multi-criteria decision making methodology (see Chapter 4), utilizing predictive accuracy and consistency measures from the classifiers performance space  $Y$ . In the data characterization process, different meta-features, belonging to different families, such as simple statistical, advanced statistical and information theoretic, are extracted to enable multi-view learning for best classifier selection from multiple perspectives.

- **Model Creation:** is the process of building classifiers selection model from the training instances produced by the DCC as output. Each training instance is a resolved case with meta-features as the problem description part and the best applicable classifier as the solution part or class label. This model can be created using different machine learning algorithms, however it is very hard to build such model using traditional learning methods due to the small number of training instances. To overcome this issue, we adopt the traditional CBR model with some enhancements in the case base creation and retrieval phases. In the proposed framework, output of the model creation is a case base of resolved cases that will be used in the online phase for real-world recommendation of right classifier for a given new dataset.

#### 5.2.2.2. Online Phase: CBR-based Multi-level Multi-view Meta-Reasoning (CBR-MlvMr)

This is the online phase of the process of automatic classifier selection, where a suitable classifier is recommended to the end user for his given new dataset. It further consists of meta-features extraction of the new dataset, application of the standard CBR methodology for selecting top-k similar cases from the case base (created model: case base) and resolving the conflict, if more than 1 similar classifiers are recommended by the CBR methodology. The detail are described as follows.

- **New Case Preparation (Multi-view Meta-features Extraction):** To recommend a classifier for a new dataset, first an un-resolved case, consisting

only feature vector, is prepared. For this purpose the same dataset characterization mechanism is used as described in the offline phase. Multiple families of meta-features, such as simple statistical, advanced statistical and information theoretic features are extracted in which each group represents a different view of the dataset. This makes the process of algorithm selection as a multi-view learning process.

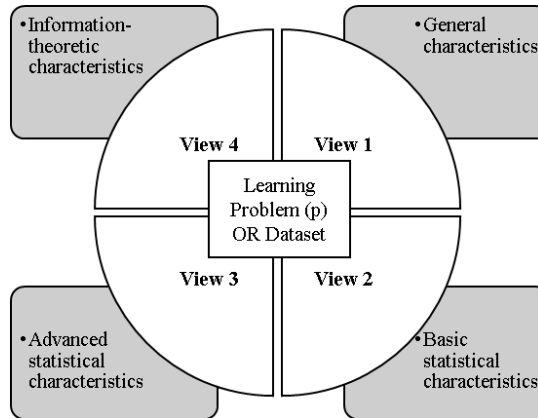
- **CBR-based Multi-views Meta-reasoning:** to recommend most suitable top-k classifiers for a new learning problem, represented as a multi-view meta-features case, a customized-CBR methodology with the retrieve, reuse and retain steps is used. Accurate local and global similarity functions are defined that search the algorithm selection model (i.e., the case base of the resolved case) and returns top-k (with  $k=3$ ) similar classifiers. If no classifier is the winner among  $k=3$ , then the classifiers conflict resolver step is activated prior to retain step to enable multi-level meta-reasoning.
- **Multi-level Meta-reasoning (Classifiers Conflict Resolver):** is enabled when the first level, CBR, reasoning recommends classifiers with similar performance score. At this second level of meta-reasoning, the classifiers meta-characteristics are used rather than the data characteristics to break the tie with a best decision. A weighted sum aggregate score computation criteria is proposed that consumes the classifiers characteristics, such as decision tree length, number of rules, depth etc., as input and returns an aggregate score to rank the tie classifier. The classifier with highest rank is selected and suggested to the end user for building his/her data mining application.
- **Retain New Case (Incremental/Evolutionary Learning):** is used to add the meta-features vector along with the recommended classifier as a new resolved case to the case base to improve quality of the system for future recommendations. One of the rationales behind the use of CBR-based methodology for classifier selection is the ability of CBR system to incrementally learn the domain and improve quality of the model with passage of time.

### 5.2.3. Methods of CBR-MLR Based Classifier Selection

This section describes the methods used in each step or module of the proposed CBR-MLR framework.

#### 5.2.3.1. Multi-view Data Characterization - Meta-Features Extraction

To design an accurate classifier, the selection of a best classifier is required. As described earlier, the selection of a best classifier is a multi-factors problem, where multiple parameters need to be considered. For example, how the classifier produce results, measured using various performance evaluation metrics? How the performance is affected by the nature of the data, which can be described in terms of data characteristics. The performance of classifiers varies from data to data. If the characteristics of data are accurately mapped against the performance of classifiers, it will help in understanding the relations of data to classifiers and ultimately will assist in the selection of a best classifier for a problem in hand. These characteristics of the data are termed as meta-features and the resulting model is called meta-learning model. Each dataset can be represented as a set of meta-features, grouped into various families, representing a different view of the dataset. A multi-view analysis of the dataset during classifier selection process enables the resulting model to best map the data, using all its characteristics, against the best classifier. The general concept of multi-view data characterization for classifier selection is shown in Figure 5.2.



**Figure 5.2.** Multi-view representation of classification dataset based on meta-characteristics

In state-of-the-art meta-learning methods for algorithm selection, the analysis or automatic algorithm selection model creation is based on various single view meta-features, which then recommends algorithms by considering only those specific features of the model for each given new dataset. A few examples of such views are statistical, information theoretic, complexity, landmarking and model-based [21] [40], that have been analyzed in Chapter 3. In this study, we propose a new multi-view meta-features based classifier selection model utilizing twenty nine characteristics from the general, basic statistical, advanced statistical and information theoretic views of the different available views of characteristics, as shown in Tables 5.1-5.4.

**Table 5.1.** General view (meta-characteristics) of classification dataset

Meta-Feature ID	Description
General 1	InstanceCount
General 2	NumAttributes
General 3	ClassCount
General 4	NumBinaryAtts
General 5	NumNominalAtts
General 6	NumNumericAtts
General 7	NumMissingValues

Basic view consists of simple measurements or general data characteristics of the dataset and are computed for the whole dataset, representing a global view using the aggregated values.

**Table 5.2.** Basic statistical view (meta-characteristics) of classification dataset

Meta-Feature ID	Description
Basic. Statistic 1	PercentageOfBinaryAtts
Basic. Statistic 2	PercentageOfNominalAtts
Basic. Statistic 3	PercentageOfNumericAtts
Basic. Statistic 4	MeanSkewnessOfNumericAtts
Basic. Statistic 5	MeanKurtosisOfNumericAtts
Basic. Statistic 6	Dimensionality

The basic statistical view consists of measurements representing the statistics regarding the dimensionality and ratios of different kinds of attributes in the dataset.



**Table 5.3.** Advanced statistical view (meta-characteristics) of classification dataset

Meta-Feature ID	Description
Advanced Statistic 1	MeanStdDevOfNumericAtts
Advanced Statistic 2	MeanMeansOfNumericAtts
Advanced Statistic 3	NegativePercentage
Advanced Statistic 4	PositivePercentage
Advanced Statistic 5	DefaultAccuracy
Advanced Statistic 6	IncompleteInstanceCount
Advanced Statistic 7	PercentageOfMissingValues
Advanced Statistic 8	MinNominalAttDistinctValues
Advanced Statistic 9	MaxNominalAttDistinctValues
Advanced Statistic 10	StdvNominalAttDistinctValues
Advanced Statistic 11	MeanNominalAttDistinctValues

**Table 5.4.** Information theoretics view (meta-characteristics) of classification dataset

Meta-Feature ID	Description
InfTheory 1	ClassEntropy
InfTheory 2	MeanAttributeEntropy
InfTheory 3	MeanMutualInformation
InfTheory 4	EquivalentNumberOfAtts
InfTheory 5	NoiseToSignalRatio

Every dataset is a combination of continuous and symbolic data features, therefore to best analyze the data for algorithm selection, the set of symbolic meta-features are also extracted, which are collectively termed as information-theoretic features. These features are based on the entropy that measures the purity level of the data with respect to the class label.

The rationales behind the selection of only these three views of meta-features are: (i) they are the global features representing every kind of classification data and (ii) can easily be computed on the fly to support building real-world application development for data mining application. These meta-features are computed using OpenML [134] data characteristics (DC) open source library, available on GitHub [147].

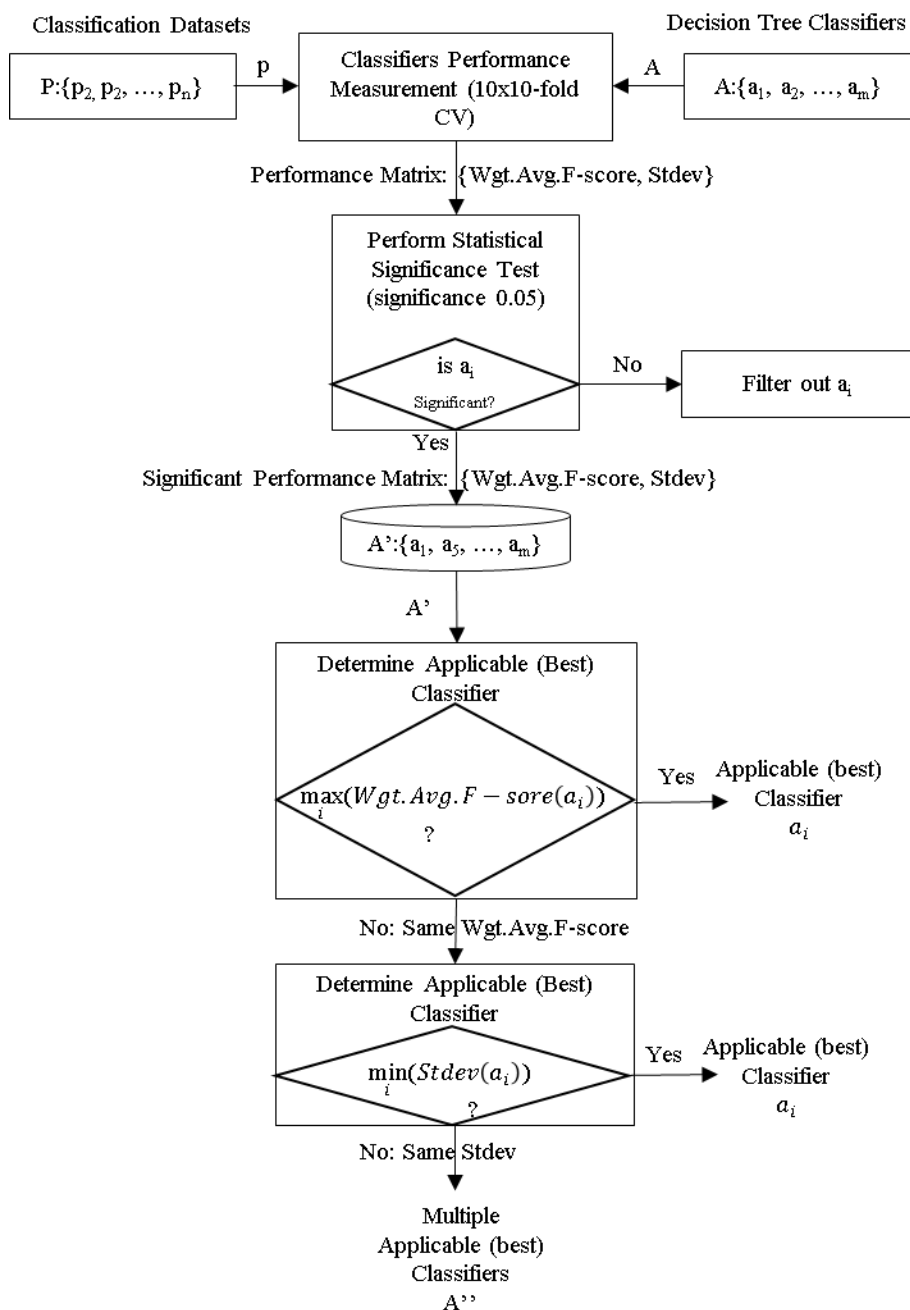
### 5.2.3.2. Multi-view Classifiers Characterization – Multi-criteria Performance Analysis

In the proposed study, the classifiers performance analysis process is designed to determine best performance classifier amongst the candidate classifiers and make it class label of the resolved case, in the case base/training dataset. The AMD methodology, described in Chapter 4, is used to perform this analysis. The performance results, for each dataset (p), are generated using the candidate set of classifiers (A) with a standard setting of 10x10-fold cross validation in Weka experimenter environment [122]. In this study, we used nine decision tree classifiers, implemented in Weka library, with their default parameters. Table 5.5 shows the list of these classifiers.

**Table 5.5.** List of Decision Tree classifiers from Weka library

Decision Tree Classifier ID	Name of Decision Tree Classifier
DT1	trees.BFTree
DT2	trees.FT
DT3	trees.J48
DT4	trees.J48graft
DT5	trees.LADTree
DT6	trees.RandomForest
DT7	trees.RandomTree
DT8	trees.REPTree
DT9	trees.SimpleCart

In each experiment, on each dataset (p), results are generated by all the classifiers (A) mentioned in Table 5.5. The results are stored into Performance Matrix. To determine, the applicable (best) classifier for the dataset (p) under consideration, we use performance metrics, predictive accuracy, measured in terms of Wgt.Avg.F-score and standard deviation (Stdev) in a sequential manner. Prior to this analysis, we use statistical significance test with significance level 0.05 to filter out those classifiers which are statically insignificant with 0.05 level of significance. The algorithm procedure used for this process is pictorially shown in Figure 5.3.



**Figure 5.3.** Multi-criteria based classifiers performance evaluation method

In figure 5.3, the processes of applicable classifier(s) identification is sequentially shown following the steps as described below.

- (i) Performance matrix is computed using 10x10-fold cross validation
- (ii) Statistical significance test is performed to filter out insignificant classifiers and reduce the algorithms space from A to A'
- (iii) Applicable classifier is determined from the list A' using the maximum Wgt.Avg.F-score sorting function. If more than one classifiers have same Wgt.Avg.F-score, step (iv) is used
- (iv) Applicable classifier is determined using the minimum Stdev function. If more than one classifiers have same Stdev values, step (v) is used
- (v) Multiple applicable classifiers (A'') are available from the same dataset, based on the considered performance metrics Wgt.Avg.F-score and Stdev only.

For further conflict resolution amongst these classifiers, other criteria can be used. However, in this study, we build the case-base only on these two metrics and use additional conflict resolution criteria at the later stage of online recommendation of best classifier for a given new dataset. For this purpose, i.e., conflict resolution amongst similar classifiers, the characterization of classifier is done from another view as well, where the characteristics of classifiers comprehensibility and interpretability are exploited (See reasoning section).

#### **5.2.3.3. Model Creation – Feature-vector (Propositional) Representation**

Once the dataset and classifiers are characterized, as described in previous sections, they are aligned with each other, i.e., applicable classifier(s) are assigned to the set of meta-features (e.g.,  $F \rightarrow applicableClassifier(s)$ ) using a simple alignment function to produce one single instance of training dataset. The mapping of features versus classifiers forms resolved cases for a CBR classifier. We adapt, propositional case representation schemes [148], where a case is represented as a proposition containing key-value pair format. In our proposed algorithm selection scenario, a case contains data characteristic (i.e., extracted meta-features) as problem description and applicable algorithm name as solution description. A generic structure of our proposed case-base, using feature-vector representation, is shown in Figure 5.6.

**Table 5.6.** Case-base structure and feature-vector representation of resolved cases

Problem or Dataset Description/Characterization					Classifier Characterization
Case-ID	Meta-Feature 1	Meta-Feature 2	...	Meta-Feature 29	Applicable-Classifier
1	MFv1	MFv1	...	MFv29	Classifier <sub>1</sub>
2	MFv1	MFv1	...	MFv29	Classifier <sub>1</sub>
...	...	...	...	...	...
100	MFv1	MFv1	...	MFv29	Classifier <sub>3</sub>

\*MFv<sub>i</sub>: represents meta-feature value for the  $i^{\text{th}}$  meta-feature in the Case-Base

The meta-features 1-29, shown in Table 5.6, are the multiple views of data characteristics given in Tables 5.1-5.4. Similarly, the Applicable-classifier (last column) is the label of one or more, best decision tree classifier(s), from Table 5.5, determined using the methodology described in Figure 5.3. The size of the case-base is 100 resolved cases, authored from 100 freely available classification datasets from UCI [115] and OpenML [134] machine learning repositories. The descriptions of these datasets is given in Table 5.7.

**Table 5.7.** Datasets used in Case-Base creation with their brief descriptions

ID	Dataset Name	General Characteristics							
		Attributes	NominalAtts	NumericAtts	BinaryAtts	Classes	IncompInstances	Instances	MissingValues
1	abalone.arff	9	1	7	0	3	0	4177	0
2	abe_148.arff	6	0	5	0	2	0	66	0
3	acute-inflammations.arff	7	5	1	5	2	0	120	0
4	ada_agnostic.arff	49	0	48	0	2	0	4562	0
5	ada_prior.arff	15	8	6	1	2	88	4562	88
6	adult- 4000.arff	15	8	6	1	2	0	3983	0
7	adult- 80000.arff	15	8	6	1	2	0	8000	0
8	aileron - 5840.arff	41	0	40	0	2	0	5795	0
9	analcatdata_aids.arff	5	2	2	0	2	0	50	0
10	analcatdata_apnea1.arff	4	2	1	0	2	0	475	0
11	analcatdata_apnea2.arff	4	2	1	0	2	0	475	0

ID	Dataset Name	General Characteristics							
		Attributes	NominalAtts	NumericAtts	BinaryAtts	Classes	IncomplInstances	Instances	MissingValues
12	analcatdata_asbestos_ciupdated	4	2	1	1	2	0	83	0
13	analcatdata_authorship.arff	71	0	70	0	4	0	841	0
14	analcatdata_bankruptcy.arff	7	1	5	0	2	0	50	0
15	analcatdata_birthday.arff	4	2	1	0	2	30	365	30
16	analcatdata_bondrate.arff	12	7	4	1	5	1	57	1
17	analcatdata_boxing1.arff	4	3	0	1	2	0	120	0
18	analcatdata_boxing2.arff	4	3	0	1	2	0	132	0
19	analcatdata_braziltourism.arff	9	4	4	1	7	49	412	96
20	analcatdata_broadway.arff	10	6	3	1	5	6	95	9
21	analcatdata_broadwaymult.arff	8	4	3	1	7	18	285	27
22	analcatdata_chall101.arff	3	1	1	0	2	0	138	0
23	analcatdata_challenger.arff	6	4	1	0	2	0	23	0
24	analcatdata_chlamydia.arff	4	3	0	1	2	0	100	0
25	analcatdata_creditscore.arff	7	3	3	2	2	0	100	0
26	analcatdata_currency.arff	4	2	1	0	7	0	31	0
27	analcatdata_cyyoung8092.arff	11	3	7	2	2	0	97	0
28	analcatdata_cyyoung9302.arff	11	4	6	2	2	0	92	0
29	analcatdata_dmft.arff	5	4	0	1	6	0	797	0
30	analcatdata_donner.arff	4	3	0	1	2	0	28	0
31	analcatdata_draft.arff	6	2	3	0	2	1	366	1
32	analcatdata_election2000.arff	16	1	14	0	2	0	67	0
33	analcatdata_esr.arff	3	0	2	0	2	0	32	0
34	analcatdata_famufsu.arff	4	2	1	0	2	0	14	0
35	analcatdata_fraud.arff	12	11	0	10	2	0	42	0
36	analcatdata_germangss.arff	6	4	1	2	4	0	400	0
37	analcatdata_gsssexsurvey.arff	10	5	4	5	5	6	159	6
38	analcatdata_gviolence.arff	10	1	8	0	2	0	74	0
39	analcatdata_halloffame.arff	18	2	15	0	3	20	1340	20
40	analcatdata_happiness.arff	4	2	1	0	3	0	60	0
41	analcatdata_homerun.arff	28	14	13	7	2	1	163	9
42	analcatdata_impeach.arff	11	8	2	4	2	0	100	0
43	analcatdata_japansolvent.arff	10	1	8	0	2	0	52	0
44	analcatdata_lawsuit.arff	5	1	3	1	2	0	264	0
45	analcatdata_marketing.arff	33	32	0	0	5	35	347	79
46	analcatdata_michiganacc.arff	5	2	2	0	2	0	108	0
47	analcatdata_ncaa.arff	20	15	4	1	2	0	120	0
48	analcatdata_neavote.arff	4	2	1	0	2	0	100	0
49	analcatdata_negotiation.arff	6	1	4	1	2	17	92	26
50	analcatdata_olympic2000.arff	13	1	11	0	2	0	66	0

ID	Dataset Name	General Characteristics							
		Attributes	NominalAtts	NumericAtts	BinaryAtts	Classes	IncomplInstances	Instances	MissingValues
51	analcata_data_reviewer.arff	9	8	0	0	2	367	379	1368
52	analcata_data_runshoes.arff	11	6	4	5	2	14	60	14
53	analcata_data_supreme.arff	8	0	7	0	2	0	4052	0
54	analcata_data_uktrainacc.arff	17	0	16	0	2	25	31	150
55	analcata_data_votesurvey.arff	5	1	3	1	4	0	48	0
56	analcata_data_whale.arff	8	1	6	1	2	5	228	15
57	analcata_data_wildcat.arff	6	2	3	2	2	0	163	0
58	anneal.arff	39	32	6	14	6	0	898	0
59	anneal.ORIG.arff	39	32	6	7	6	898	898	22175
60	appendicitis.arff	8	0	7	0	2	0	106	0
61	ar1.arff	30	0	29	0	2	0	121	0
62	ar3.arff	30	0	29	0	2	0	63	0
63	ar4.arff	30	0	29	0	2	0	107	0
64	ar5.arff	30	0	29	0	2	0	36	0
65	arsenic-female-bladder.arff	5	1	3	0	2	0	559	0
66	arsenic-female-lung.arff	5	1	3	0	2	0	559	0
67	arsenic-male-bladder.arff	5	1	3	0	2	0	559	0
68	arsenic-male-lung.arff	5	1	3	0	2	0	559	0
69	audiology (binary version of audiology).arff	70	69	0	61	2	222	226	317
70	australian.arff	15	0	14	0	2	0	690	0
71	automobile.arff	26	10	15	3	6	0	159	0
72	autoMpg.arff	8	3	4	0	2	6	398	6
73	autos.arff	26	10	15	4	7	46	205	59
74	autoUniv-au6-1000.arff	41	3	37	2	8	0	1000	0
75	autoUniv-au7-1100.arff	13	4	8	2	5	0	1100	0
76	autoUniv-au7-700.arff	13	4	8	2	3	0	700	0
77	backache.arff	33	26	6	22	2	0	180	0
78	badges2.arff	12	3	8	3	2	0	294	0
79	balance-scale.arff	5	0	4	0	3	0	625	0
80	balloon.arff	3	0	2	0	2	0	2001	0
81	banana.arff	3	0	2	0	2	0	5300	0
82	bands.arff	20	0	19	0	2	0	365	0
83	bank32nh - 1956.arff	33	0	32	0	2	0	1918	0
84	bank8FM.arff	9	0	8	0	2	0	8192	0
85	banknote-authentication.arff	5	0	4	0	2	0	1372	0
86	basketball.arff	5	0	4	0	2	0	96	0
87	BC-breast-cancer-data.arff	10	9	0	3	2	9	286	9
88	biomed.arff	9	1	7	0	2	15	209	15

ID	Dataset Name	General Characteristics							
		Attributes	NominalAtts	NumericAtts	BinaryAtts	Classes	IncomplInstances	Instances	MissingValues
89	blogger.arff	6	5	0	2	2	0	100	0
90	blood-transfusion-service-center.arff.arff	5	0	4	0	2	0	748	0
91	bodyfat.arff	15	0	14	0	2	0	252	0
92	bolts.arff	8	0	7	0	2	0	40	0
93	boston.arff	14	2	11	1	2	0	506	0
94	boston_corrected.arff	21	3	17	1	2	0	506	0
95	breast-cancer.arff	10	9	0	3	2	9	286	9
96	breastTumor.arff	10	8	1	4	2	9	286	9
97	bridges_version1.arff	13	9	3	2	6	37	107	73
98	bridges_version2.arff	13	12	0	2	6	37	107	73
99	bupa.arff	7	0	6	0	2	0	345	0
100	car.arff	7	6	0	0	4	0	1728	0

In the proposed Case-Base, all the features are real numbers, therefore their data types are set to numeric.

#### 5.2.3.4. CBR-based Multi-level Multi-views Meta-reasoning (CBR-MlvMr)

The final output of the model creation is a labelled dataset, called Case-Base (in this scenario), that contains meta-characteristic of the datasets and classifiers. This Case-Base can equally likely be learned using any supervised machine learning algorithm to produce the corresponding meta-learning based classifier selection model. However, the problem of classifier selection is an estimation problem, where the algorithms performances, over datasets, are estimated rather than providing a cutting-edge solution, as looked in state-of-the-art methods for algorithms selection. This problem can be viewed as a multi-dimensional analysis of the data and classifiers characteristics and the estimation of a similar solution can best predict a suitable classifier for a given new learning problem. Keeping in view all these aspects, we adopt standard CBR as our meta-learner and reasoner and enhance its classical methodological steps by introducing accurate similarity functions along with its multi-level integration with a new reasoner to improve its final output and recommend a right classifier. Furthermore, the incremental learning capability of the



CBR methodology is exploited to improve the algorithms selection model (Case-Base in this case). The CBR-MlvMr methodology works as follows.

**New Case Preparation:** A Query Case ( $Q$ ) is prepared from a given new dataset using the meta-feature extractor.

**CBR-based Multi-views Meta-reasoning (CBR-MvMr):** The CBR methodological steps including retrieve, reuse and retain are used in sequential order if the finally resolved case is unique, otherwise the retain step is preceded by CCR method.

- In the retrieve step, similarity functions are defined for matching the meta-features of query case against the resolved cases  $R$  in Case-Base and retrieving top-k cases as the suggested solutions. For individual meta-features similarity matching, the local similarity function, shown in equation 1, is defined, while for matching the whole new case with the existing resolved case  $R$  in Case-Base, a global function, shown in equation 2, is defined.

$$\text{Sim}_l(\mathbf{nC}_{\mathbf{mf}_i}, \mathbf{eC}_{\mathbf{mf}_i}) = \text{idealSim}_{\mathbf{mf}_i} - \frac{d_l(\mathbf{nC}_{\mathbf{mf}_i}, \mathbf{eC}_{\mathbf{mf}_i})}{d_g(\mathbf{Max}_{\mathbf{mf}_i}, \mathbf{Min}_{\mathbf{mf}_i})} \quad (1)$$

where,  $\text{idealSim}_{\mathbf{mf}_i}=1$  &  $d_g(\mathbf{Max}_{\mathbf{mf}_i}, \mathbf{Min}_{\mathbf{mf}_i})$  is the global interval or range of the values of each continuous value meta-feature. Similarly,  $\mathbf{nC}_{\mathbf{mf}_i}$  represents meta-feature of new case and  $\mathbf{eC}_{\mathbf{mf}_i}$  represents meta-feature of existing case.

$$\begin{aligned} &\text{Sim}_g(\mathbf{nC}, \mathbf{eC}) \\ &= \frac{\alpha_1 * \text{Sim}_l(\mathbf{nC}_{\mathbf{mf}_1}, \mathbf{eC}_{\mathbf{mf}_1}) + \dots + \alpha_n * \text{Sim}_l(\mathbf{nC}_{\mathbf{mf}_n}, \mathbf{eC}_{\mathbf{mf}_n})}{\alpha_1 + \alpha_2 + \dots + \alpha_n} \end{aligned} \quad (2)$$

where,  $\alpha_i$  is the weight value of each  $\mathbf{mf}_i$  in the Case-Base and we assigned equal weight value to each meta-features, based on the assumption made that all the 29 meta-features are equally important for selecting a right classifier.

- In the reuse step, the solution part, i.e., the label of applicable classifier, of the top-k similar cases are assigned to the problem description part of the new case as a suggested solution (recommended classifier in this case).

This process of retrieve and reuse are described in Algorithm 1.

---

**Algorithm 1.** CBR-based multi-views meta-reasoning (CBR-MvMr) process for generating top-k cases

---

**Begin**

**inputs:**  $Q$  – //the queryCase, built from extracted meta features

$R = \{r_1, r_2, \dots, r_n\}$ ; // case base e.g. all previously resolved cases

$K$  – // int, number of CBR cases to return

**output:**  $C = \{c_1, c_2, \dots, c_k\}$ ; // Collection of top-k most similar CBR Cases

- 1  $I = \text{calculateSimilarityIntervals}()$  // Calculates similarity interval, weight for each feature
- 2  $S = \text{buildNNConfig}(I)$  // procedure : builds and returns NN similarity config
- 3  $RR = \text{evaluateSimilarity}(R, Q, S)$  // procedure defined below
- 4  $C = \text{selectTopK}(RR, K)$  //Select top K CBR cases from RR

**End**

**PROCEDURE: buildNNConfig(I)**

**Begin**

**inputs:**  $I = \{i_1, i_2, \dots, i_n\}$ ; //Similarity Intervals,  $i$  denotes an interval for specific feature, where  $i \in I$

**output:** **similarityConfig** // returns NN similarity configuration,

1. **simConfig** = new NNConfig(); // initialize similarity configuration
2. **simConfig.setDescriptionSimFunction**(global.average) //Global similarity function, see eq. 2
3. **foreach** SimilarityInterval  $i$  in  $I$
4. **simConfig.addMapping**( $i.getFeature$ , **interval**( $i.getInterval$ ))//**interval** local similarity function, see eq. 1
5. **simConfig.setWeight**( $i.getFeature$ ,  $i.getWeight$ )
6. **return simConfig**

**End**

**PROCEDURE: evaluateSimilarity(R, Q, S)**

**Begin**

---

---

**inputs:**  $Q$  – //the queryCase, built from extracted meta features

$R = \{r_1, r_2, \dots, r_n\}$ ; // case base e.g. all previously resolved cases

$S$  – // NN similarity configuration

**output: RetrievalResults : RR**

**Description:** evaluates similarity of each  $c_i$  where  $c_i \in C$  against this queryCase  $Q$  using similarity function mapped in NN similarityConfig  $S$ , and returns a collection of retrievalResults **RR**

**return RR**

**End**

---

## DESCRIPTIONS OF THE PROCEDURES

- i. **CalculateSimilarityIntervals:** This procedure loops through all meta-features, calculates interval value and defines weight for each of feature. The interval value is computed using  $d_g(\mathbf{Max}_{mf_i}, \mathbf{Min}_{mf_i})$ , while the weight assigned to each meta-feature is same, i.e., 1.
  - ii. **BuildNNConfig( $I$ ):** This procedure performs the main task of finding nearest neighbor computation. The set of tasks performed using this procedure are:
    - a. Initialize NNConfig
    - b. set *global similarity function*, see eq. 2
    - c. Map a *local similarity function* with each feature, see eq. 1
    - d. Set *weight* for each feature, i.e., assign 1 to each feature in this case
    - e. Return NNConfig
  - iii. **evaluateSimilarity( $R, Q, S$ ):** evaluates *similarity of each  $c_i$  where  $c_i \in C$*  against the *queryCase  $Q$*  using *similarity function* mapped in *NN similarityConfig  $S$* , and returns a collection of *retrievalResults RR* (most similar cases)
  - iv. **selectTopK( $RR, K$ ):** this procedure *Selects top  $K$  most similar CBR cases from the collection of retrievalResult RR*
- In the retain step, the recommended solution is added to the Case-Base. The Case-Base grows in size and improves quality of the algorithm recommendation model.

**Classifiers Conflict Resolver:** is enabled when the first level, CBR, reasoning recommends classifiers with similar performance score. At this second level of meta-reasoning, the classifiers meta-characteristics are used rather than the data characteristics to break the tie with a best decision. One of the criteria to select best of the best accurate and consistent classifiers (suggested by CBR-MvMr), is to use comprehensibility criteria. However, comprehensibility is a debatable concept and there is no universally acceptable criteria to quantify it, due to its nature of subjectivity and domain specificity. In the proposed study, we characterize the classifiers using the characteristics shown in Table 5.8, which can be used to measure comprehensibility of the decision tree (DT) classifiers.

**Table 5.8.** Decision Tree classifiers characterization

ID	DT Classifier Comprehensibility Characteristics
1	measureNumRules
2	measurePercentAttsUsedByDT
3	measureTreeSize
4	measureNumLeaves
5	measureNumPredictionLeaves

The comprehensibility indirectly describes interpretability and understandability of the decision trees model, which are self-explanatory, by non-experts to grasp the knowledge represented in the model [149]. In state-of-the-art methods, the comprehensibility is evaluated using the size characteristic of the model, i.e., decision tree size [150], however it has a number of issues as described in [151]. In some of the domain applications, e.g., bioinformatics a larger tree size is favored by physicians rather than smaller size as in business application, etc. Similarly, there is always accuracy-comprehensibility-complexity trade-off, which means if one is increased the other is decreased. To overcome this issue, a multi-objective criteria need to be defined, to provide an optimum solution for the final comprehensible classifier selection. In this thesis, the following solutions are recommended.

- (i) Define an aggregate weighted sum criteria

- (ii) Use multi-criteria decision making methodology, AMD (presented in Chapter 4)

As this conflict resolution is application dependent, therefore a semi-automatic expert-oriented criteria setting is required to add the required characteristics of the classifiers to form the criteria and their corresponding weight to compute the final preference score for each conflicting classifier. The aggregate scores are ranked for the conflicting classifiers and the one with top-rank is finally recommended to the end user for building his application.

## **5.2.4. Implementation, experiments and evaluation**

### **5.2.4.1. Implementation**

The proposed meta-learning and reasoning methodology for accurate classifier selection is implemented in Java environment as an Open source application. The key components of the methodology are the extraction of meta-features from the dataset and performing meta-reasoning by exploiting a Case-Base. These meta-features are computed using OpenML [134] data characteristics (DC) open source library, freely available on GitHub [147]. For the CBR-based reasoning process, jColibri 2.0, a case-based reasoning framework [152], is used where we implemented our own case similarity functions for accurate matching of the existing cases. The resulting CBR-based meta-learning and reasoning system is released as an Open Source application on GitHub with an extensible and adoptable implementation strategy to enable end [153] users to use it for selecting a suitable decision tree classifier for their applications data. The interface of CBR-MLR application for meta-features extraction is shown in Figure 5.4.

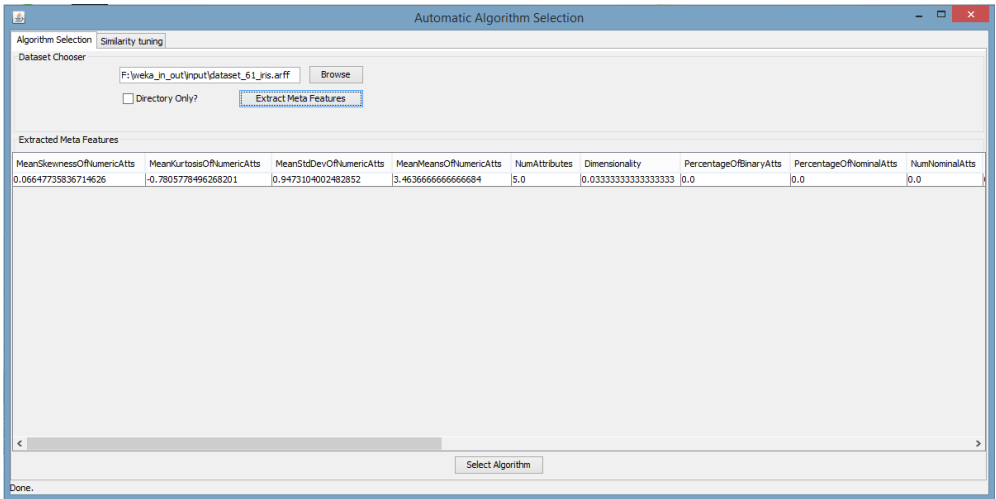


Figure 5.4. Meta-features extraction got new case creation

The process of meta-learning based multi-view reasoning using CBR is shown in Figure 5.5.

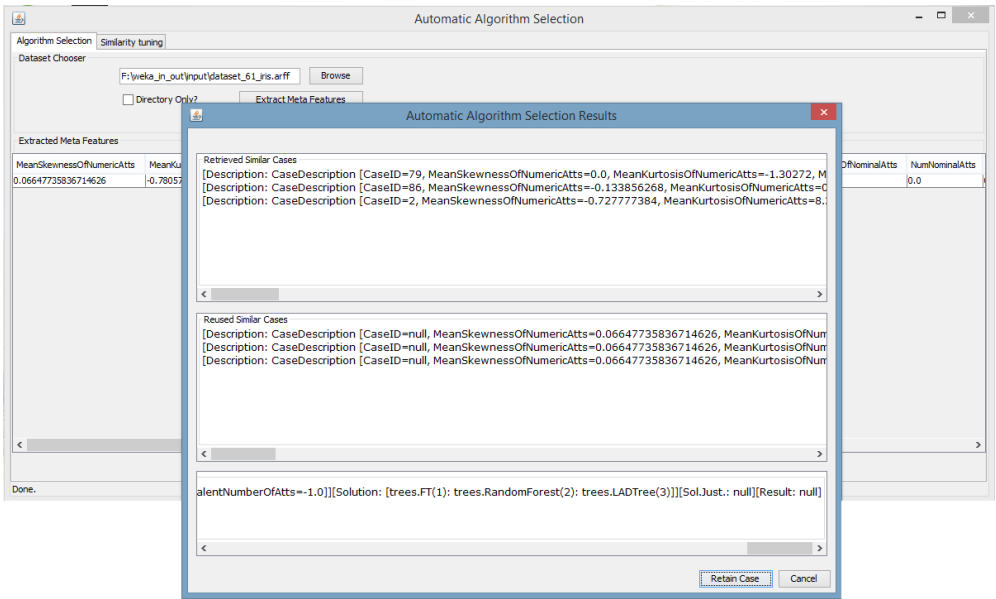


Figure 5.5. CBR-based reasoning for best classifier selection

## 5.2.4.2. Experimental Setup

### a. Classifiers used

We performed the experiments on 09 most commonly used multi-class classification algorithms, shown in Table 5.5, which are implemented in Weka machine learning library [122]. These algorithms belong to the decision tree family of classifiers and we have used them with their default parameters.

### **b. Training and testing datasets**

For training and testing the proposed methodology, two disjoint set of datasets are used. For training, the CBR model, i.e., Case-Base, is built using 100 multi-class classification datasets<sup>3</sup>, shown in Table 5.7, taken from UCI machine learning repository [115] and OpenML repositories [134], are used. Similarly, a 52 datasets disjoint set is used for testing the methodology. All the classifier of Table 5.5 are evaluated for each of the test dataset, using the method described in Figure 5.3, and the actual best classifiers are determined later on to see the performance of our proposed methodology.

#### **5.2.4.3. Evaluation methodology and criteria**

To evaluate the accuracy of the proposed method, the follows steps are used.

- i. For each given dataset (test datasets in this case), meta-features are extracted using the developed meta-feature extractor to prepare a Query Case (Q).
- ii. CBR-MvMr methodology is used to recommend top-k ( $k=3$ ) best classifiers for each Q.
- iii. Measure the similarity of the recommended top-k ( $k=3$ ) classifiers to the actual classifiers of those datasets. If the recommended classifier for a given dataset belongs to any of the top-k ( $k=3$ ) classifiers, the recommendation is declared as correct, otherwise incorrect.

---

<sup>3</sup> Some of the datasets are used with minor modifications by changing the type of the class label to nominal etc.

#### 5.2.4.4. Experiments and analysis of the results

When experiments on the test Case-Base of 52 datasets are performed and the results were evaluated, 48 out of 52 datasets were recommended with accurate classifiers. Hence, the overall accuracy of the proposed methodology was found as  $48 \times 100 / 52 = 94\%$  for the correct classifiers recommendations in top  $k=3$  actual classifiers.

**Table 5.9.** Results of meta-learning based method for decision tree classifier selection

Dataset ID	Dataset Name	Position of Recommended Classifier in Top-3 Actual Classifiers
d1	cardiotocograph-10clas	1
d2	cars	1
d3	cars_with_names	2
d4	CastMetal1	1
d5	chess-small	1
d6	cholesterol	1
d7	chscase-adopt	3
d8	chscase-census2	3
d9	chscase-census3	2
d10	chscase-census5	1
d11	chscase-census6	2
d12	chscase-funds	1
d13	chscase-geyser1	1
d14	hscase-health	1
d15	chscase-vine1	1
d16	chscase-vine2	3
d17	chscase-whale	1
d18	cjs	1
d19	cleveland	1
d20	climate-simulation-crac	3
d21	cloud	3
d22	cm1_req	1
d23	cmc	1
d24	horse-colic	1
d25	horse-colic.ORIG	2



Dataset ID	Dataset Name	Position of Recommended Classifier in Top-3 Actual Classifiers
d26	colleges-aaup	1
d27	colleges-usnews	2
d28	collins	1
d29	onfidence	2
<b>d30</b>	<b>ontact-lenses</b>	<b>9</b>
d31	contraceptive	3
d32	costamadrel	1
d33	cps_85_wages	3
d34	cpu	3
d35	cpu_act	1
d36	cpu_small	1
d37	credit-rating	3
<b>d38</b>	<b>crx</b>	<b>4</b>
d39	DATATRIEVE	1
d40	dbworld-subjects-stemme	1
d41	dbworld-subjects	1
d42	delta_ailerons	1
d43	dermatology	1
d44	desharnais.csv-weka.fil	2
d45	pima_diabetes	1
d46	diggle-Table_A1_Luteniz	2
d47	disclosure-X_BIAS	1
<b>d48</b>	<b>disclosure-X_NOISE</b>	<b>4</b>
d49	disclosure-X_TAMPERED	3
d50	disclosure-Z	3
d51	dresses-sales	1
d52	eastWest	1

The results of Table 5.9 shows that only 3/52 classifiers, i.e., d30, d38 and d48 were not recommended with correct classifiers. Similarly, for top k=1, the proposed methodology correctly recommended accurate classifiers for 30 datasets and hence the accuracy obtained is  $30 \times 100 / 52 = 57.6\%$ . To analyze the results in top k=2, the methodology correctly recommended classifiers for 38 dataset and achieved accuracy of  $30 \times 100 / 52 = 73\%$ .

### 5.3. Limitations of CBR-MLR for classifiers selection

The results show that the proposed framework performs well for accurate classifier selection using automatic recommendation method rather than using the exhaustive search mechanism and ranking classifiers performances to pick the best one. However, the methods used to implement the framework and generate accurate results, have a number of limitations that need proper handling and resolution. A few of these major limitations are mentioned here.

- **Finding an optimum and suitable set of meta-features:** the process of finding right classifier for a dataset using a machine learning model that is based only on the datasets global features is not enough and may lead to a wrong decision. The reason is that the proposed 29 features for the selection of classifier does not represents the whole meta-feature space of the datasets and thus cannot be declared as the final optimum list of features.
- **Classifiers performance analysis for selection of best classifier to find solution part of a resolved case:** while creating the successful cases, the proposed method analyses the performance results of the candidates classifiers using predictive accuracy and standard deviation, however this evaluation is application dependent. The users may interested in other characteristics of the classifiers to be selected. In that case, the proposed Case-Base may not work well for them and need to be updated according to their application requirements, which is an exhaustive experimental work. This is a difficult task and requires an efficient method to automatically or semi-automatically perform this analysis and produce the class label of the resolved cases.
- **Ranking classifiers with similar performance results (tie cases):** while analyzing the performance results of the classifiers for finding the best classifiers to make them class-labels, we perform the process of ranking the classifier. However, in case of small datasets most of the classifier perform with equal performance and are thus ranked same. This makes the process

more complex because each dataset has the list of almost all the candidates' classifiers as the class labels, which makes the problem of classifier selection as a multi-label learning problem. However, the correct solution has no such strategy to properly address this situations. We simply create multiple cases with the same problem description part (i.e., meta-features list) and different class labels, each for a classifier with same rank. This needs a more sophisticated multilevel analysis of the classifiers by exploiting other characteristics to break the tie situation and also need consideration of multilevel learning.

#### **5.4. Summary**

The proposed study has presented a CBR-based meta-learning and reasoning framework for accurate classifier selection using data characteristics, called meta-features, and classifiers characteristics, called performance metrics. The relationship of data and classifiers characteristics is represented as cases to form a training dataset, called Case-Base, for a CBR classifiers. The recommendations of an accurate classifier for a new case or test dataset is performed using the CBR multi-view, multi-level reasoning methods, developed as part of the proposed framework. In the study, a set of four view of data characteristics are introduced and represented. These are: general characteristics, basic statistical, advanced statistical and information-theoretic families. These families represents the datasets from multiple aspects and are thus a good representative set of characteristics for building a model. The candidate nine decision tree classifiers, considered for this study, are taken from Weka environment with their default settings. In the online recommendation part, the CBR standard methodology is enhanced with accurate similarity functions and a post processing classifier conflict resolution methods to recommend the most appropriate classifier for a given new dataset or learning problem. The methodology is tested on 52 test datasets, taken from UCI/OpenML repositories, which has produced overall accuracy of 94%.

## Chapter 6

### Selection and Design of Rough Set Classifier

---

#### 6.1. Overview

In real-world application scenarios, the nature of data varies from domain to domain and application to application. Usually, the datasets exhibit uncertainty and vagueness in its values, which may result in low performance when some models are built over it. The majority of classification algorithms have not been initially designed for dealing such type of vague and ambiguous data within datasets. In literature, some techniques, especially fuzzy approaches are available that can solve this issue [9, 10], however they depend on several factors. To overcome these issues, this chapter presents the idea of expert's heuristics-based rough set classifier selection among the other available candidate models and designing the classifier to accurately learn a real-world domain problem and provide predication results in a healthcare application of diabetes mellitus. The idea of rough set classifier is based on the well-known rough set theory (RST), initially proposed by Pawlak [7]. The proposed classifier follows the general data mining steps and attempts to enhance certain preprocessing and discretization problems, which are the most common problems in realistic application scenarios. A certain levels of generalization is applied to the methodology of rough set classification model in a way that they are not restricted to the exact problem constraints of the use-case application for which it is proposed. The model is applied in a realistic healthcare scenario with enhancements of guideline-enabled dataset preparation and semantics-enabled discretization to add more into the semantics-preserved accuracy of the model. For verification of the proposed rough set classifier, diabetes scenario with fifty patients' data is used. First, the data is transformed into structured format and then rules are mined using the rough set method for prediction of the diabetes types. Additionally, the chapter also describes application specific task, i.e., risk predictions for a diabetic person, in terms of diabetic symptoms. This

risk prediction is implemented using the correlation-based trend analysis techniques that consumes application specific guidelines as knowledge sources. A series of experiments are performed which prove the higher classification accuracy of the proposed rough set classifier as compared to the state-of-the-art classifiers.

#### **6.1.1. Key Contributions**

To our knowledge, the classifiers proposed in literature for healthcare prediction and classification services, as described in Chapter 3, have a number of limitations, which include: (1) neither of the classifier present classification task for both type-1 diabetes mellitus (T1DM) and type-2 diabetes mellitus (T2DM), but restricted either to one or the other type; (2) the developed classifiers lack the feature of comprehensibility and understandability in the form of the rules acquired from the data; (3) classifiers are restricted either to prediction tasks or future trend analysis over the structured contents; (4) they lack the competence for handling dimensionality, inconsistencies and vagueness issues of clinical data and providing a semantic-preserved data transformation and discretization mechanism. To overcome these limitations, in this chapter, a rough set classifier is designed and implemented that uses experiential and domain knowledge to accurately classify diabetes types. The experiential knowledge is obtained from patients' clinical charts using experts' rigorous inspection method, while domain knowledge is translated from online diabetes guidelines with the help of domain experts who use their expertise during the guidelines translation process. The experiential knowledge is first mined for prediction rules, using rough set' LEM2 algorithm to build a rough set classifier. These rules forms the classifier and used for classification services. Domain knowledge is used to assist physicians in predicting future trend of risky observations and enabling them for prognosis services. The key contributions made through the design of rough set classifier are summarized as follows.

- Extraction of experiential knowledge, from semi-structured clinical charts format using SOAP (Subjective, Objective, Assessment and Plan)-based data representation protocol and experts' rigorous inspection method.

- Unstructured guidelines translation using experts' rigorous inspection method to produce knowledge rules for assisting experts in future trend analysis and prediction of potential risky behaviors that result in accurate decisions.
- Presenting a new semantics-preserved guideline-enabled discretization scheme to transform continuous values of the data features into discrete format in a way the accuracy is kept preserved.
- Mining understandable and self-explanatory prediction rules from high dimensional, inconsistent, and vague clinical data by using a number of novel methods for missing values completion, guideline-enabled discretization and LEM2 algorithm-based features selection and rules extraction.
- Generating services for predicting types of diabetes (*i.e.*, T1DM, T2DM) integrated with future trend analysis of risky observations or behaviors that support experts in prognosis services.

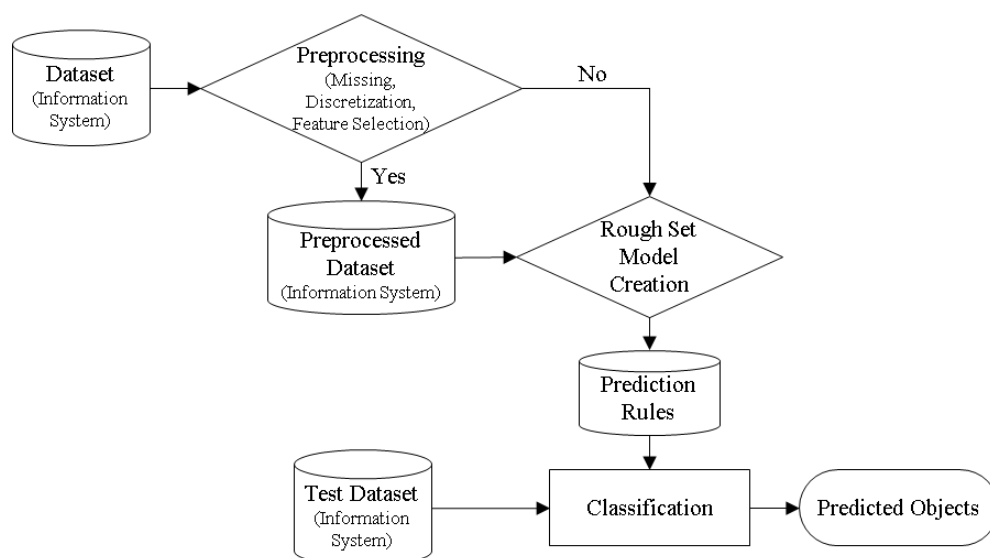
The rest of the chapter is structured as follows: Section 6.2 describes the rationales behind the selection of the rough set classifier, Section 6.3 describes how rough set classifier is designed and Section 6.4 describes the design and development methodology for the accurate rough set classifier that has other modules integrated together, collectively termed as H2RM model. Section 6.5 describes experiments and evaluation of the proposed model. Section 6.6 and Section 6.7 enlists limitations of the proposed idea and concludes the work done, respectively.

## **6.2. Heuristics-based selection of the rough sets classifier**

The domains where semantics, interpretability and accuracy of the final decisions and the model itself are required, comprehensible and accurate classifiers are favored over the non-comprehensible classifiers. Similarly, healthcare and medical applications/datasets are high dimension [154] and usually contain incomplete values [155], which domain experts either consider default values or less essential to be recorded. This makes the data inconsistent and vague in nature. Rough sets theory (RST) [7] has powerful nature of analyzing and handling vague and uncertain

information in efficient problems [156]. The uncertainty issue of data is resolved by the lower and upper approximations methods of the RST which better estimate the vague boundaries of rough sets data [157]. Unlike the traditional classifiers, rough set classifier does not require additional factors and parameter while building the classification model. Rough set classifier is one of those classifiers which can best handle such situations, hence it is the most favored choice of the domain experts in their applications. Based on the above key characteristics of the rough set classifier, it is selected for building a classification and prediction model for the application use-case, considered in this study, i.e., diabetes mellitus prediction and classification.

In the rest of the chapter, an accurate semantics-preserved rough set classifier is designed with enhancements in some of the steps of the classical rough set classifier. A general design of the rough set classifier is shown in Figure 6.1 that can be customized for specific application scenario, based on the requirements of the application in-hand.



**Figure 6.1.** Rough set classification and prediction

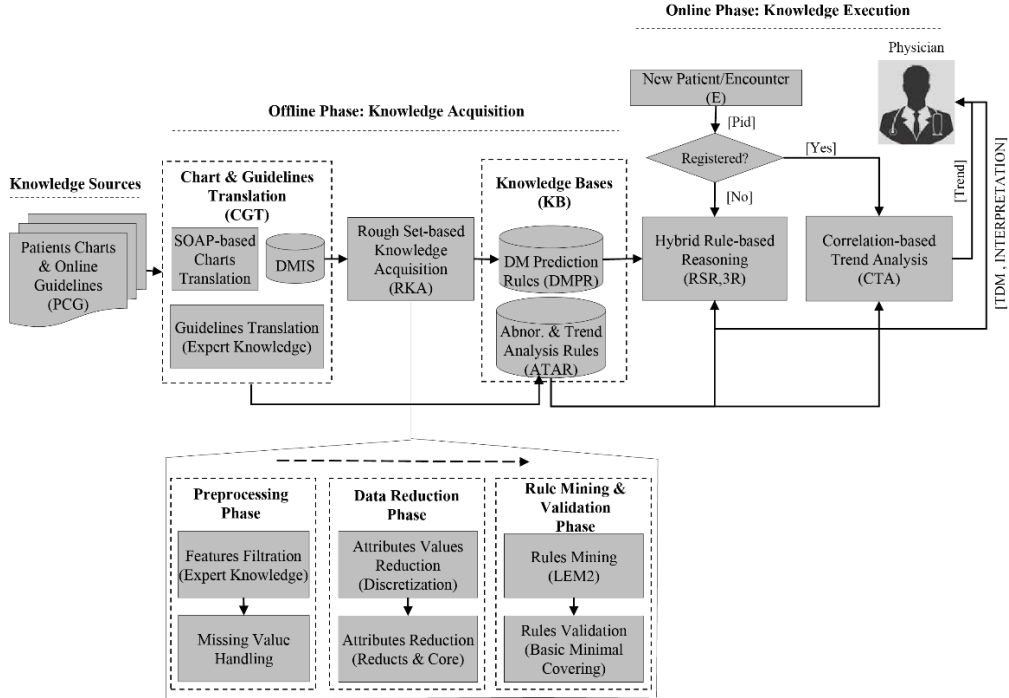
In general, rough set classification process is similar to other data mining processes, therefore the major steps the design of this classifier remains the same. These design

steps include: dataset preparation, preprocessing, mining or discovering knowledge rules (i.e., model creation) and classifying new test instances. Rough set classifier uses rough set theory (RST) [11], which uses a formalism for representing and analyzing data in a specific structured format called information system. The methods used at some steps of the rough set classifier design are different from those of the traditional classifiers generation. The consequent sections describe each step of the rough set classifier design with the help of diabetes scenario and the proposed enhancements are highlighted.

### **6.3. Design of rough sets classifier**

This chapter presents the rough sets classification model applied to the diabetes use case. A hybrid rough set reasoning model (H2RM) that works as a rough set classifier is designed, implemented and tested for predicting type-1 diabetes mellitus (T1DM) and type-2 diabetes mellitus (T2DM). In H2RM, a set of RS prediction rules are mined from 50 diabetes patients' data (collected during 2008-2011) acquired from a local hospital that records patient's encounters in clinical charts following SOAP-based protocol. For management, including finding abnormalities identification and predicting future trend, online diabetes guidelines are translated to simple reference range rules which assist physicians in their decisions and prognosis. To support these functionalities, the rough set classifier is designed using the following components: patients' charts & online guidelines (PCG) as knowledge sources, charts and guidelines translation (CGT), rough set-based knowledge acquisition (RKA), knowledge bases (KBs), hybrid rule-based reasoning (HRBR) and correlation-based trend analysis (CTA), as shown in Figure 6.2.





**Figure 6.2.** Hybrid rough set classification model (H2RM) for prediction

The order followed by the proposed model is: PCG + CGT + RKA + KBs + HRBR + CTA. Abstractly, this model can be represented as a 6-tuple:  $\langle \text{PCG}, \text{CGT}, \text{RKA}, \text{KBs}, \text{HRBR}, \text{CTA} \rangle$ , where:

- PCG (patients' charts and online guidelines): set of patients' clinical charts, which are recorded by physicians during the patients' visits to hospital, and online diabetes guidelines for managing patients' abnormalities in observations and trend analysis. These constitutes knowledge sources for the diabetes prediction and management.
- CGT (charts and guidelines translation): set of methods and procedures used to translate clinical charts and online guidelines to structured data format and reference range rules, respectively. For charts, SOAP-based protocol is used to transform data into structured dataset, while for guidelines translation expert knowledge is used.
- RKA (rough set-based knowledge acquisition): integrates a set of artificial intelligence and mathematical techniques, comprising discretization of

continuous values attributes to discrete values, reducts generation (RG) for selecting essential attributes and LEM2 algorithm for rules mining.

- KBs (knowledge bases): repositories of rough set rules to predict T1DM and T2DM and guideline rules to identify abnormal observations and predict future trends. The rules are represented as production rules.
- HRBR (hybrid rule-based reasoning): rule-based reasoning methodology that implements rough set rules for prediction of T1DM and T2DM and guidelines rules for future prediction.
- CTA (correlation-based trend analysis): a set of statistical methods, such as regression analysis and trend analysis to identify abnormal observations and predict future trends for prognosis service to help physicians in assistance.

The proposed model works in two phases, offline phase and online phase. The offline phase is focused on data preparation (structuring) from external knowledge sources that are presented in unstructured clinical charts and online diabetes guidelines (i.e., PCG) and acquiring knowledge from these sources. The knowledge acquisition composed of manual and automatic procedures. In manual process, first, patients' clinical charts are transformed to structured form called diabetes mellitus information system (DMIS<sup>4</sup>) and guidelines to abnormalities and trend analysis rules (ATAR). In automatic acquisition, a set of rough set-based knowledge acquisition techniques are used to mine DM prediction rules (DMPR) from the DMIS. The rules are stored in knowledge bases that are used in the online process. The online phase is the live or execution phase of the model that delivers prediction and management services to physicians for supporting them in decision making. This phase is activated by the arrival of a patient either a new for diagnosis or the registered one for follow-up. In case, the patient gets registered for the first time, HRBR methodology is triggered. In HRBR, rough set reasoning (RSR) diagnoses and predicts type of diabetics and the reference range reasoning (3R) reasons for abnormalities in observations. In the case patient is registered, the only part of online phase to activate is CTA. At this point,

---

<sup>4</sup> RST uses a formalism that represents and analyses data in its specific format that is described in a structured form called information system, therefore we named our dataset as DMIS.

physician is assisted to look into the previous history of the patient encounters in a consolidated way and see abnormal patterns along with observations trends in future. The physician is in position to see the future and take preventive measures.

## **6.4. Methodology of Rough Set Classifier Design and Development**

The complete design methodology of the proposed rough set classifier, integrated with the correlation-based prediction module of the so called H2RM model, is described in the rest of the chapter.

### **6.4.1. Patient Charts and Online Guidelines**

Data of 50 diabetes patients, 20 with type-1 and 30 with type-2 is acquired from a local hospital that records patients observations in clinical charts, following SOAP (Subjective, Objective, Assessment and Plan)-based data representation protocol [158]. In hospital, data is collected over the period of four years from 2008 to 2011 with average eight encounters for each patient. The minimum number of encounters recorded for a patient are two and maximum are eighteen. In the charts, patients' information containing physiological data, clinical laboratory tests findings, diagnosis information and recommendations are recorded in Subjectivity, Objectivity, Assessment, and Planning sections. In all the charts, Subjectivity and Objectivity sections are merged in one section headed with S&O. The Assessment section is put at the top of each encounter and sometime before the Planning section. Different encounters of same patients are recorded in same chart so that to maintain their history in one document. An example of encounter of a T2DM patient's chart is shown in Figure 6.3.

	A	B	C	D	E	F	G	H	I
1	A: Type 2 DM without complications(M/58)								
2	Outpatient record-Freetext(JCI)								
3	[Outpatient]Date:2011-09-21 Department: Endocrinology Doctor name:XYZ [Revisit]								
4	Treatment Date : 2011-09-21 19:56								
5	Pain								
6	Pain : Non(0)								
7	S&O								
8	- 118/79 mmHg - 93 times /min								
9	(2011-09-21)								
10	- fasting Glucose = 81								
11	Postprandial blood glucose = 269								
12	HbA1C = 7.5%								
13	- TC/TG/HDL/LDL = 210/138/50/122								
14	- AST/ALT = 24/18								
15	A								
16	- Main Type 2 DM without complications								
17	P								
18	#1. Diet & Exercise								
19	#2. Repeat medication & Lab. follow up.								
20	#3. OPD follow up 2 months later								
21	( Consider Dyslipidemia )								
22	Smart Care								
23	First written : XYZ 2011-09-21 13:47 Last written : XYZ 2011-09-21 13:47								

**Figure 6.3.** Clinical encounter of patient in SOAP protocol format.

A number of inconsistencies were there in the charts, such as naming variations, incomplete values, and miss-placement of observations etc., in chart of the patient.

Similarly, to assist physician in automatic abnormalities identification in observations and predicting future trends, online diabetes guidelines are identified for rules creation. The most important predictors in diabetes prediction are BMI, blood pressure, fasting blood glucose, HbA1c, lipids, and liver function tests (LFT), therefore online guidelines associated with these predictors are searched with experts support and listed in Table 6.1.

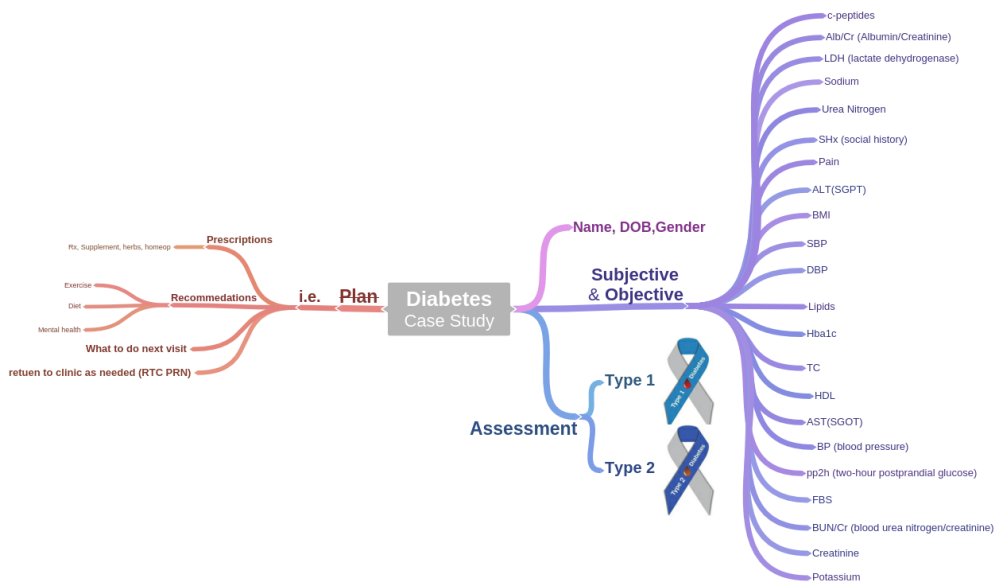
**Table 6.1.** List of guidelines used for managing diabetes mellitus

S.No	Predictor	Guidelines	References
1	BMI	WHO: BMI classification	WHO [36]
2	BP: SBP, DBP	JNC 7 report, AHA	JNC [37–39]
3	FBS	American Diabetes Association. Diabetes Care	ADA [40,41]
4	HBA1c	American Diabetes Association, NICE	ADA [40], NICE [42,43]

5	Lipids: TC, TG, HDL, LDL	NCEP, ADA	NCEP [44], ADA [45]
6	LFT: ALT, AST	Liver disease (LD), Mayo Clinic	LD [46], Mayo Clinic [47]

### 6.4.2. Charts and Guidelines Translation

We have carefully analyzed unstructured charts and manually parsed all the observations to transform into structured format. The list of observations, extracted from charts is shown in right-hand side of Figure 6.4. As we have restricted this research only to prediction and management of diabetes in terms of abnormalities identifications and trend analysis, therefore the plan part of chart is not considered.



**Figure 6.4.** Distribution of patient's observations in clinical chart.

A structured schema is created for observations of the patient recorded in the clinical charts. The schema records the following observations: PID, encounter ID, height, weight, waist, BMI, FHx (family history), SHx (social history), Gender, Age, TDM (type of diabetes mellitus), Complication, Pain, BP (blood pressure), Symptoms, pp2h (two-hour postprandial glucose), FBS (fasting blood glucose), Hba1c (glycosylated hemoglobin), diabetes history, hypoglycemia, Lipids, BUN/Cr (blood urea nitrogen/creatinine), AST/ALT, Urea Nitrogen, Creatinine, Sodium, Potassium,

LDH (lactate dehydrogenase), Alb/Cr (Albumin/Creatinine), and c-peptides for each patient. Each encounter is translated to a record in the schema (i.e., DMIS). For each patient, all encounters are parsed and added into the dataset. The total number of encounters recorded are 391 with distribution of 113 for T1DM and 278 for T2DM.

As a number of attributes have incomplete values, therefore they are dropped from DMIS. The criterion used is that the attributes with missing values  $\geq 20\%$  are most likely produce miss-leading results, therefore they are filtered out. Similarly, we have split BP attribute to SBP (systolic blood pressure) and DBP (diastolic blood pressure) and lipids to its four constituents TC (Total cholesterol), TG (Triglycerides), HDL (High-density lipoprotein), and LDL (Low-density lipoprotein). Liver function tests, AST/ALT are split into AST and ALT. the final output of SOAP-based charts translation is a computer processable dataset, i.e, DMIS.

Similarly, the guidelines listed in Table 6.1 are translated to simple reference rules that defines normal and abnormal reference ranges of values of BMI, blood pressure, glucose, glycosylated hemoglobin, lipids, AST, and ALT attributes. These are shown in Table 6.2a-k.

**Table 6.2.** Set of reference range rules.

<b>(a) BMI</b>		<b>(b) TC</b>	
<b>Interval (Condition)</b>	<b>Interpretation (Decision)</b>	<b>Interval (Condition)</b>	<b>Interpretation (Decision)</b>
$(-\infty, 18.5)$	underweight	$(-\infty, 200)$	desirable
$[18.5, 24.9]$	normal	$[200, 239]$	borderline high
$[25, 30)$	overweight	$[240, \infty)$	high
$[30, \infty)$	obese		
<b>(c) SBP</b>		<b>(d) TG</b>	
<b>Interval (Condition)</b>	<b>Interpretation (Decision)</b>	<b>Interval (Condition)</b>	<b>Interpretation (Decision)</b>
$(-\infty, 120)$	normal	$(-\infty, 150)$	normal
$[120, 139]$	prehypertension	$[150, 199]$	borderline-high
$[140, 159]$	hypertension stage 1	$[200, 499]$	high
$[160, 180]$	hypertension stage 2	$[500, \infty)$	very high
$[181, \infty)$	hypertensive crisis		
<b>(e) DBP</b>		<b>(f) LDL</b>	

<b>Interval (Condition)</b>	<b>Interpretation (Decision)</b>	<b>Interval (Condition)</b>	<b>Interpretation (Decision)</b>
$(-\infty, 80)$	normal	$(-\infty, 100)$	optimal
$[80, 89]$	prehypertension	$[100, 129]$	near or above optimal
$[90, 99]$	hypertension stage 1	$(129, 159]$	borderline high
$[100, 110]$	hypertension stage 2	$(159, 189]$	high
$(110, \infty)$	hypertensive crisis	$(189, \infty)$	very high
<b>(g) FBS</b>		<b>(h) HDL</b>	
<b>Interval (Condition)</b>	<b>Interpretation (Decision)</b>	<b>Interval (Condition)</b>	<b>Interpretation (Decision)</b>
$(-\infty, 70)$	hypoglycemia	$(-\infty, 40)$	low
$[70, 99]$	normal	$[40, 60)$	normal
$(99, 126]$	pre-diabetic	$[60, \infty)$	high
$(126, \infty)$	diabetic		
<b>(i) HbA1c</b>		<b>(j) AST (SGOT)</b>	
<b>Interval (Condition)</b>	<b>Interpretation (Decision)</b>	<b>Interval (Condition)</b>	<b>Interpretation (Decision)</b>
$[4, 5.9]$	hypoglycemia	$(-\infty, 5)$	low
$(5.9, 6.4]$	prediabetes	$[5, 40]$	normal
$(6.4, 7.4]$	diabetes	$(40, \infty)$	high
$(7.4, \infty)$	diabetes with Higher risk		
<b>(k) ALT (SGPT)</b>			
<b>Interval (Condition)</b>	<b>Interpretation (Decision)</b>		
$(-\infty, 7)$	low		
$[7, 56]$	normal		
$[57, \infty)$	high		

Legend: “[” or “]” means inclusive, “(” or “)” means exclusive, “ $\infty$ ” means  $\pm$  infinity.

After creation of these rules, they are stored in knowledge base under the abnormal and trend analysis rules repository, ATAR, to be used in later live execution process. In these tables, column # 2 shows code of the intervals in discrete format that are used for discretization process, Section 6.4.3.2.

### 6.4.3. Rough Set-based Knowledge Acquisition

The translated diabetes dataset, DMIS, contains 391 instances for T1DM and T2DM as the training data for mining prediction rules to predict diabetes for new patient. Generally, clinical datasets are high dimensional [154] and usually contains incomplete values [155] which physicians either consider default values or less essential to be recorded. This makes the data inconsistent and vague in nature. To cope with these situations, we adopt well-known RST, initially proposed by Pawlak [7, 8]. We mine prediction rules from the diabetes data using techniques supported by RST. Our choice of RST is due to its powerful nature of analyzing and handling vague and uncertain information in classification problems[156]. RST uses a formalism that represents and analyses data in its specific format that is described in a structured form called information system, therefore we named our dataset as DMIS. The lower and upper approximations concepts of RST help to solve the problems of vagueness, uncertainty, and incompleteness data [157]. During this process, the training instances are partitioned into equivalence classes [159], upper approximation and lower approximation. The proposed RKA model includes the following phases, such as preprocessing, data reduction and rules creation. These phases work in a sequential flow, as shown in Figure 6.2.

#### **6.4.3.1. Preprocessing Phase**

In dataset, sometimes attributes contain redundant information which need to filter out using expert knowledge to get the list of essential attributes. In our case, we first use expert knowledge to pre-select attributes essential for rough set information system. For example, the calculated attribute BMI is selected and its ingredients, height and weight, are dropped to avoid duplications. Similarly, the attributes ‘past history’ and pain are dropped because their values are mostly same throughout the dataset. FHx and SHx are filtered based on their proportion of missing values (>50%) to the available. The list of essential attributes obtained after applying experts’ knowledge includes: Gender, Age, BMI, SBP, DBP, FBS, Hba1c, TC, TG, HDL, LDL, AST, ALT, as condition attributes and TDM as class attribute.



The filtered dataset with essential attributes still contains incomplete values for attributes SBP, DBP, FBS, Hba1c, TC, TG, HDL, LDL, AST, and ALT with 6%, 6%, 4%, 1%, 9%, 9%, 9%, 15%, 19% and 19%, respectively. In these cases, the patient level criteria become valid and their corresponding strategies are applied. The description of these criteria and the associated strategies are described in Table 6.3. The average and frequent values strategies are the most frequently used techniques that are applied to numeric and nominal value attributes [160]. The experiments for this strategy are performed within Rapid Miner [161]. Similarly, if only two values are missing in an attribute of the encounters of a patient then immediate previous/next value strategy is used. In this case, either  $E_{n-1}$  or  $E_{n+1}$  encounter value is used, depending on the position of missing value that either appears in consecutive or non-consecutive encounters. If values are missing in two consecutive encounters, one is filled with  $E_{n-1}$  and the other with  $E_{n+1}$ . The rational of this strategy is that physicians usually do not record values if they see no change in the observation of a patient. Therefore, either preceding or proceeding value can be the best candidate for the missing value.

**Table 6.3.** Missing value treatment, criteria and strategies, applied to the diabetes mellitus dataset.

Scope	Criteria	Strategy
Dataset level (whole population)	If any attribute of the dataset has missing values in 20% or more than 20% records of the whole dataset	Drop the attributes from the dataset, this may leads to incorrect results
Patient level (whole encounters of one patient)	If any attribute has missing values in 2 or less than 2 encounters of a patient	<p>Use immediate previous/next encounter's values of the same patient</p> <ul style="list-style-type: none"> <li>• Immediate previous/next encounter value, if missing values are non-consecutive</li> <li>• Immediate previous encounter value for the first missing value and immediate next value for the second missing value, if missing values are consecutive</li> </ul>

Scope	Criteria	Strategy
	If any attribute has missing values in less than 20% of the encounters of a patient	Use average/frequent value strategy within encounters of the same patient <ul style="list-style-type: none"> <li>• Compute average of all the values of that attribute for the same patient, if attribute is numeric</li> <li>• Compute frequent value within all the encounters of the same patient, if the attribute is nominal</li> </ul>
	If any attribute has missing values in more than 20% of the encounters of a patient	Use average/frequent value strategy within patients of the same class <ul style="list-style-type: none"> <li>• Compute average of all the values of all the patients in the same class, if attribute is numeric</li> <li>• Compute frequent value within all the patients of the same class, if the attribute is nominal</li> </ul>

The final preprocessed dataset, with filtered attributes and filled missing values, has the following clinical characteristics, summarized in Table 6.4.

**Table 6.4.** Clinical characteristics of the diabetes patients.

Characteristic	Average	Min. Value	Max. Value	Std. Deviation
BMI	23.0	16.2	32.0	3.2
Gender		M (256), F (135)		
Age	48.8	20.0	85.0	15.4
SBP	120.8	89.0	190.0	14.9
DBP	74.5	45.0	115.0	10.2
FBS	137.6	49.0	394.0	43.9
Hba1c	8.0	4.2	14.6	2.0
TC	169.5	0.0	371.0	37.7
TG	101.0	18.0	634.0	80.9
HDL	64.5	31.0	196.0	23.7
LDL	82.2	15.0	180.0	29.4
AST (SGOT)	22.0	11.0	65.0	7.8
ALT (SGPT)	26.6	8.0	120.0	18.0
TDM		T2DM (278), T1DM (113)		

#### 6.4.3.2. Data Reduction Phase

Clinical data have continuous values that randomly vary. If these values are used in its original form for mining rules then rough set will extract huge number of rules

which are intractable [162]. Therefore, all continuous values attributes (e.g., except gender and TDM) are first need to abstract to finite number of intervals [163] and then apply rules mining process. Traditional rough set theory uses different types of discretization methods [163], which define discrete intervals without taking domain knowledge into account. These methods use statistical, entropy, genetic algorithms, fuzzy set theory and Boolean reasoning approaches to split continuous values into discrete intervals [163]. However, none of these methods use semantics of the values of attributes. In the healthcare domain and service generation, semantics of medical data values have significant importance. For example, in the case of our diabetes dataset, continuous values of the SBP attribute (measured in mm Hg) give information that the patient is either in normal ( $<120$ ), prehypertension (120–139), hypertension stage 1 (140–159), hypertension stage 2 (160–180), or hypertensive crisis ( $\geq 181$ ) status. Here, it is very important to discretize the continuous values of SBP in a way to retain their semantic categories in the discretized range/interval. If not, then the rules mined, based on these discretized values, will not reflect the correct range or interval of the value. The exiting discretization approaches do not care about such semantics. For example, if we use the well-known Boolean reasoning approach [57], it gives only three intervals for the same SBP attribute in our dataset. These are,  $(SBP < 110)$ ,  $(SBP 110-116)$ ,  $(SBP \geq 117)$ , which do not reflect the real semantic categories of SBP. To overcome this problem, we propose a semantic interval-based discretization scheme that consumes domain knowledge for discretizing continuous values. In the scheme, we first define cut points for discretization using standard reference ranges for each attribute, as shown in Table 6.2. This knowledge makes the intervals and cut-points more meaningful from clinical perspective and results in meaningful rules. The set of cut-points, their corresponding intervals, and the discrete value for each attribute are shown in Table 6.5.

**Table 6.7.** Set of cut-points and corresponding intervals for discretization

Attributes	# Cut-Points: Cut-Points Description	# Intervals: Interval Description	Discrete Value for Interval	Guidelines
BMI	3: 18.5; 25; 30	4: $(-\infty, 18.5)$ , $[18.5, 24.9]$ , $[25, 30)$ , $[30, \infty)$	0, 1, 2, 3	WHO [164]
Gender	NA	NA	NA	-
Age	2: 30; 50	3: $(-\infty, 30)$ , $[30, 50]$ , $(50, \infty)$	0, 1, 2	-
SBP	4: 120; 140; 160; 181	5: $(-\infty, 120)$ , $[120, 139]$ , $[140, 159]$ , $[160, 180]$ , $[181, \infty)$	0, 1, 2, 3, 4	JNC 7 report, AHA [165-167]
DBP	4: 80; 90; 100; 110	5: $(-\infty, 80)$ , $[80, 89]$ , $[90, 99]$ , $[100, 110]$ , $(110, \infty)$	0, 1, 2, 3, 4	JNC 7 report, AHA [165-167]
FBS	3: 70; 99; 126	4: $(-\infty, 70)$ , $[70, 99]$ , $(99, 126]$ , $(126, \infty)$	0, 1, 2, 3	ADA [168, 169]
Hba1c	3: 5.9; 6.4; 7.4	4: $[4, 5.9]$ , $(5.9, 6.4]$ , $(6.4, 7.4]$ , $(7.4, \infty)$	0, 1, 2, 3	ADA [168], NICE [170, 171]
TC	2: 200; 240	3: $(-\infty, 200)$ , $[200, 239]$ , $[240, \infty)$	0, 1, 2	NCPE [172], ADA [173]
TG	3: 150; 200; 500	4: $(-\infty, 150)$ , $[150, 199]$ , $[200, 499]$ , $[500, \infty)$	0, 1, 2, 3	NCEP [172], ADA [173]
HDL	2: 40; 60	3: $(-\infty, 40)$ , $[40, 60)$ , $[60, \infty)$	0, 1, 2	NCEP [44], ADA [45]
LDL	4: 100; 129; 159; 189	5: $(-\infty, 100)$ , $[100, 129]$ , $(129, 159]$ , $(159, 189]$ , $(189, \infty)$	0, 1, 2, 3, 4	NCEP [44], ADA [45]
AST(SGOT)	2: 5; 40	3: $(-\infty, 5)$ , $[5, 40]$ , $(40, \infty)$	0, 1, 2	LD[46], Mayo Clinic [47]
ALT(SGPT)	2: 7; 57	3: $(-\infty, 7)$ , $[7, 56]$ , $[57, \infty)$	0, 1, 2	LD[46], Mayo Clinic [47]

Legend: “[” or “]” means inclusive, “(” or “)” means exclusive, “ $\infty$ ” means  $\pm$  infinity.

After applying discretization process based on the cut-points, we obtained discretized information system (DIS). A partial view of the discretized DIS is presented in Table 6.6.

**Table 8.6.** Partial Information System (training dataset) in interval format after discretization.

DBMI	Gender	DAGE	DSBP	DDBP	DFBS	DHbA1c	DTC	DTG	DHDL	LDL	DAS	DAL	TDM
[18.5,24.9]	M	(50, $\infty$ ]	[120, 139]	[ $-\infty$ , 80]	(99,126]	(7.4, $\infty$ ]	[ $-\infty$ , 200]	[ $-\infty$ , 150]	[ $-\infty$ , 40]	[ $-\infty$ , 100]	[5, 40]	[7, 56]	T2DM
[25, 30]	M	[30, 50]	[140, 159]	[100,110]	[70, 99]	(7.4, $\infty$ ]	[ $-\infty$ , 200]	[150,199]	[40, 60]	[ $-\infty$ , 100]	[5, 40]	[7, 56]	T1DM
[18.5,24.9]	F	(50, $\infty$ ]	[ $-\infty$ , 120]	[ $-\infty$ , 80]	(126, $\infty$ ]	(6.4, 7.4]	[200, 239]	[ $-\infty$ , 150]	[40, 60]	[100,129]	[5, 40]	[7, 56]	T2DM
.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.
[ $-\infty$ ,18.5)	F	[30,50]	[ $-\infty$ , 120]	[ $-\infty$ , 80]	(126, $\infty$ ]	(7.4, $\infty$ ]	[ $-\infty$ , 200]	[ $-\infty$ , 150]	[60, $\infty$ ]	[ $-\infty$ , 100]	[5, 40]	[7, 56]	T1DM
[25, 30]	F	(50, $\infty$ ]	[ $-\infty$ , 120]	[80, 89]	(99,126]	(5.9,6.4]	[ $-\infty$ , 200]	[ $-\infty$ , 150]	[60, $\infty$ ]	[ $-\infty$ , 100]	[5, 40]	[5, 7, $\infty$ ]	T2DM

Legend: “[” or “]” means inclusive, “(” or “)” means exclusive, “ $\infty$ ” means  $\pm$  infinity.

After attributes values reduction using discretization, the next step is to create reducts, which are feature subsets of attributes of the original information system (i.e., DMIS) that facilitate in the process of rule mining and classifying same dataset with same accuracy [7, 8, 174]. We adopted Lattice Reduct Search method, implemented in Rough Set Data Explorer (ROSE2) system [175, 176] with the default configuration for finding all possible reducts. The set of all possible reducts obtained, are shown in Table 6.7.

**Table 6.9.** List of all possible Reducts after applying reduct operation

Reduct #	# Attributes	Reduct (Attributes)
1	10	{BMI, Gender, Age, SBP, DBP, FBS, HbA1c, HDL, LDL, PT}
2	10	{BMI, Age, SBP, DBP, FBS, HbA1c, TG, HDL, LDL, PT}
3	10	{BMI, Gender, Age, SBP, FBS, HbA1c, HDL, LDL, OT, PT}
4	10	{BMI, Age, SBP, FBS, HbA1c, TG, HDL, LDL, OT, PT}

In all the reducts, total number of participating attributes are 12 and only one attributes TC is not considered in either of the reduct. The frequency of attributes Gender, DBP, TG and OT in all the reducts is 50% while the rest of attributes have

100% participation; which means that they are appearing in all the reducts and therefore very essential.

Like reduct, core is another important concept of RST which comprises only the most relevant attributes in the original information system. If any of the attribute is removed from the core, the accuracy of classification rules built will drastically dropdown, therefore we apply the core generation operation in ROSE2 to get the final key attributes. It is obtained using intersection operation over all the reducts. In our case, the core consists of the features shown in equation 1

$$\text{Core(DIS)} = \text{Intersection (RED(DIS))} = \{\text{BMI, Age, SBP, FBS, Hba1c, HDL, LDI} \quad (1)$$

Prediction accuracy of the original set of attributes and the core attributes was measured. The objective of measuring accuracy is to show effectiveness of the reduced attributes and overall attributes in the original dataset. When measured, core attributes produced 0.9744% accuracy, while the all 13 attributes of the original information system produced 0.9872% accuracy. The total reduction in accuracy is only 0.0138%, which is almost negligible. However, the reduct and core operations of RST reduced the number of attributes by more than one third, which reduce the complexity of building the prediction model.

#### **6.4.3.3. Rules Mining and Validation Phase**

Once the core attributes are selected, the next step is to mine decision rules from the discretized information system for the core attributes using LEM2 algorithm [177]. We have used the basic minimal covering criteria of LEM2 algorithm implemented in ROSE 2 system[176]. The DIS contains 391 instances that are used to mining rules to predict diabetes types. Total 23 rules are mined, out of which one rule is approximate with inconsistent prediction for the same condition attributes. Extracted partial rule set is shown in Table 6.8.

**Table 6.10.** A Partial rules list extracted from discretized information system using LEM2 algorithm

Rule #	Prediction for TDM	Prediction Rule	Significance
1	(T1DM)	(BMI = [18.5, 24.9]) and (Age = (50, $\infty$ )) and (SBP = 120,139]) and (Hba1c = (7.4, $\infty$ )) and (TC = ( $-\infty$ , 200)) and (SGPT = [7, 56])	20 (17.70%)
2	(T2DM)	( Gender = M) and (SBP = ( $-\infty$ , 120)) and (Hba1c = (6.4, 7.4]) and (LDL = [100, 129])	17 (6.12%)
3	(T2DM)	(BMI = [18.5, 24.9]) and (Age = [30, 50]) and (SBP = ( $-\infty$ , 120)) and (TG = ( $-\infty$ , 150)) and (HDL = [40, 60])	23 (8.27%)
4	(T1DM)	(SBP = [120, 139]) and (DBP = [80, 89]) and (Hba1c = (5.9, 6.4]) and (HDL = [40, 60]) and (SGPT = [7, 56])	7 (6.19%)
5 (approx. rule)	(T1DM) OR (T2DM)	(BMI = [18.5, 24.9]) and (Age = (50, $\infty$ )) and (FBS = 3) and (Hba1c = (126, $\infty$ )) and (TG = ( $-\infty$ , 150)) and (LDL = ( $-\infty$ , 100)) and (SGPT = [7, 56])	[5, 5] [2, 3]

Legend: “[” or “]” means inclusive, “(” or “)” means exclusive, “ $\infty$ ” means  $\pm$  infinity.

Table 6.8 shows decision attribute of the rule, ingredients of the rules (i.e., condition attributes with values) and significance value that describes its coverage within its own class. For example, rule 1 has 17.7% significance value in its class T1DM that supports 20 instances of the training information system. After creation of the rules, the whole prediction model is stored in knowledge base within DM prediction rules repository, DMPR, to be used in the live execution process.

Validation of the prediction model (rules extracted using rough set LEM2 method) is performed using 10-foldcross validation approach over the whole diabetes dataset. The details are given in Section 6.5 for evaluating the rules using prediction accuracy.

#### 6.4.4. Hybrid Rule-based Reasoning

Online phase of the proposed H2RM is based on RBR methodology, which internally uses two levels of reasoning in sequential way. In the first level, rough set-based reasoning (RSR) methodology is activated for those patients who are not registered

before. In this process, the RSR engine loads rules from the DMPR repository and executes them on the current observations of the patient. Diabetes type is predicted from the patient's observations and withheld till the second level of reasoning process is not completed. In the second level, reference range-based reasoning (3R) is performed over BMI, SBP, DBP, FBS, Hba1c, TC, TG, HDL, LDL, AST (SGOT), and ALT (SGPT) using reference range rules defined in Table 6.2 to categorize the observations as either normal, borderline, abnormal, risky, *etc.* This automatic categorization of the observations further assist physician in easy understanding and quick decision-making. The final results of HRBR are provided to physicians to assist them in diagnosis and analysis of the patient's current observations. This process is shown in detail in Algorithm 1.

---

**Algorithm 1** Hybrid Rule-based Reasoning (HRBR)

---

**Begin**
**Input:** KB: Knowledge Base, E: Encounter

**Output:** TDM, INTERPRETATION

**ApplyHRBR (E)**, where {E|E is EncounterOfNonRegisteredUser, E: = {Pid, OBS}}, OBS: = {BMI, SBP, DBP, FBS, Hba1c, TC, TG, HDL, LDL, AST(SGOT), ALT(SGPT)}

**A. PerformRSR(E) // Rough Set Reasoning**

[Load Prediction Rules From Knowledge Base]

1. DMPR: = LoadRulesFromKB(RULES that contain TDM as CONC); where CONC: = { T1DM, T2DM}
2. [Execute Rules For Predicting Types of Diabetes] **Foreach** RULE in DMPR
  - a. **Foreach** CA in RULE //CA: = {BMI, Age, SBP, FBS, Hba1c, HDL, LDL, PT}
  - b. **If** CA.values  $\neq$  E.OBS.value  
    **THEN** Try next RULE
  - EndIf**
  - c. TDM := CONC of the RULE;
  - d. **Goto** Step B
  - e. **EndFor**

**EndFor**

3. TDM = Message("UNDEFINED")

**B. Perform3R (E) // Reference Range-based Reasoning**

[Load Reference Range Rules From Knowledge Base]

4. ATAR: =  
    LoadRulesFromKB(RULES that contain INTERPRETATION as CONC); where CONC: = { Table 6.2. INTERPRETATION. Value}
- [Execute Rules For Finding Current Status of Each Observation]
5. **Foreach** RULE in ATAR
  - a. **Foreach** CA in RULE //CA := {BMI, SBP, DBP, FBS, Hba1c, TC, TG, HDL, LDL, AST(SGOT), ALT(SGPT)}
  - b. **If** CA.values  $\neq$  E.OBS.value  
    **THEN** Try next RULE
  - EndIf**
  - c. INTERPRETATION [] := CONC of the RULE;



**Algorithm 1** Hybrid Rule-based Reasoning (HRBR)

---

```

    EndFor
  EndFor
C. PHYSICIAN := ProvideResults (Pid, TDM, INTERPRETATION)
End

```

---

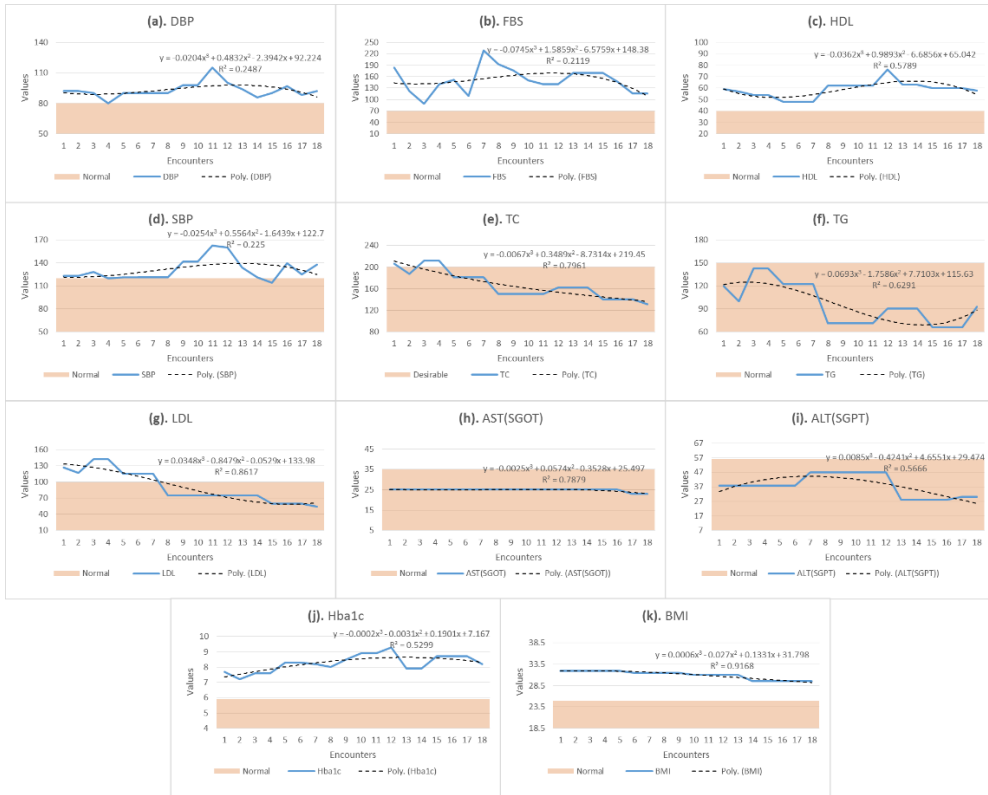
Algorithm 1 has four main functions. These are defined for activation of HRBR, rough set-based reasoning, reference range-based reasoning and final results propagation. When a new patient arrives in the hospital and his observations are recorded, the main function of HRBR, ApplyHRBR (), is activated. This function, called the rough set-based reasoning function, PerformRSR(), for predicting diabetes type. The process of rough set reasoning starts with loading rules from knowledge base using the function, LoadRulesFromKB(). Once rules are loaded, execution of rules starts and final decision is obtained either as T1DM, T2DM or UNDEFINED.

After diabetes prediction, physicians are usually interested in knowing the exact status of the observations of the patient. For this purpose, reference range-based reasoning is activated using the function, Perform3R (). Like rough set reasoning, first, rules are loaded from knowledge base, and then they are executed one-by-one to find out whether the current value for that observation is normal, borderline, risky, etc. Finally, the function ProvideResult() propagates the results of rough set reasoner and reference range reasoned, in integrated from, to physician for further assessment and final decision.

### 6.4.5. Correlation-based trend analysis

In the online phase of H2RM, when a registered patient visits hospital for follow-up and new observations are recorded then physician usually desires to review past history of all encounters of the patient. This is an essential step for them to further analyze the patient's conditions and prescribe medications or provide general wellbeing recommendations or consult patient regarding the next follow-up, etc. Moreover, they are also interested in seeing future trends of the patient's observations, based on the current and past observations, in order to predict future and take preventive measures. However, they are unable to get all these benefits in

the current scheme of clinical charts, where the observations are inconsistent and placed randomly with different naming convention, etc. in excel sheets. The literature listed in this thesis lack the capability of transforming these clinical charts to structured data format and building management and trend analysis services for physician to support them in decision-making. To overcome these shortcomings, and support physicians with comprehensive insights of the past observations of patients, we propose a correlation-based trend analysis technique.



**Figure 6.5.** Correlation-based trend analysis for prognosis

Correlation analysis is one of the important future trends prediction technique applied to numeric data [178], therefore we adopt it in our research for analyzing abnormal trends in patients observations. In the knowledge execution flow of the proposed H2RM, when a registered patient visits hospital for follow-up, his observations are recorded and scattered line graphs is drawn for the current and past observations, as shown in Figure 6.5(a-k) by the bold-face blue line. After the scattered graph, a

correlation-based polynomial trend line, order 3, is added to the graph to predict trend in near future. We also compute residue  $R^2$  value to the trendline to show accuracy of future prediction for new encounters. The selection of polynomial trendline for future prediction is due to the fact that clinical values fluctuate gradually rather than abruptly. Polynomial trendlines with order 3 have two peaks or bottom values in the regression equation. In Figure 6.5(a-k), the dotted black line shows trend line with the regression equation and  $R^2$  values. In the same figure, normal ranges of the observations are reflected with light orange strap that internally uses rules (Table 6.2) extracted from guidelines (Table 6.1).

The proposed CTA provides two insights to physicians from the patient data in the form of abnormalities identifications and future trends that assist them to see all the relevant information in a consolidated form. Figure 6.5 shows detail of all the 18 encounters of a single patient with T2DM.

## **6.5. Experiments and Results**

### **6.5.1. Evaluation Criteria**

To evaluate the proposed hybrid RS reasoning model, a number of evaluation criteria can be used, such as prediction accuracy, precision, recall, F-measure, balanced accuracy and end user (physician in our case) satisfaction, *etc.* [179]. These criteria can be grouped into system-centric (focus on system accuracy, precision, recall, *etc.*) and user-centric (focus on user satisfaction, *etc.*). A good evaluation criterion can be the one taking both system centric and user centric parameters into account. However, in our evaluation, we stick to only the system centric approach due to the prototype implementation of H2RM. We use average accuracy and balanced accuracy evaluation metrics to evaluate the performance of our proposed model. The prediction rules derived by the rough set knowledge acquisition component are used to test data in the diabetes dataset and assess the performance.

### 6.5.2. Experimental Setup

H2RM consists of two main modules: offline knowledge acquisition and online knowledge execution. Therefore, we setup two sets of experiments. The first set is to mine prediction rules from the diabetes dataset and the second one is to provide real time services on top of these rules for new patient/encounter. For both sets, we used ROSE2 software [176] in Windows environment in a PC with specification of Intel Pentium Dual-Core™ (2.5 GHz) and RAM 4GB. For the first set of experiments, setup and detailed description is given in Section 6.4.3.3. The second set of experiments further consists of validation of mined rules and trend analysis of past and current encounters of a patient. Setup for the latter experiment is explained in Section 6.4.5, while for validation of mined rules, we use basic minimal covering technique of the RST with default parameters setting in ROSE 2 system. The default parameter settings are shown in Table 6.9.

**Table 6.11.** Experimental setup for validation of prediction rules in ROSE 2 system.

S.No	Parameters	Values
1	Test	k-fold cross validation
2	Number of passes	10
3	Majority threshold	21%
4	Minimum similarity	50%
5	Partially matched rules	All
6	Rule support	strength $\times$ similarity

### 6.5.3. Results

The results of first set of experiments are described in Section 6.4.3.3. In total, 23 rules are extracted from 391 instances of the dataset. Table 6.8 shows a partial list of the rules along with their significance values. Results of the validation experiment are shown in Table 6.10.

**Table 6.12.** Confusion matrix describing overall output of the validation process.

Type of DM	T1DM	T2DM	None
T1DM	106 (TP)	7 (FN)	0
T2DM	9 (FP)	269 (TN)	0

Table 6.10 shows that 7/113 cases of T1DM are incorrectly predicted as T2DM and 9 T2DM cases are incorrectly predicted as T1DM. There is no such example, either of T1DM or T2DM, in which neither T1DM nor T2DM is predicted. Therefore, the “None” column is zero for both class. Average accuracy (%) of the prediction model and individual accuracies of each class (T1DM, T2DM) are shown in Table 6.11. The average predictive accuracy of the model is 95.91% with 4.09% incorrect predictions. Standard deviation of the percent incorrect predictions, for all the 10-folds of the model is 2.61, while for the individual classes are 6.16 and 4.11, respectively. The individual class level accuracy for class T1DM is 94.59% and for class T2DM is 96.85%.

**Table 6.13.** Average accuracy (%) of the model for individual class and overall model.

Type of DM	Correct	Incorrect	None
T1DM	94.59 $\pm$ 6.16	5.41 $\pm$ 6.16	0.00 $\pm$ 0.00
T2DM	96.85 $\pm$ 4.11	3.15 $\pm$ 4.11	0.00 $\pm$ 0.00
Total	95.91 $\pm$ 2.61	4.09 $\pm$ 2.61	0.00 $\pm$ 0.00

The results show that the predication accuracy for class T2DM is higher than the prediction accuracy of class T1DM. The reason for incorrect prediction of T1DM cases as T2DM and *vice versa* is due to the approximate rule (rule #23) of the prediction model.

To know results in terms of percent accuracy and percent error for each fold, we generate fold-wise test results. Figure 6.6 show the test results for each fold of the 10-fold cross validation process.



Pass Number	Fold Size	Incorrect Examples	Correct Examples	Percent Accuracy	Percent Error
Pass 1	40	1	39	97.5	2.5
Pass 2	39	3	36	92.30769231	7.6923077
Pass 3	39	2	37	94.87179487	5.1282051
Pass 4	39	3	36	92.30769231	7.6923077
Pass 5	39	2	37	94.87179487	5.1282051
Pass 6	39	0	39	100	0
Pass 7	39	1	38	97.43589744	2.5641026
Pass 8	39	2	37	94.87179487	5.1282051
Pass 9	39	1	38	97.43589744	2.5641026
Pass 10	39	1	38	97.43589744	2.5641026
<b>(b) Average Accuracy and Standard Error for 10-Folds</b>					
No. Instances				391	
Total Number of Incorrect Examples				16	
Total Number of Correct Examples				375	
Average Accuracy				95.90384615	
Average Error				4.096153846	
Standard Error based on Percent Error of each Fold				2.61660764	
Average Accuracy $\pm$ Standard Errors				95.9 $\pm$ 2.6	

**Table 6.15.** Evaluation parameters for computing balanced accuracy.

True Positive (TP)	False Positive (FP)	True Negative (TN)	False Negative (FN)
106	9	269	7

The results of balanced accuracy (Equation (2)) and conventional average accuracy (Tables 6.11 and 6.12) are the same, which shows that our predictive model performs equally well on either class (T1DM and T2DM).

The results of the final correlation-based future trend analysis experiment are shown in Figure 6.5. These results assist physicians in assessing patient observations from three perspectives: pattern of past and current observations (blue line graph),

deviation of the observations from normal ranges (light orange strap), and prediction of future trend (dotted black line). A correlation equation of order 3, along with  $R^2$ , show accuracy of future trend prediction for that observation.

#### 6.5.4. Comparison

To compare results of the proposed rough set classifier with state-of-the-art rule-based classifiers, an experiment is performed in Weka Experimenter environment with the default parameters of 10x10 fold-cross validation. Six state-of-the-art classification algorithms, shown in Table 6.14, are selected to generate classification models for each classifier using the same dataset, shown in table 6.14.

**Table 6.16.** Comparison of the rough set classifier with state-of-the-art-classifiers

Classifier	rules.DTNB	rules.JRip	rules.NNge	rules.PART	rules.Ridor	rules.DecisionTable	Rough.Set.LEM2
Average Accuracy	91.31(2.74)	95.13(2.73)	94.16(3.72)	<b>96.16(2.18)</b>	94.88(2.79)	89.52(3.69)	<b>95.9(2.6)</b>

The results shows that rough set classifier performs better than five of the state-of-the-art algorithms even though it was selected based on the experts' heuristics knowledge. The only algorithm that performs better than the proposed model is the rules.PART classifier which has an average accuracy of 96.16%. The increase in accuracy is only 0.16%, which is not significant over the proposed model. For the significance, corrected-paired t-test is applied with a significance level of 0.05, and the results of the test shows that there is no significant change in the results of these two classifiers. This proves that the proposed rough set classification model is the accurate classifier and correct choice of the experts for the considered application.

#### 6.6. Limitations of the proposed rough set classifier

As the results show, the proposed rough set classifier enables the generation of semantics-preserved accurate predictions on real-world applications data, however it has a number of limitations as well. These limitations are enlisted as follows.



1. The proposed rough set classifier is used in the diabetes domain utilizing data from the patients' clinical charts. Though, SOAP-based protocol provides a semi-structured format for structuring data, however it requires experts' involvement and rigorous inspection method to transform data into structured format. There is no support of an automatic data extraction from the patient's charts, which is one of the limitations of this study.
2. In this study, a semantics-preserved guideline-enabled discretization method is proposed to discretize continuous data in a way the semantics remains intact. However, the proposed discretization scheme can be easily used in situations where the number of intervals of the continuous values of the attributes are explicitly known in advance and are limited in number as well. If either the intervals are not known in advance, from the domain knowledge, or the number of intervals itself is huge then the proposed scheme will not be efficient with respect to time taken for the transformation of data to discrete format.
3. Approximate rules generated by the rough set classifier needs further processing using domain knowledge to reach accurate and more correct decision without any ambiguity. However, the proposed methodology lacks methodology for handling such rules.

## **6.7. Summary**

In this chapter, a rough set classifier designed and presented. The selection of the rough set classifier is done based on the experts' heuristics knowledge about the real world application scenario of the diabetes mellitus. Its selection is done based on its high interpretability and understandability qualities of the rules in the model and their accuracy on inconsistent and vague data. First, a standard guideline-enabled rule-based approach is used to prepare the dataset (information system) from the set of semi-structured clinical notes using the SOAP-based protocol. Guidelines are translated using experts' rigorous inspection method into simple rules that are used during the preprocessing stage of the training dataset and in the application specific

services generation. The default rough set discretization scheme lacks the capability to transforming data correctly into the discrete form, therefore the chapter discussed a semantics-enabled discretization scheme that utilized the guideline knowledge for deciding the correct cut-points and the associated intervals, which are used during the discretization process. Comprehensible rules, represented in if-then form, are mined, i.e., rough set classification model is built using the classical rough set theory. The semantics-enabled discretization scheme has empowered the classifier to keep the semantics preserved in the rules during the induction phase. The proposed model is evaluated and compared with state-of-the-art methods on the diabetes dataset and the results are generated. The results shows that the rough set model outperforms the compared methods and produces average and balanced accuracy of 0.95% on a test dataset of 391 records of type-1 and type-2 diabetes.

## Chapter 7

### **Selection and Design of Hybrid-CBR Classifier**

---

#### **7.1. Overview**

In real-world applications, where more precise and accurate classification results are required over small set of data instances with a large number of classes, the traditional classifiers cannot be easily used to model the problem and build an accurate classifier for it. In such situations, instance learning methods can perform well if they are designed properly. The reason is that in rule-based classification, as discussed in last chapter, the classifiers are built based on the concept of generalization rather than specialization, where generalized heuristics rules are extracted from the dataset. These classifiers represent similar sets of instances of the dataset in general rules and ignore specificity of the data. The rules have minimum support to generate classification decisions closer to the users' specific requirements. The classification mechanism of the rule-based classifiers and the associated systems are based on the principal of exact pattern matching. If all the patterns of a new test case are exactly matched against any of the existing rules in the classification model, the corresponding class label is predicted as the decision, otherwise the input case is declared as unclassified even if it is too close to any of the existing rule in the model. In real-world application domains, it is very essential requirement to predict the closet solution for the new case, even if the exact match is not found. This feature of the applications increases the acceptability of such systems in real-world domains. To address the same issues, this chapter presents the idea of a hybrid case-based reasoning (hybrid-CBR) classifier and tests its methodology in a real setup of a wellness application scenario where physical activity recommendations are supposed to generate as close as to the specific requirements of an individual. The proposed hybrid-CBR classifier is first designed and then implemented and evaluated in a real-world project Mining Minds (MM) platform for health and wellness services. State-

of-the-art recommendation models in this domain are based on the rule-based classifiers and results they generate are highly generalized, based on the general guideline rules from wellness domain. In the proposed hybrid-CBR classifier, guidelines-enabled rule-based methodology is used to create the resolved cases for the CBR model and also for the real new input case preparation during the classification and recommendation generation phase. So, this chapter first describes the knowledge acquisition process from the guidelines for the case-base creation and for the rule-based classification and recommendation systems to be compared with the proposed hybrid-CBR model. Then the design and implementation methodology of the proposed hybrid-CBR classifier is presented, where accurate similarity functions (local and global) are defined for accurately matching new input case against the existing resolved cases. At the end, the proposed methodology is tested against a newly created test case base and the results are compared with two rule-based classifiers called baseline rule-based reasoning (baseline-RBR) classifier and modified rule-based reasoning (modified-RBR) classifier. Similarly, the newly designed similarity functions for accurate case retrieval are compared with the existing jCOLLIBRI<sup>5</sup> system's similarity functions and the results are generated, which show that the proposed hybrid-CBR classifier performs significantly better than the state-of-the-art methods.

As the hybrid-CBR model integrates two reasoning methodologies (rule-based reasoning and case-based reasoning) in addition to a third preference-based reasoning (optional and application specific in this case), therefore in this chapter the methodology is generally referred as multimodal hybrid reasoning (HRM). These terminologies will be carried out throughout this chapter.

#### **7.1.1. Key Contributions of Hybrid-CBR Classifier**

In Chapter 3, a summary of hybrid classifiers, used in wellness applications, is presented and the shortcomings are highlighted. To overcome those limitations, the

---

<sup>5</sup> <http://gaia.fdi.ucm.es/research/colibri/jcolibri>

proposed HRM, especially the hybrid-CBR classifier, is accurately designed and implemented. The key contributions made through the design of hybrid-CBR are summarized as follows.

- (i) Design of a flexible multimodal hybrid reasoning framework to support implementation of accurate classifiers for generating precise classification with the focus on specialization rather than generalization.
- (ii) Acquisition and translation of implicit and explicit knowledge, using experts' rigorous inspection and guidelines-enabled rule-based method.
- (iii) Efficient exploitation of the acquired knowledge for personalized decision making using the integration of multiple reasoning methodologies, such as RBR, CBR and preference-based reasoning (PBR), deployed in a linear combination.
- (iv) Design and creation of a case-base of successful physical activity recommendation cases using an accurate guideline-enabled rule-based case preparation methodology. This process requires experts' rigorous inspection of the prepared cases and the knowledge required for the creation of the cases.
- (v) Definition of new accurate local and global similarity functions for the proposed hybrid-CBR classifier to enhance performance of the retrieval phase of the traditional CBR classifier.
- (vi) Reducing the bottlenecks of traditional single reasoning methodologies, which exploit only single knowledge sources for generating a single service at a time.

## **7.2. Selection and Design of Hybrid-CBR Classifier**

Chapter 3 has provided a detailed critical analysis of state-of-the-art reasoning and recommendation methods, used for different applications in healthcare and wellness domains, which use a variety of design strategies for the integration and building a classifier. Based on the surveyed literature, this chapter presents a unique methodology of case-based reasoning (CBR) classifier design and implementation

along with its integration with a guidelines-enabled rule-based reasoning approach to achieve the objective of accurate and precise recommendation decisions.

### **7.2.1. Rationales behind the selection of hybrid-CBR**

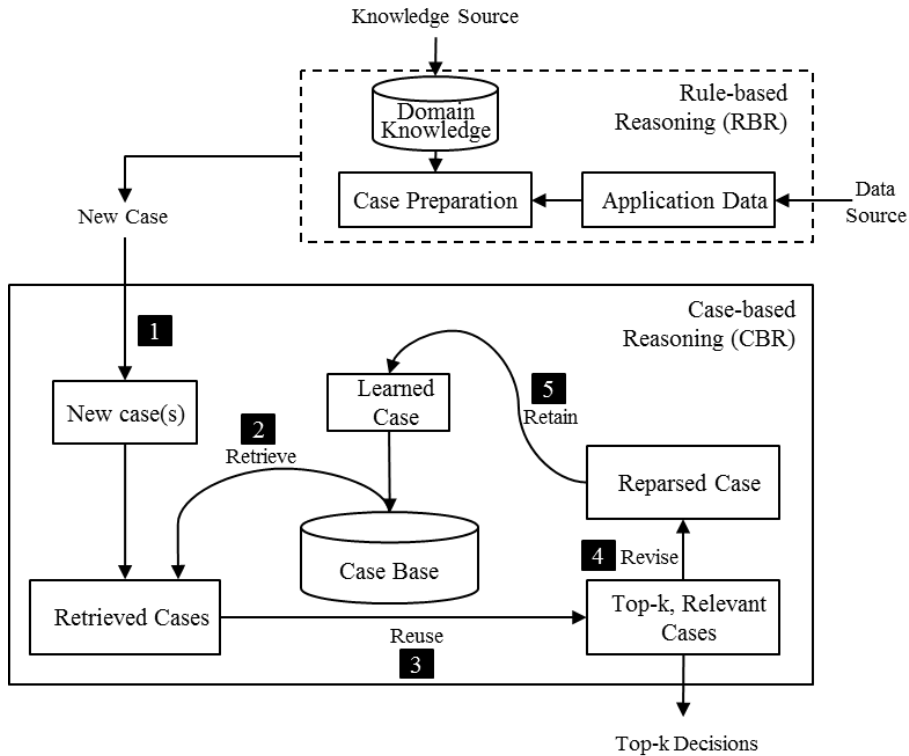
In applications scenarios, where precise classification decisions are required over small set of data instances, i.e., small datasets, that have large number of classes, then traditional classifiers cannot perform well because they cannot generalize the small number of instances in the form of a model. Similarly, if the exact match is not available for the new cases in the already built model, then the result will be unsuccessful for traditional classifiers. Moreover, with passage of time, the number of new cases increases and the model needs to be updated to improve the performance, however it cannot be done easily. In similar situations, instance-based learning methods, such as CBR and k-NN, etc., perform well, therefore the CBR classifier is heuristically selected in this study for generating accurate classification results of physical activity recommendations to individuals, specific to their requirements. The classical CBR classifier is integrated with RBR methodology with a certain level of generalization to produce a generalized hybrid-CBR classifier for generating accurate decisions in other domains as well. The rationales behind the selection of hybrid-CBR are summarized as follows.

- (i) Supports accurate and precise classification results in case of small datasets and datasets with high dimensionality of classes.
- (ii) Works on the basis of similarity score rather than exact matching, therefor always returns relevant decisions (i.e., resolved cases) that can equally likely be applicable to the new query case.
- (iii) Supports incremental learning by using the revise and retain steps of the classical CBR methodology. This improves quality of the model with the passage of time without relearning or training of the classifier.

Based on the above listed rationales, hybrid-CBR classifier is selected for the real-world application of wellness recommendations generation, considered in this chapter.

### 7.2.2. Design of hybrid-CBR classifier

The design methodology of hybrid-CBR classifier is shown in Figure 7.1.



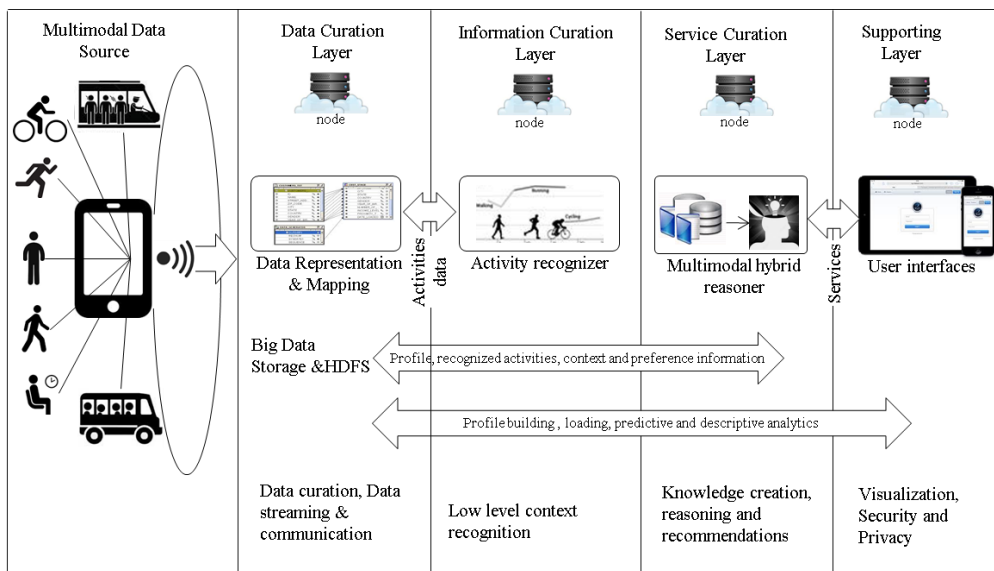
**Figure 7.1.** Hybrid case-based reasoning classifier

In the design of CBR classifier, the essential but most challenging task is the creation of accurate train test cases, called case base. Hence, in the proposed hybrid-CBR model, an accurate rule-based, case preparation methodology is proposed along with accurate local and global similarity functions. For the preparation of successful cases of the case base, guidelines-enabled case creation is proposed. The idea is that as the successful cases are the knowledge of a CBR system, used for resolving new unresolved cases, therefore they must be carefully prepared from the domain guidelines. Once these cases are created, accurate case retrieval function need to be defined. So this chapter present the process of creating accurate case similarity functions. The proposed hybrid-CBR classifier is tested and evaluated in a real-world

application scenario of physical activity recommendation and showed significant improvements in performance with respect to state-of-the-art methods. The rule-based reasoning part of the whole HRM model is used for creation of new input case, unresolved case, from the domain data and the domain knowledge, at run time. This new input case is passed to the CBR classifier, where the case-based reasoning process utilizes the retrieval functions and recommends top-k most relevant cases as the proposed decisions.

### 7.3. A real-world application scenario

The proposed hybrid-CBR model is developed as a part of multimodal hybrid reasoner in a real-world application project called Mining Minds (MM) platform [182, 183], as shown in Figure 7.2. A brief overview of the MM platform is first provided, here, before going to the technical details.



**Figure 7.2.** Abstract view of the real-world application Mining Minds

The overall MM platform is divided into four layers: data curation layer (DCL), information curation layer (ICL), service curation layer (SCL) and supporting layer (SL). The DCL is responsible for curating the data. It consists of different modules for data streaming and communication, data representation and mapping and big data storage in a Hadoop Distributed File System (HDFS). HDFS addresses the volume,



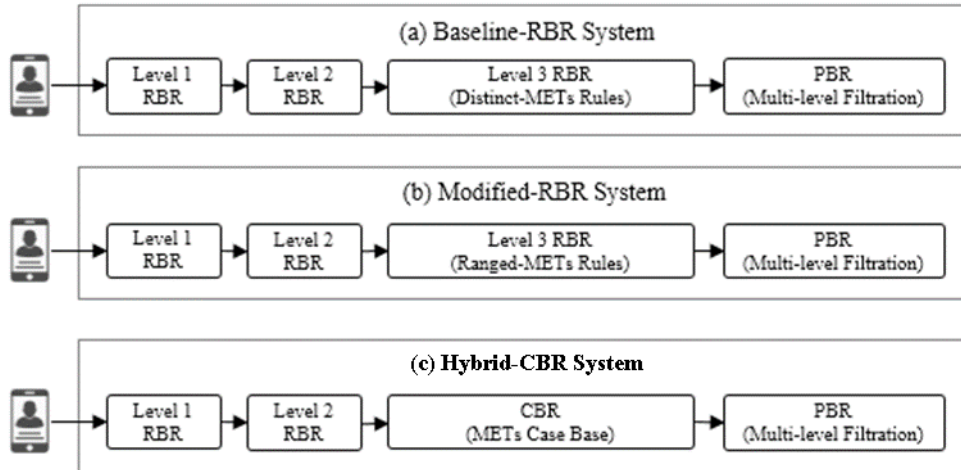
velocity and variety aspects of raw sensory data acquired using mobile sensors. The accelerometer raw data for low-level activities (i.e., sitting, standing, moving in a bus, moving in a subway, walking, running, and cycling) are transferred to the DCL virtual machine, which is transformed to have a structured format and stored in a relational data model on the DCL server machine. The mobile device used in this case works as a gateway to connect to the DCL cloud server over the Internet. The stored data are fed to the ICL for activity recognition that leads to context formulation and behavior analysis of the users' daily activities. The information is stored back in the HDFS logs of the DCL. The processed activities, context, behavior information, and personal profile information are utilized by the SCL for reasoning and providing personalized physical activity recommendations. In SCL, knowledge bases are created by domain experts based on the online guidelines and experts' past experiences. This enables the process of provisioning personalized recommendations to users based on their needs, preferences, and interests. SL facilitates other layers by providing security, privacy, visualization and user interfaces. The user's personal profile information is collected using a mobile application and stored on the DCL server in a relational data model.

A multimodal hybrid reasoner is a key component of MM and plays the role of an intelligent service provisioning agent. It performs execution on the server side of the SCL and enables personalization of physical activity recommendations by integrating data and knowledge from diverse sources. The focus of this chapter is on the reasoning methodology and its usefulness in MM for generating personalized physical activity recommendations.

#### **7.4. Methodology - design and implementation of HRM and hybrid- CBR classifier**

For building an intelligent physical activity recommendation system, this thesis moved beyond the traditional single reasoning methodology systems to a multiple reasoning methodology system. This research integrates RBR and CBR with PBR into a single methodology called multimodal hybrid reasoning methodology (HRM).

HRM forms the basis of multimodal hybrid reasoner for the MM platform, which is the focus of this chapter. In HRM, these methodologies can be integrated in any of the following design strategies, shown in Figure 7.3.

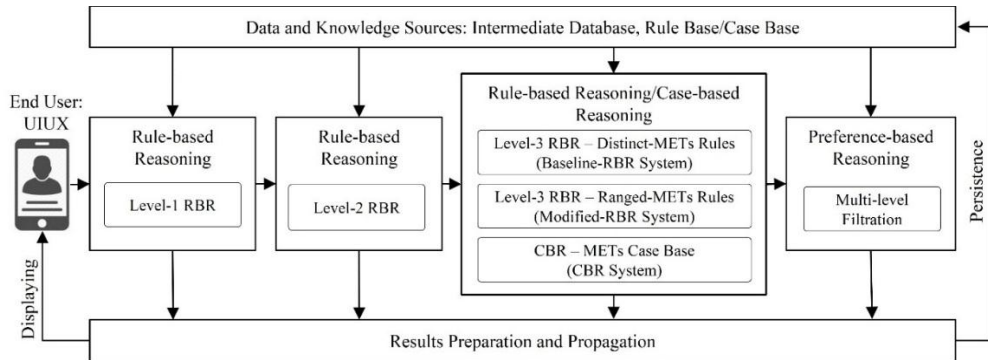


**Figure 7.3.** Design strategies of the proposed hybrid reasoning methodology

In Figure 7.3 (a-c), the sequence of the design strategy of baseline-RBR is as follows: level-1 RBR is followed by level-2 RBR, which is followed by level-3 RBR and PBR. The design strategy of the modified-RBR follows the same strategy as the baseline-RBR, except for the ranged-METs<sup>6</sup> rules, which are used at the level-3 RBR. The strategy of hybrid-CBR differs from those of the first two strategies at level-3, where CBR is used instead of RBR. In our chapter, the first strategy is used for building a baseline system to compare the results of the other strategies. The second strategy is the improved version of strategy 1, which is implemented in MM system (v1.0) but has its own limitations. To eradicate the shortcomings of the first two strategies, the third strategy of hybrid-CBR is used, which integrates RBR, CBR, and PBR. This strategy is experimented and realized outside the MM platform on a local set up in our lab.

<sup>6</sup> A metabolic equivalent, or *METs*, is a unit used to describe the energy expenditure of a specific physical activity. A *METs* is the ratio of the rate of energy expended during an activity to the rate of energy expended at rest (2008 Physical Activity Guidelines for Americans).

Based on the idea illustrated above, the core components of the proposed multimodal hybrid reasoner have been defined and depicted them in the functional flow diagram shown in Figure 7.4.



**Figure 7.4.** Functional diagram of proposed multimodal hybrid reasoning model

Figure 7.4 shows high-level interactions of the different components of the reasoner along with the methodology used in each component. Like any other reasoning system, the core components of the proposed reasoner include the following: input/output interfaces, input data sources, knowledge bases, reasoning methodology and outputs. They are explained below as follows.

*Input/output interfaces:* user's smart phone that runs the MM application works as the input/output interface for the reasoner.

*Input data sources:* inputs of the reasoner include user requests, personal profile data, and daily physical activity data. The input data, except for the user requests, are stored in an intermediate database. The request for recommendation is received from the user's mobile application.

*Knowledge Base:* knowledge of the reasoner is composed of rules created from physical activity guidelines and past successful cases obtained from the implicit experience of the domain experts. The rules are stored in the rule base, while the past successful cases (METs index) are stored in the METs case base (METCB).

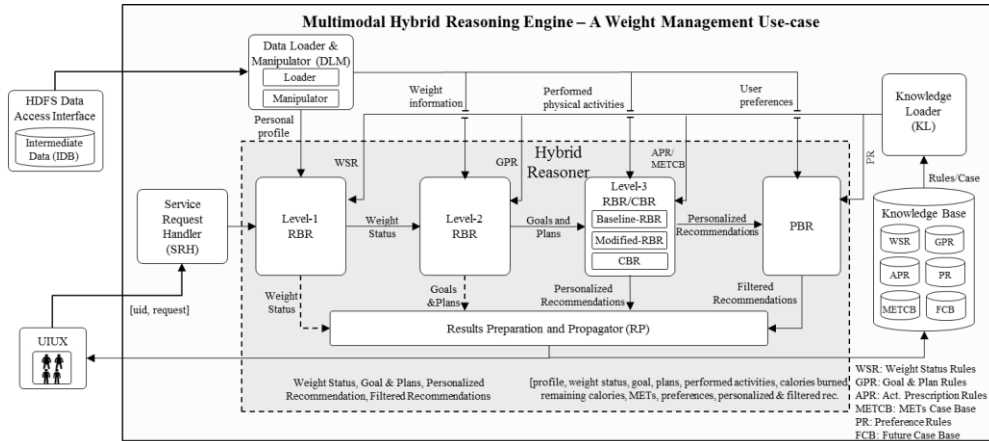
*Reasoning methodology:* the reasoning methodologies include RBR, CBR, and PBR, which are integrated in a linear combination. The RBR methodology is applied at multiple levels: level-1, level-2, and level-3. At level-3, RBR is either used with distinct-METs rules or with ranged-METs rules. At the same level, CBR can also be used (using METs cases) as a counterpart of RBR for improved services. At the end, the multi-level filtration mechanism is applied in PBR to filter out irrelevant recommendations by utilizing the user's preferences and interests.

*Outputs:* outputs of the reasoner include wellbeing recommendations for users, weight status, weight management plans, personalized physical activity recommendations and personalized filtered physical activity recommendations. These recommendations are aggregated and prepared by Results Propagator and then delivered to the end user and intermediate database. In the intermediate database, it is stored for future use as a successful case.

In the subsequent sub-sections, a detailed description of the architectural design of the proposed HRM is provided, and then, the process of knowledge creation is discussed. Finally, the reasoning methodology is described in detail.

#### **7.4.1. Architectural design and workflow**

A detailed data flow diagram of the multimodal hybrid reasoning engine illustrating communication is shown in Figure 7.5.



**Figure 7.5.** Detailed data flow diagram of the proposed multimodal hybrid reasoning model

The key components of the HRM are service request handler (SRH), data loader and manipulator (DLM), knowledge base (KB), knowledge loader (KL), hybrid reasoner (HR) and result preparator and propagator (RP). The hybrid reasoner consists of RBR, CBR, and PBR modules along with the PR module. The RBR, CBR, and PBR modules work cooperatively in a linear combination for enhancing recommendations. CBR is the key reasoning methodology that is activated by the output of RBR. The output of CBR in turn activates the PBR methodology to personalize the recommended physical activity.

From the service execution perspective, when a user requests service, the SRH analyzes the request and activates the appropriate module of the reasoner. SRH supports the MM platform for multiple service generation. SRH forwards the request to HR, where the RBR (level-1, level-2), level-3 RBR/CBR, and PBR methodologies are sequentially executed. Outputs of the HR are forwarded to the RP module for final preparation and forwarded to the user mobile application interface (UIUX) for being displayed to the users.

For the weight management scenario, the multimodal hybrid reasoning methodology operates in the following flow.

- First, level-1 RBR is applied, which loads the weight status rules (WSR) from the KB and the required personal profile data from the intermediate database (IDB) using the data loader (DL) component. The necessary computation on

the personal data, e.g., BMI calculation from height and weight information, is performed using the data manipulator (DM) and passed to the level-1 RBR. The level-1 RBR uses RBR methodology to recommend weight status recommendations (normal, overweight, underweight) as a service to the user and to level-2 RBR for further processing.

- Level-2 RBR receives the output of the level-1 RBR as input and performs the same reasoning procedure as level-1 RBR for recommending the goal state and associated calorie consumption plan and weight management plans. The level-2 RBR uses the goal and plan recommendation rules (GPR) loaded by KL from the KB and the personal profile data loaded by the DL from the IDB. The purpose is to generate goal and plan recommendations, which are provided to the users as a service and to level-3 RBR/CBR for further processing.
- Level-3 RBR/CBR receives the output of the level-2 RBR as input and further generates physical activity recommendations. Level-3 RBR/CBR supports both the RBR and CBR methodologies. The RBR results in baseline-RBR and modified-RBR systems. The baseline-RBR uses distinct-METs rules, while the modified-RBR uses ranged-METs rules that are loaded from the KB during the activity recommendation generation. The CBR methodology uses the METCB of the historical successful physical activity recommendations. In our case, the 2011 compendium of physical activity guidelines [184] are used as our key physical activity case base, which has physical activity recommendations associated with METs values. In either case (i.e., baseline-RBR, modified-RBR or hybrid-CBR), the list of all of the performed physical activities is loaded from the IDB and commutated for the duration, amount of consumed calories, remaining calories and corresponding metabolic equivalent (METs) value. The corresponding physical activities for the METs value are recommended and provided to the users. These physical activities are not filtered according to the preferences and interests of the users; therefore, they are forwarded to the PBR module for further personalization.

- PBR first receives the physical activities recommended by level-3 RBR/CBR and then loads the personal preferences and interests information from the IDB. The associated preference-based rules (PR) are loaded from the KB to apply multi-level filtration for filtering out irrelevant recommendations. The final filtered recommendations are personalized according to the user's personal preferences.
- The personalized recommendations are passed to the RP for proper preparation and packaging to be forwarded to the user application to be displayed on the user's mobile application.
- The user query, intermediate recommendations, and final personalized recommendations are stored in the future case base (FCB) for future use.

#### **7.4.2. Knowledge Acquisition**

Knowledge is one of the most important ingredients of a reasoning system. This section describes how the knowledge used by the HRM is created. The key methodologies of HRM are RBR and CBR; therefore, first need is to create knowledge in the form of rules and cases. The process of knowledge acquisition is discussed below.

##### **7.4.2.1. Rules Creation: Translating Guidelines**

Wellness guidelines are the key source of information for improving the quality of life. Translating guidelines into computer-processable rules is a challenging task because it requires the involvement of knowledge engineers and domain experts [185]. In our work, knowledge from the unstructured guidelines of a weight management scenario is translated to rules with the help of three knowledge engineers and two domain experts. Based on the design of our research, the knowledge engineers first studied the weight management scenario, surveyed the weight management guidelines, indexed them, and categorized them into two groups: (a) standard equations to compute standard values and (b) indexes to be used in rule creation. An example of the first category is the calculation of calories burned/day,

while an example of the second category is the BMI index. These rules are used by the RBR to generate physical activity recommendations. The process of guideline translation is described below.

### Personal profile assessment

To classify users into underweight, normal or overweight states, personal profile assessment based on the standard BMI index is required [186]. The BMI index and personal profile information are combined together to form rules, which are shown in Table 7.1. For the BMI calculation, the standard BMI formula is used.

**Table 7.1.** Weight status rules (WSR) based on Body Mass Index (BMI)

Gender	Age	BMI value	Weight status
M or F	>20	<18.5 kg/m <sup>2</sup>	Underweight
M or F	>20	>18.5 and <25 kg/m <sup>2</sup>	Normal
M or F	>20	>25 and <30 kg/m <sup>2</sup>	Overweight

These rules are applicable for adults and used by level-1 RBR for finding the weight status of the users.

### Goal setting and plan management

A weight management system requires goals and the associated plans to achieve the goals. A goal can be either a local goal or global goal (gloGoal). A global goal is the final objective of the user to be achieved, while the local goal refers to a set of sub-goals to reach the global goal. For example, the total weight to be lost is considered a global goal, while weekly targets are considered local goals. To set a global goal in the context of the weight management scenario, first, an estimation of the ideal body weight (idlWgt) is required, which can be obtained using the Robinson JD [187] equation. The difference between the current weight (curWgt) and ideal weight yields the best estimation for the target goal in terms of the number of kg to be lost. The ideal body weight and global goal are computed using equation 1 and equation 2.

$$\begin{aligned} \text{idlWgt} &= 51.65 \text{ kg} + 1.85 \text{ kg/inch over 5 feet} && \text{(man)} \\ \text{idlWgt} &= 48.67 \text{ kg} + 1.65 \text{ kg/inch over 5 feet} && \text{(woman)} \end{aligned} \quad (1)$$

The ideal body weight is a debatable topic but has successfully been used in healthcare systems, such as drug dosage estimation [187] and cell transplantation



[188]. Therefore, it has been adopted for the estimation of the global goal in our research scope.

$$\text{gloGoal}(\text{kg}) = \text{curWgt}(\text{kg}) - \text{idlWgt}(\text{kg}) \quad (2)$$

In our system, gloGoal by itself is a user service, but it is aimed towards devising plans for achieving the global goal. The rules defined for identifying appropriate plans, such as a weight loss plan, weight gain plan and weight maintenance plan (GPR), are shown in Table 7.2.

**Table 7.2.** Goals and weight management Plan Rules (GPR)

Gender Male (M)/ Female (F)	Global Goal (gloGoal) - Kg	Weight Status (WS)	Plan Prescription (PP)
M or F	> 0 (+ive)	Normal or Overweight	Weight Loss Plan (WLP): lose gloGoal(Kg)
M or F	= 0 (neutral)	Normal	Weight Maintenance Plan (WMP): motivational statements
M or F	< 0 (-ive)	Underweight	Weight Gain Plan (WGP): gain gloGoal(Kg)

In Table 7.2, the focus is only on the first two cases.

Details of the suggested plan, i.e., duration for achieving the global goal, can be computed using equation 3.

$$\text{wghRedPlan}(\text{days}) = \text{roundup} \left( \frac{7(\text{days}) * \text{gloGoal}(\text{Kg})}{0.5(\text{Kg})} \right) \quad (3)$$

In equation 3, a constant value of 0.5 represents the weight to be lost during one week. From this equation, local goals for weeks and months can be determined by subtracting a weight of 0.5 kg from the weight of the previous week (weekly plan). These plans can also be estimated in terms of the calories burnt (per day, per week, per month, etc.) using equation 4.

$$\text{calToBurDay} = \frac{\text{gloGoal}(\text{kg}) * \text{Cal}(1\text{kg fat})}{\text{wghRedPlan}(\text{days})} \quad (4)$$

In equation 4, Cal represents the number of calories equivalent to burning 1 kg of body fat.

All of these rules are used for setting the goal, devising plans, and managing weight and are used by level-2 RBR.

### Physical activities assessment

Once a weight management plan is assessed, monitoring and recognition of the user's physical activities are required. Based on monitoring the previous day's activities, using the accelerometer sensor of the smartphone, the next day recommendations are planned. This process is performed in terms of the duration spent in each activity and the estimated amount of calories burnt. The amount of each activity (amtAct) is calculated by taking sum of all of the time slots (timSlot) during which the user performed that activity (Act), computed using equation 5.

$$\mathbf{amtAct}_i = \sum_{j=1}^t \mathbf{Act}_i \cdot \mathbf{timSlot}_j \quad (5)$$

The estimation of calories (Cal) for a specific activity ( $\mathbf{Act}_i$ ) in a specific time duration,  $\mathbf{amtAct}_i$ , can be estimated by the product of the METs of that activity with the amount of activity and current weight of the subject. This calculation is shown in equation 6, which is adapted from the compendium of physical activities [184].

$$\mathbf{Act}_i \cdot \mathbf{Cal} = \mathbf{Act}_i \cdot \mathbf{METs} * \mathbf{amtAct}_i(\mathbf{h}) * \mathbf{weight}(\mathbf{kg}) \quad (6)$$

METs estimates the capacity and tolerance level of an individual to exercise in which he/she may participate safely without hurting him/herself [189]. In the proposed system it is used to estimate calories from the physical activities and vice versa. In the calorie estimation process, the average METs, rather than the exact MET value is used. The average METs for an activity (e.g., walking) is calculated by considering all types of walking included in the METs guidelines [184] and taking the average. The same procedure is used for other activities that are considered (i.e., running, jogging, transportation, sitting, and standing). The rationale behind the average METs

is the limitation of our current activity recognition system in recognizing the exact intensity of every sub-type of activity, for example, walking.

After applying equation 6, for all of the activities, equation 7 is used to sum all of the estimated calories.

$$\text{totalBurnedCal} = \sum_{i=1}^a \text{Act}_i. \text{Cal} \quad (7)$$

The remaining calories (remCalToBurn) for the rest of the day (in a daily calorie consumption plan) are computed using equation 8.

$$\text{remCalToBurn} = \text{calToBurDay} - \text{totalBurnedCal} \quad (8)$$

The aim of estimating the remaining calories is to recommend the appropriate physical activity using our reasoning system to meet the goals of the day. This recommendation requires the METs value computed from the remCalToBurn using equation 9 [184].

$$\text{METs} = \frac{\text{remCalToBurn}}{(\text{amtAct} = 1\text{h}) * \text{weight (kg)}} \quad (9)$$

The METs value is used, both, in RBR and CBR to recommend the appropriate physical activity. For RBR, rules need to be created using the user's personal information and the required METs value. For CBR, a case base is to be prepared.

### Rules creations

Based on the estimated METs value and the user's personal information (e.g., age), two types of rules are created. The first type is based on distinct-METs values, and the second type is based on ranged-METs value. The distinct-METs rules are used to build the baseline-RBR system, while the ranged-METs rules are used for building the modified-RBR system. When distinct-METs is considered and age together, total 122 rules are created for the 48 distinct-METs values. The distribution of the rules is as follows: 22 rules belong to the Young age group, 33 rules belong to the Older Adults group, and 47 belong to the Adults group. In the context of physical activity recommendation, age plays an important role; therefore, it is considered an essential

part of the rules. The transformation of age to different age groups is supported by the guidelines from WHO [190] and UK [191]. These guidelines categorize users into three major age groups: Young (age 5-17), Adults (age 18-64), and Older Adults (age  $\geq 65$ ). A partial list of the distinct-METs rules is shown in Table 7.3.

**Table 7.3.** Distinct-METs rules for baseline-RBR

Rule ID	Age Group	METs value	Activity prescription
R#1	Young	2	Walking, household
R#2	Older Adults	6.5	Climbing hills with 0 to 9 lb load; Race walking; rock or mountain climbing
R#3	Young	7.8	Backpacking; hiking or organized walking with a daypack
R#122	Adult	15	Running; stairs up

In the MM implementation, ranged-METs rules are used; therefore, first, ranges are defined for the METs values used in these rules. According to the well-known physical activity guidelines from the center for disease control and prevention (CDC), American College of Sports Medicine (ACSM) [192], WHO [190], US [193] and UK [191], physical activities can be grouped into three categories: light ( $< 3.0$  METs), moderate ( $3.0$  to  $6.0$  METs) and vigorous ( $> 6.0$  METs). According to these guidelines, moderate to vigorous-intensity physical activities are recommended to Young, Adults and Older Adults, but with slightly changed doses and patterns. For example, the Young group is recommended a physical activity of METs  $\geq 3$ -7, and the Adults and Older Adults groups are recommended a physical activity of METs  $\geq 3$ . However, the Older Adults group is recommended the same physical activities in the range of METs values for the Adult group but with lower intensity and dose due to their lower capabilities for exercise and physical activities. These guidelines are formulated by considering the threshold value of METs  $\leq 10.25$  for Older Adults, METs  $\leq 7$  for Young and METs  $\leq 23$  for Adults. The light-intensity activities (i.e., METs  $< 3$ ) are appropriate for all age groups because they do not lead to injuries. Based on this grouping of the METs values by the age groups, the ranged-METs rules are defined and summarized in Table 7.4.

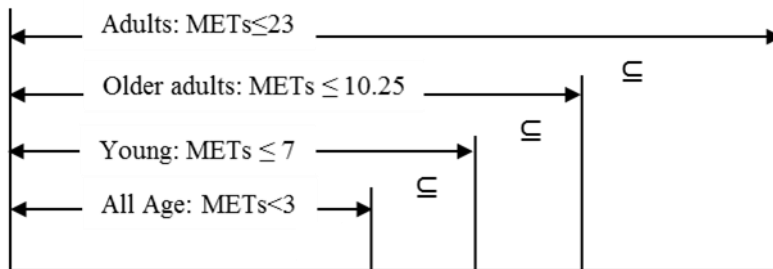
**Table 7.4** Ranged-METs rules for modified-RBR

Rule ID	Age Group	METs value	Activity prescription
R#1	Young, Adults, Older Adults	$< 3$	Light activity
R#2	Adults	$\leq 23$	Moderate – vigorous-intensity
R#3	Older Adults	$\leq 10.25$	Moderate – vigorous (lower intensity level)
R#4	Young	$\leq 7$	Moderate

#### 7.4.2.2. Case Base Creation

The CBR part of HRM operates based on well-established past successful cases to generate physical activity recommendations. The cases in the case base are adapted from the 2011 compendium of physical activity guidelines [184]. These guidelines contain a list of physical activities associated with METs values. The METs values are used and the associated physical activities as the two key attributes of our case base. This case base is named as METs case base (METCB). Based on the discussion made in last section, the number of attributes of the METCB are extended to include an additional attribute, age group. The relationship between age group and METs ranges is represented in equation 10 and depicted in Figure 7.6.

$$\text{Age Group} = \{\text{All Age} \subseteq \text{Young} \subseteq \text{Older Adults} \subseteq \text{Adults}\} \quad (10)$$

**Figure 7.6.** Distribution of subjects based on age factor

In the above equation 10 and Figure 7.6, it can be seen that a fourth age group, which is labelled as ‘All Age’ (METs  $< 3$ ), is added to the case base. It is a subset of all of the other age groups because activities of this intensity are not injurious and can equally be recommended to any age group. The current METCB contains 119

instances, which may increase in the future. Table 7.5 presents the detailed characteristics of the METCB.

**Table 7.5.** Case base structure

Attribute	Data type	Possible value	Description
Age Group	Symbol	{All Age, Young, Adults, Older Adults}	Age of the subject
METs	Float	Min=1.3, Max=23.0	Metabolic Equivalents of Tasks one hour
Recommendations	String	Physical activities {running, walking, cycling, traveling-bus and subways, standing, sitting}	Physical activities

### 7.4.3. Hybrid Reasoning and Recommendation

Hybrid reasoning is the key methodology implemented in the proposed reasoning engine that generates personalized physical activity recommendations in the MM system. It is composed of RBR, CBR, and PBR and is discussed in the subsequent sub-sections.

#### 7.4.3.1. Rule-based reasoning and recommendation

In HRM, the RBR methodology works at three levels (level-1, level-2, and level-3). Its objectives include the following: (1) assessment of personal information and recommendation for weight status, (2) assessment of the ideal body weight and recommendations for goals and plans and (3) assessment of the performed physical activities and recommendations for appropriate physical activity. The recommendations of each level are provided to the user, on one end, and to the next level, on the other. For example, the first level of recommendations is provided to the user and to the level-2 RBR. This process involves a sequential flow, and finally, recommendations are generated, which are provided to the users on their mobile applications. The idea of provisioning intermediate results to the users is motivated from the fact that MM system supports the PULL service model, where users can subscribe either to a single service or a combination of services. Using this approach, some of the users subscribe only for weight status recommendations, while others

subscribe for goal and plan recommendations and physical activity recommendations. If the MM system is constrained only to support the PUSH service model, then it may be enough for the users who require services on the subscription basis but will not support users who require customized subscription-based services.

### Level-1 RBR

Once the user query arrives at the HRM, level-1 RBR gets activated, loads personal profile information, performs the necessary computations, retrieves the WSR (Table 7.1) and starts the rule-based reasoning process [194]. The outputs are provided to the end user and to the level-2 RBR. The steps of the level-1 RBR are listed in Algorithm 1.

---

**Algorithm 1.** Rule-based reasoning for the recommendations of weight status

---

```

Begin
  Input: UID:uid
  Output: Weight Status (WS): List <Weight Status>

  Let SID:sid = Weight Status Service Id
  WSR: Set of Weight Status Rules, WSR =  $\emptyset$ 
  KB: Knowledge Base
  1. Foreach RULE R in KB
    If ( $R \in \text{sid}$ )
       $\text{WSR} := \text{WSR} \cup R$ ;
    End If
  End for
  2. Foreach RULE R in WSR
     $\text{WS} := \text{ExecuteWSRule}(R, \text{uid})$ 
    If  $\text{WS} \neq \text{"Underweight"}$ 
      PropgatWSResultsToUIUX (uid, WS);
      InvokeLevel2RBR (uid, WS ); // See Algorithm 2
      Go to step 3
    Else

      PropgatWSResultsToUIUX (uid, educational & motivational statments for Weight Gain)
      Go to step 3;
    End If
  End for
  3.  $\text{FCB} := \text{AddWStatus}(\text{uid}, \text{WS})$ ; // See discussion
  4. Exit;
End

```

---

In first step of Algorithm 1, WSR are loaded from the knowledge base using an iterative loop process. The design of the knowledge base is based on the types of services, and rules are stored accordingly. Therefore, the type of service identifies the type of rules to be loaded. The type of service can be identified by the service Id (sid,

in this case). Once the rules are loaded, the execution commences. The definition of ExecuteSWRule() is given in Function 1, and it loads the personal profile data of the user from the IDB and performs the necessary computations. The data loading process of the IDB uses a simple object access protocol (SOAP)-based service, defined in the SCL. Finally, the pattern matching process starts, and when a rule is matched, it is fired, and its corresponding weight status recommendations are generated. The results of this function are returned to Algorithm 1 for further processing.

Function 1. Rules execution for the weight status recommendations

---

ExecuteWSRule (RULE R, UID uid)

---

Let WS = Weight Status, showing BMI status of the user

IDB: Intermediate Database

PPROF: Personal Profile

BMI: Body Mass Index

RHS: Right Hand Side

LHS: Left Hand Side

1. Load PPROF of uid from IDB;
  2. Compute BMI;
  3. **If** R.LHS. values = (PPROF and BMI)  
    WS := RHS of R;
  - End If**
  4. Return (WS)
- 

When the weight status recommendations are received by Algorithm 1, they are forwarded to the user mobile application interface (UIUX) and to the level-2 RBR. The function PropgatWSResultsToUIUX() is responsible for providing the recommendations to the user while the function InvokeLevel2RBR() is used to invoke the level-2 RBR. The propagation function first communicates with the user's mobile application and then provides the generated intermediate recommendations along with some metadata for display purposes. In case the intermediate result of the level-1 RBR is the underweight status, then the system propagates motivational and educational statements using the PropgatWSResultsToUIUX() function.

### Level-2 RBR

Level-2 RBR is activated by level-1 RBR for setting goals and prescribing the associated weight loss and calorie consumption plan recommendations. In level-2 RBR, the goal and plan rules (GPR) specified in Table 7.2 are used along with eqs. 1-4. The algorithmic steps of level-2 RBR are given in Algorithm 2.



---

**Algorithm 2.** Rule-based reasoning algorithm for goals and plans prescription recommendations

---

**Begin**

**Input:** UID:uid, WS

**Output:** Weight Loss Plan (WLP)

Let SID:sid = Weight Loss Service Id

GPR: Goal and Plan Rules,  $GPR = \emptyset$

PP: Plan Prescription

1.   **Foreach** RULE R in KB // KB: Knowledge Base

**If** ( $R \in \text{sid}$ )

$GPR := GPR \cup R$ ;

**End If**

**End for**

2.   **Foreach** RULE R in GPR

    PP = ExecuteGPRRule (RULE R, UID uid)

**If** PP = "WLP"

        Let wlPlan:= List <WLPlan>;

        wlPlan = ComputeWLPlansInKgAndCalories(); // use equation 3 and 4

        PropgatWLPResultsToUIUX (uid, wlPlan);

        FCB := AddRecommendedPlan(uid, wlPlan); // See discussion

        InvokeLevel3RBR – CBR (uid, wlPlan ["caloriesPlan"]); // See Algorithm 3

        Go to step 3;

**Else**

        PropgatWMPResultsToUIUX(uid, educational & motivational statments  
        for Weight Maintenance)

        Go to step 3;

**End if**

**End for**

3.   Exit;

**End**

---

In Algorithm 2, the rules are loaded from the KB on the basis of service type (sid). The service is goal and plan recommendations, and the associated rules are the GPR. After the rules are loaded, Algorithm 2 executes ExecuteGPRRule() to generate the plan prescription (PP) recommendations. The definition of this function is shown in Function 2, which takes each rule from the GPR and retrieves the required personal

profile data from IDB and computes the ideal body weight (idlWgt) and global goal (gloGoal). The pattern matching process then starts, and each attribute of the left hand side (LHS) of the rule R is checked against the loaded and computed values. When a match is found, rule R is fired, and its right hand side (RHS) is provided as the PP recommendation. These recommendations are returned to Algorithm 2 for further processing.

Function 2. Execution of the goal and plan rules for goal and plan  
recommendations

---

ExecuteWMPPlanRule (RULE R, UID uid)

---

Let IDB: Intermediate Database

gloGoal: global Goal

idlWgt: ideal Weight

PPROF: Personal Profile

LHS: Left Hand Side

RHS: Right Hand Side

PP: Plan Prescription

1. Load PPROF of uid from IDB;
  2. Compute Ideal Weight (idlWgt) ; //use equation 1
  3. Compute Global Goal (gloGoal); //use equation 2
  4. **If** R.LHS.values = (PPROF, gloGoal)
    - PP := RHS of R;
    - End if**
  5. Return (PP);
- 

If the output retained in PP is weight loss plan (WLP), then the ComputeWLPlansInKgAndCalories() function is activated for computing daily, weekly, and monthly plans in terms of the number kg to lose and the associated calorie consumption plans. These plans are forwarded to the users and are displayed on their mobile application interface (UIUX) and are also forwarded to level-3 RBR-CBR. The functions responsible for these tasks are PropgatWLPResultsToUIUX() and InvokeLevel2RBR – CBR(), respectively. In case the PP value is the weight maintenance plan (WMP), then educational and motivational statements are provided to the users using the PropgatWMPResultsToUIUX() function.

### Level-3 RBR-CBR

In HRM, level-3 RBR-CBR uses either baseline-RBR or modified-RBR or CBR methodology. For these methodologies, an assessment of the performed physical activities is required in terms of the burned calories, remaining calories, and equivalent METs value. This assessment and the computations are performed using equations 5-9. In the baseline-RBR, distinct-METs rules (Table 7.3) are used, while in the modified-RBR, ranged-METs rules (Table 7.4) are used to generate personalized physical activity recommendations. The algorithmic steps for both the baseline-RBR and modified-RBR are given in Algorithm 3 and are the same from the methodology perspective but different based on the nature of rules they use (for the level-3 CBR, see section 7.4.4).

---

**Algorithm 3.** Assessment of physical activities and prescription of physical activity recommendations using rule-based reasoning

---

**Begin**

**Input:** UID:uid, wPlan

**Output:** Personalized Physical Activity Recommendations (PAR): List <Recommendations>

Let SID:sid = Personalized Physical Activity Recommendation Service

APR: activity prescription rules and APR =  $\emptyset$

1.     **Foreach** RULE R in KB // KB: Knowledge Base
  - If** (R  $\in$  sid)
    - APR := APR  $\cup$  R;
  - End if**
2.     **End for**
3.     **Foreach** RULE R in APR
  - PAR := ExecuteActPrescRule (RULE R, UID uid)
  - If** PAR  $\neq \emptyset$ 
    - Break;
  - End If**
4.     **End for**
5.     PropagatPARResultsToUIUX (uid, PAR);
6.     FCB := AddRBRPAR(uid, PAR); // See discussion
7.     InvokePBR(uid, PAR); // See Algorithm 5
8.     Exit;

**End**

---

Algorithm 3 first loads the activity prescription rules (ARP) from the KB based on the service id, specified in the service request. For generating appropriate personalized physical activity recommendations (PAR), the ExecuteActPrescRule() function is used, the details of which are given in Function 3. The physical activities are recommended on the basis of the final computed METs values and the user's

personal profile information. The METs value represents the intensity level of a physical activity. Within the same physical activity type, for example, walking, different intensity values exist that range from a METs value of 2.3 to a METs value of 12 [184]. Similar ranges exist for other activities as well, such as running, cycling, transportation, standing, and sitting. In the METs guidelines, a large number of distinct METs values are available, which makes it hard to define distinct METs rules. One of the solutions to this issue is to define range-based METs rules. In the MM implementation for the weight management scenario, METs range-based rules are used.

Function 3. Execution of distinct-METs and ranged-METs rules for physical activity recommendations

---

ExecuteActPrescRule (RULE R, UID uid)

---

Let IDB: Intermediate Database  
 METs: Metabolic Equivalent of Task  
 PPROF: Personal Profile  
 AMTACT: Amount of Physical Activity Performed  
 PAR: Personalized Physical Activity Recommendations: List <Recommendations>  
 LHS: Left Hand Side  
 RHS: Right Hand Side

1. Load PPROF, AMTACT of uid from IDB;
2. Compute AMOUNT OF PHYSICAL ACTIVITY performed so far; //use equation 5
3. Compute CALORIES for each ACTIVITY; //use equation 6
4. Compute TOTAL BURNED CALORIES ; //use equation 7
5. Compute REMAINING CALORIES ; //use equation 8
6. Compute METs value; //use equation 9
7. **If** R. LHS.values = (PPROF, METs)  
     PAR := RHS of RULE;  
     **End if**
8. Return (PAR)

---

Once PAR are generated, they are provided to the end users on their mobile application interface (UIUX) using the PropgatPARResultsToUIUX() function. The output of Algorithm 3 can be a list of physical activities that are generated either on the basis of ranged-METs rules or multiple physical activities against a single METs value in a rule. To filter this list of recommendations and personalize them to another level, they are provided to the PBR methodology by using the InvokePBR() function call of Algorithm 3 (see section 7.4.5 for the PBR functionality).

#### 7.4.4. Case-based Reasoning (CBR)

To overcome the limitations of level-3 RBR implemented in the MM platform, CBR is used for generating more personalized recommendations. The CBR implementation is performed outside the MM implementation in our lab with the aim of enhancing the performance of HRM. The CBR methodology helps in recommending specific physical activity to users based on their gender information and required intensity for physical activity i.e., METs value. The CBR methodology is selected due to its capabilities of (1) recommending specific and precise physical activities to the user, (2) providing a list of top relevant physical activities as recommendations (e.g., walking) with multiple similar alternatives (e.g., running or cycling) and (3) refining the suggested recommendations based on the user's feedback for enhancing recommendation accuracy and specificity. CBR execution follows the standard CBR cycle (retrieve, reuse, revise and retain) to complete the process of suggesting and refining recommendations along with an incremental learning approach. In our thesis, the revise step could not be performed in the HRM due to the limitation of the MM system in being unable to handle user feedback. This phase is left as future work.

#### 7.4.4.1. Retrieve and Reuse Steps

In our CBR model, the case query contains two attributes, age group and METs value. The age value is retrieved from the personal profile of the user, which is transformed to the predefined age group. The value of the METs attribute is computed from the user's personal profile information and the physical activities the user performed so far. For this purpose, steps 1-6 of Function 3 are used. These values are provided to the retrieve step of the CBR, which starts retrieving similar cases from the METCB. For the retrieval of age group and METs values, two local similarity functions are defined, which are shown in equation 11 and equation 12.

$$\text{METSim}_1(\text{nC}, \text{eC}) = \frac{d_g(\text{Max}_{\text{MET}}, \text{Min}_{\text{MET}}) - d_l(\text{nC}_{\text{MET}}, \text{eC}_{\text{MET}}) - 1}{d_g(\text{Max}_{\text{MET}}, \text{Min}_{\text{MET}})} \quad (11)$$

Here,  $\text{METSim}_1$  calculates the similarity of the METs between the new query case (nC) and existing cases (eC) in the METCB. Similarly,  $d_g$  is the global distance

function that calculates the distance between  $\text{Max}_{\text{MET}}$  (maximum METs value in the METCB, i.e., 23 for running) and  $\text{Min}_{\text{MET}}$  (minimum METs value in the METCB, i.e., 1.3 for resting). Here,  $d_l$  is the most important local similarity function that computes the distance between the METs values of  $nC$  and  $eC$ .

$$\text{AGSim}_l(nC, eC) = \begin{cases} \text{AG}_{ij} = 1 & \text{for } \forall (i \geq j) \text{ OR } (i = 0 \text{ OR } j = 1) \\ 0 & \text{otherwise} \end{cases} \quad (12)$$

In equation 12,  $\text{AGSim}_l$  is the local similarity function that matches the METs values of  $eC$  with  $nC$ . The similarity criterion used in the equation is the exact match, which is denoted as value 1. The interpretation of this value is that if the age group of the query case is similar to that of the existing case (i.e.,  $\text{AG}_{nC} = \text{AG}_{eC}$ ), then this value will be 1; otherwise, it will be 0. The symmetric view of the local similarity function of this attribute is represented in a confusion matrix shown in Table 7.6.

**Table 7.6.** Local similarity matrix of ‘age group’ attribute

Age Group	All Age	Young	Older Adults	Adults
All Age	1	1	1	1
Young	1	1	0	0
Older Adults	1	1	1	0
Adults	1	1	1	1

In the above confusion matrix, the diagonal value of each age group is equal to 1, which shows the exact match relationship of each age group with itself. The age group, labeled as All Age, represents the list of METs values (less than 3) that can be equally recommended to the rest of the age groups; therefore its value is 1 for all of the other age groups. Similarly, the METs values of the age group Young (less or equal to 7) are also a subset of the METs values of the Older Adults and Adults age groups; therefore its value is 1 for all these age groups. This makes both the columns identical in the similarity matrix table.

After computing the local similarities, the weighted sum global similarity function,  $\text{Sim}_g$ , is used to compute the global distance between  $nC$  and  $eC$ , as shown in equation 13.

$$\text{Sim}_g(nC, eC) = \beta(\text{AGSim}_1(nC, eC)) + \gamma(\text{METSim}_1(nC, eC)) \quad (13)$$

Here,  $\beta$  denotes the weight value assigned to the attribute age group and  $\gamma$  denotes the weight value assigned to the METs attribute. The value of  $\beta$  is 0.1 (i.e.,  $\beta = 0.1$ ), and the value of  $\gamma$  is 0.9 (i.e.,  $\gamma = 0.9$ ). The higher value of  $\gamma$  represents the importance and contribution of the METs attribute in the final decision. For the selection of similar cases, k-NN [195] with  $k=3$  is used to select the top three similar cases and reuse them as the suggested recommendations. In the MM system, the selection of the top three cases provides choices to the users for following any of the proposed recommendations based on their personal preferences and interests. The top recommended activities are of the same intensity or close to each other in intensity and have similar impacts on an individual's health. The acceptance of the top three recommendations is based on the threshold value (confidence), denoted by symbol  $\mu$ . The threshold value is set to be greater than or equal to 95 (i.e.,  $\mu \geq 95$ ). If a single case satisfies the threshold, only one recommendation is provided as the final physical activity recommendation. If more than 1 case is retrieved, then PBR is activated for further filtration and personalization of the suggested physical activity recommendations (see section 7.4.5). The confidence value for the acceptance of recommendations is the threshold value, which is computed using equation 13. It is the aggregate score obtained from the local similarity values of equations 11 and 12. The method used for aggregation is the weighted sum, which has a higher weight  $\gamma = 0.9$  for the METs attribute and lower weight  $\beta = 0.1$  for the Age Group attribute. To set the confidence/threshold value as  $\mu \geq 95$  (0.05 threshold), the motivation is taken from the well-known studies [196-198] of the statistical community. The authors considered a 95% confidence interval or 0.05 threshold value as the acceptable value for accepting a hypothesis. The detailed working methodology of the proposed CBR is presented in Algorithm 4.

---

**Algorithm 4.** Case-based reasoning methodology for generating personalized physical activity recommendations

---

**Begin**

**Input:** UID:uid, METCBurl, nC:= new Case, where  $nC \in \{\text{PProf, METs}\}$  and nC is computed using equations 5-9

**Output:** Personalized Physical Activity Recommendations (PAR) : List <Recommendations>

Let PAR:= A set of top 3 relevant existing cases as the proposed recommendations

Sim<sub>g</sub>[]:= Array of global similarities of existing cases

1. METCB<sub>r</sub>:= ReteriveCaseBaseFromKB(METCBurl), Where METCB<sub>r</sub> is the matrix  $eC_m \times A_n$ ,  $eC_m$  is the set of existing cases, i.e.,  $eC = eC_1, eC_2, eC_3, \dots, eC_m$ . Similarly,  $A_n$  is the set of attributes, i.e.,  $A_n = A_1, A_2, A_3, \dots, A_n$
2. **For** i = 1 to SizeOfCases(METCB<sub>r</sub>)  
     Let Sim<sub>l</sub>[]:=Array of local similarities of attributes of individual cases  
     **For** j = 1 to SizeOfAttributes(METCB<sub>r</sub>)  
         Sim<sub>l</sub>[A<sub>j</sub>] = ComputeLocSim(nC.A<sub>j</sub>, METCB<sub>r</sub>[i, j]); // use equation 11 and equation 12  
     **End for**  
     Sim<sub>g</sub>[eC<sub>i</sub>] = ComputeGlobSim (Sim<sub>l</sub>); // weighted sum method (equation13)
3. **End for**
4. PAR:= ApplyKNN(Sim<sub>g</sub>); //where k = 3
5. PropgateCBRResultsToUIUX(uid, PAR);
6. FCB := RetainCBRRPAR(uid, PAR); // See discussion
7. InvokePBR(uid, PAR); //See Algorithm 5
8. Exit;

**End**

---

Algorithm 4 begins execution when nC is input to the CBR algorithm. In the first step, the ReteriveCaseBaseFromKB() function is used to load the existing cases from KB to the METCB<sub>r</sub>. For this purpose, the URL of METCB, METCBurl, is used. Each eC is matched against nC, and the distance is calculated using the local and global similarity functions (i.e., equation 11-12). k-NN with k=3 is used to obtain the top three similar cases as the suggested physical activity recommendations. These recommendations are specific and precise compared with the results of the baseline-RBR and modified-RBR systems. The retrieved case(s) is/are passed to the end users as the proposed personalized physical activity recommendations with the help of the PropgateCBRResults()function. Similarly, this/these recommendations(s) is/are also forwarded to PBR using the InvokePBR() function to filter them according to the user's preferences and interests.



#### **7.4.4.2. Retain steps**

Once the reuse step suggests recommendation(s), the whole case needs to be retained in the case base as a new case. In the proposed HRM, this new case is added to a data store, called the future case base (FCB). If the retrieve step ends with a single recommendation, the whole case, including the user's personal profile and suggested activity, is stored in the FCB. However, if more recommendations are generated, the new case is stored in the FCB after applying the PBR methodology (see section 7.4.5).

#### **7.4.5. Preference-based Reasoning (PBR)**

The recommendations generated by the RBR and CBR methodologies are based on the knowledge created based on general guidelines, which are unable to reflect the user's personal interests and preferences. These recommendations are not personalized from the perspective of the user's personal interests and preferences; to satisfy them, another level of refinement and filtration of the suggested recommendations is required that is performed by the PBR methodology. The PBR mechanism exploits the user model, built on top of the user profile. A user model contains the user's personalized requirements, such as preferences and interests. This information is initially acquired from the user, during the registration process and updated thereafter. The recommendations provided by the RBR and CBR exploit data only from the user's personal profile and physical activity behaviors and do not take into account the preferences. When recommendations are provided on the basis of these methodologies, multiple interpretations can be made. For example, consider a scenario where a user U requires X METs of physical activity to burn an amount Y of calories. The RBR or CBR can generate the following set of recommendations for the mentioned scenario.

- Walking M1 minutes OR Running M2 minutes OR Cycling M3 minutes OR Hiking M4 minutes, etc.

These recommendations are equivalent and can meet the user's requirement mentioned in the scenario' however some of them may not fit the user's personal

interests and preferences adequately. It may be that the user is interested in walking and cycling but not in running and hiking. Therefore, the final recommendations should only include walking and cycling.

To obtain the user's final preference-based personalized recommendations, a multi-filter approach is proposed and implemented as part of the PBR. According to this approach, filtered personalized physical activity recommendations (FPAR) are obtained from the list of generated personalized physical activity recommendations using the user preferences (UPreferences). This process of filtration is shown in Algorithm 5.

---

**Algorithm 5.** Filtration of the personalized physical activity recommendations using user preferences

---

**Begin**

**Input:** UID: uid, PAR

**Output:** Filtered Personalized Physical Activity Recommendations(FPAR): List <filteredRecommendations>

    Let UPreferences[] = List of user preferences

    FCB:=Future Case Base

    FPAR: =  $\emptyset$

    1. UPreferences[] = loadUserPreferences (uid); //Load user preferences from user profile in IDB

    2. **Foreach** Recommendation Rec in PAR

**If** (Rec  $\in$  UPreferences)

            FPAR:= FPAR  $\cup$  Rec;

**End if**

**End for**

    3. PropagateFilteredPARToUIUX (uid, FPAR);

    4. FCB := AddFPAR(uid, FPAR); // see discussion

    5. Exit

**End**

---

The process of preference-based reasoning starts by loading the user's list of preferences, denoted by UPreferences, from the intermediate database. The filtration process is performed in step 2 by taking each recommendation from the PAR and checking it against the preference list of the user. If the recommendation does not satisfy the user's preference, it is filtered out; otherwise, it is added to the filtered list FPAR. This process is continued till all of the recommendations in PAR are checked. Finally, the filtered personalized recommendations are provided to the user on his mobile application interface using the PropagateFilteredPARToUIUX () function. At the same time, the final FPAR are retained in the FCB as the recommended physical

activity. This incrementally grows the FCB, which can be best used in future for successful cases of physical activity recommendations.

## 7.5. Experiments and evaluation

For evaluating the performance of the proposed HRM, the following set of tasks are performed. Initially, a weight management scenario is defined, then set up a set of experiments, and finally performed the experiments and analyzed the results.

### 7.5.1. Case-study: weight management

A weight management scenario is designed and implemented for healthy individuals who are overweight or tend to overweight. After implementation of the methodology, ten volunteers (ages 26-38 years) were asked to use the system for a couple of weeks. The basic personal information of these individuals is shown in Table 7.7.

**Table 7.7.** Personal profile information of the volunteers for system evaluation

User ID	Gender: Male (M), Female (F)	Age (Years)	Height (Feet)	Weight (Kg)	Preferred activities
1	M	26	6.2	84.5	running, walking
2	M	28	5.7	72.5	running, walking, cycling
3	M	28	5.8	70.1	walking
4	M	31	5.4	68	running, cycling
5	M	31	5.6	71.9	walking, traveling
6	M	32	6	85.9	running
7	F	32	5.2	65	walking, jogging
8	M	37	5.8	75	walking, cycling
9	F	30	5.2	75	walking running, cycling
10	M	38	5.8	71	running, cycling

The individuals were asked to use the application during the specified period of time and follow the recommendations provided. During the user's physical activity, the mobile application collected the user's daily physical activity data using the accelerometer sensor of the smartphone. These activities included sitting, standing, moving in a bus, moving in a subway, walking, running and cycling, which are recognized by the activity recognizer module (in the ICL) of the Mining Minds platform (Figure 7.2). For the detailed methodological process of recognition of these

activities and the support of ICL, refer to the work of Han et al.[199], and Banos et al., [200]. The data are stored in the DCL, from where they are recognized by the ICL and provided to the SCL for recommending the appropriate physical activity for the remaining targets.

### **7.5.2. Experimental setup**

To perform experiments, first experimental environment is set up, then the data and knowledge, required for the experiments, is specified and finally the evaluation criteria is defined.

#### **7.5.2.1. Environment**

The implementation of HRM was performed on a distributed framework in the Microsoft Azure public cloud environment. As described in section 7.3, the MM platform is composed of four layers, and each layer is deployed on an individual virtual instance. The proposed HRM is part of SCL, which was hosted on a standard A3 MS Azure instance with Microsoft Windows Server 2012 R2 as the guest Operating System (OS). HRM communicates with DCL and SL and communicates with DCL to load data for reasoning and storing final recommendations. With SL, HRM provides a recommendation service on the request and response model. The services in SCL are implemented as SOAP-based web services, and their accessibility is defined using service contracts between layers. Web services are implemented in Java and deployed on Glassfish server on virtual machine (VM).

For implementation of the third experiment, hybrid-CBR, which operates on METCB, we used myCBR<sup>7</sup>, which is an open-source similarity-based retrieval tool. We used the Windows environment on a PC with an Intel Pentium Dual-Core™ (2.5 GHz) with 4 GB of memory.

---

<sup>7</sup> <http://mycbr-project.net/index.html>

### 7.5.2.2. Data and knowledge (rules/case base)

As we evaluate our proposed hybrid-CBR methodology in terms of the performance of the baseline-RBR and modified-RBR systems, we therefore require data and knowledge on all of these systems. For the baseline-RBR and modified-RBR experiments, we used the user's personal profile, physical activity data and knowledge rules created based on the guidelines (Table 7.3, Table 7.4). For the hybrid-CBR experiments, we use METCB, prepared from METs guidelines [184]. The size of our 'METCB' is 119 instances. It contains the activities we focus on in the MM platform. The distribution of these activities in METCB is shown in Table 7.8.

**Table 7.8.** Distribution of the physical activities in the METs Case Base

S.No	Type of activity	Distribution
1	Running	25
2	Walking	56
3	Cycling	18
4	Standing	5
5	Sitting	4
6	Transportation	4
7	Volunteer	7
<b>Total instances</b>		<b>119</b>

In the compendium of physical activity guidelines [184], “standing” and “sitting” are the sub-categories of volunteer physical activity. More details on the structure of METCB are given in Table 7.5. For the offline testing and evaluation of the methodology, we designed a Test Case Base (TCB) that contains 64 test instances. We prepared these test cases from the original METCB. The method used for defining the value of the METs attribute of the TCB was random value computation. The random value is computed from the METs attribute of the original METCB using Microsoft Excel [201]. The function used for the random value generation is shown in equation 14.

$$\text{METs.value} = \text{randbetween}(\text{bottom}, \text{top}) \quad (14)$$

Here, bottom represents the minimum value of the METs and top represents the maximum value of METs for the new test cases. We used bottom = 1.3 and top = 23. The values 1.3 and 23 are the minimum and maximum values, respectively, of the METs attribute in the original METCB.

#### **7.5.2.3. Evaluation criteria**

To evaluate the proposed reasoning methodology, a group of system-centric evaluation criteria are used [202]. We evaluated the system using Type I (False positive-FP) and Type II (False negative-FN) errors, precision, recall, accuracy, and f-score criteria. We do not focus on a user-centric evaluation that addresses the user's satisfaction because in the current implementation, only a prototype of the MM platform is implemented. The hybrid-CBR experiments were performed in a closed environment in our lab; therefore, we leave user-centric evaluation as future work when the MM platform will be fully implemented with the feedback mechanism.

### **7.5.3. Experiments and Analysis of the Results**

As the design of HRM is based on RBR-first followed by the CBR strategy, we therefore first evaluate the RBR and then tailor its results to CBR. During the RBR execution, the level-1 RBR is first executed for reasoning the weight status of all of the subjects using Algorithm 1 and presenting the output as recommendations to the users, as shown in Table 7.9,. If the weight status is not underweight, the output is fed to level-2 RBR for setting goals and recommending weight loss and calorie consumption plans using Algorithm 2. The resulting recommendations of the level-2 RBR are also shown in Table 7.9.

**Table 7.9.** Output of level-1- and level-2 rule-based reasoning models

User ID	Level-1 RBR (algorithm1) Results		Level-2 RBR (algorithm 2) Results				
	BMI	Weight status	Ideal body weight (Kg)	Goal (# of Kg to lose)	Weight management plan	Duration plan (weeks)	Calories burning plan (daily)
1	23.9	normal	78.0	6.5	weight loss	13	550
2	25.02	overweight	64.8	7.7	weight loss	15	550
3	23.5	normal	66.6	3.5	weight loss	7	550
4	25.7	overweight	59.1	8.9	weight loss	18	550
5	25.8	overweight	62.9	9.0	weight loss	18	550
6	25.7	overweight	74.2	11.7	weight loss	23	550
7	26.2	overweight	52.0	13.0	weight loss	26	550
8	25.14	overweight	66.6	8.4	weight loss	17	550
9	30.24	obese	52.0	23.0	weight loss	46	550
10	23.8	normal	62.1	8.9	weight loss	18	550

These recommendations include the goal in terms of kg to lose, weight management plan, number of weeks to successfully execute the plan and daily calorie consumption plan. The volunteers were asked to follow these plan recommendations. The objective of HRM is to recommend appropriate physical activities for these plans. The HRM estimates METs values to materialize the plans. The METs estimation is required in two cases:

- At the start of plan, when HRM initially recommends the physical activity for starting the plan
- During the plan, i.e., the subject follows the plan and the system makes further recommendations

In the first case, the METs estimation is performed only for the recommended ‘daily calorie consumption plan’, which is the output of the level-2 RBR. In the second case, the METs estimation is based on the remaining calories (see equation 8). Once the METs value is computed, the corresponding physical activity recommendations are generated. These recommendations can be generated using the baseline-RBR, modified-RBR and hybrid-CBR systems; therefore, we perform three different sets of experiments, which are discussed below.

### 7.5.3.1. Experiment 1: Baseline-RBR system

The purpose of this experiment is to build the initial baseline-RBR system for comparing the results of the systems. The results of this experiment were generated prior to the implementation of the proposed idea in the MM platform. In level-3 RBR, distinct-METs rules, shown in Table 7.3, are used to generate physical activity recommendations using Algorithm 3 with exact match criteria. A few examples of the prescribed recommendations are shown in Table 7.10. These are based on the initial calorie consumption plan of the 10 volunteers.

**Table 7.10.** Recommendation of the baseline rule-based reasoning system

User ID	METs	Personalized physical activity recommendations	
1	6.5	i.	Climbing hills with 0 to 9 lb load.
		ii.	Race walking; rock or mountain climbing
2	7.6	X	
3	7.8	i.	backpacking; hiking or organized walking with a daypack
4	8.1	X	
5	7.6	X	
6	6.4	X	
7	8.5	i.	bicycling; BMX
		ii.	bicycling; mountain; general
		iii.	bicycling; 12 mph; seated; hands on brake hoods or bar drops; 80 rpm
8	7.3	i.	climbing hills with 10 to 20 lb load
9	7.3	i.	climbing hills with 10 to 20 lb load
10	7.7	X	

While generating these recommendations, the first METs values for all volunteers are computed based on their calorie plans and then combined with the attribute age group to prepare the data for the rules. The symbol ‘X’ in Table 7.10 denotes that no recommendation is generated for these query cases. From Table 7.10, it is clear that five out of ten queries cases are unsuccessful and that recommendations could not be generated for them. These include the queries of users 2, 4, 5, 6 and 10. The reasons for the empty recommendations are that these queries do not match any rule described in Table 7.3. The distinct rules used in this experiment use METs values adopted from the METs guideline for physical activity, which does not include the values 7.6, 8.1, 7.6, 6.4, and 7.7. Therefore, no rule with these values exists in Table 7.3, and hence, no match is found during the reasoning process for the specified input query cases. For the detailed evaluation of the baseline-RBR system, the whole ‘TCB’ is used as



a test case. The results are calculated and presented in Figure 7.7 and Figure 7.8, which show that the recall of the baseline-RBR is very low (45%) and that the Type II errors are very high (54.5%). The limitations of this experiment are summarized as follows: (1) creation of distinct rules for each value of METs is a difficult task that results in a rule intractability problem, (2) the closest similar recommendations are overlooked if an exact match is not found, and (3) a high Type II error rate is observed.

### **7.5.3.2. Experiment 2: Modified-RBR system**

Based on the lesson learnt from the baseline-RBR system, level-3 RBR is implemented with ranged-METs rules (Table 7.4) in the MM platform. Algorithm 3 is used to execute these rules. To demonstrate the effectiveness of this experiment, we consider an example query for volunteer 4 (Table 7.7) with age group = adults and METs = 8.1 (see Table 7.10). The modified-RBR generates multiple recommendations for this query, though baseline-RBR fails to do so. To fully evaluate Algorithm 3, the whole ‘TCB’ is applied, and the results produced are shown in Figure 7.7 and Figure 7.8. The recall and accuracy are increased from 0.45 to 0.89 and the f-score is increased from 0.62 to 0.66, while the Type II error rate is reduced from 54.7 to 10.9. The advantage of the modified-RBR system is that all queries are served and no query is returned with empty recommendation results. For example, when the query case with ‘age group’ = All Age and METs = 2.7 is processed, a total of 17 recommendations are generated, as shown in Table 7.11. When the baseline-RBR is used for this query, no recommendation is generated because the METs value of the query case has no match with the METs values of the distinct rules. However, in the modified-RBR, the ranged-METs rule with a METs value less than 3 is satisfied, and hence, all of the associated recommendations are generated. Similarly, all of the queries yields results, and no query is unsuccessful.

**Table 7.11.** Recommendations generated using modified rule-based reasoning system

Recommendation #	METs	Suggested physical activity recommendations
1	1.3	riding in a car or truck
2	1.3	riding in a bus or train
3	1.5	sitting; meeting; general; and/or with talking involved
4	1.5	sitting; light office work; in general
5	2.0	walking; household
6	2.0	walking; less than 2.0 mph; level; strolling; very slow
7	2	sitting; child care; only active periods
8	2	walking; less than 2.0 mph; very slow
9	2.3	carrying 15 lb child; slow walking
10	2.3	standing; light work (filing; talking; assembling)
11	2.5	bird watching; slow walk
12	2.5	walking from house to car or bus; from car or bus to go places; from car or bus to and from the worksite
13	2.5	walking to neighbor's house or family's house for social reasons
14	2.5	walking; to and from an outhouse
15	2.5	sitting; moderate work
16	2.5	automobile or light truck (not a semi) driving
17	2.8	walking; 2.0 mph; level; slow pace; firm surface

The limitation of the system is its high False Alarm rate (i.e., Type I error), as shown in Table 7.11. From this table, we see that a list of 17) recommendations is generated for a single query. On average, 52 options of physical activities are provided as recommendations for each query, which is problematic. A summary of the Type I error for this experiment is shown in Figure 7.8. The high False Alarm rate results in a wide scope of recommendations that may not fit well with the user's required physical activity. This effect is normalized in PBR when multiple filters are applied for filtering unnecessary and irrelevant recommendations.

### 7.5.3.3. Experiment 3: CBR system

The objective of using CBR is to minimize limitations of the baseline-RBR and modified-RBR systems. To overcome these problems, we performed the CBR experiment in a local set up without involving the MM setup. The outputs of level-1 RBR and level-2 RBR and the estimated METs value generate a query case for the

CBR methodology. Algorithm 4 uses the local similarity function, global similarity function, k-NN with  $k=3$  and a threshold  $\mu \geq 95$  to generate appropriate physical activity recommendations. The CBR methodology has significantly improved Type I and Type II errors, as shown in Figure 7.8. CBR offers the following advantages:

- Type I errors are reduced – k-NN with  $k=3$  retrieves the top cases that are most relevant to the query case and specific to the user's requirement. Hence, the False Alarm rate is significantly reduced.
- Type II errors are reduced and recall is improved – the global similarity function of CBR with threshold  $\mu \geq 95$  has reduced Type II errors. The retrieval of most similar recommendations minimized the False Negative cases and improved recall.
- Relevant and specific recommendations – the retrieve phase of CBR retrieves the top three recommendations that are either exactly the same as required by the user or close to the user's specific requirements for physical activity. Hence, the number of recommendations is reduced to an optimum level on the one hand and is closer to the user's specific requirements on the other.

To demonstrate the effectiveness of the CBR methodology for these objectives, we consider the case of 10 volunteers of the MM evaluation team and their estimated METs values (Table 7.10). The initial recommendations for the calculated METs values and age group=adults are shown in Table 7.12.

**Table 7.12.** Recommendations generated using case-based reasoning methodology

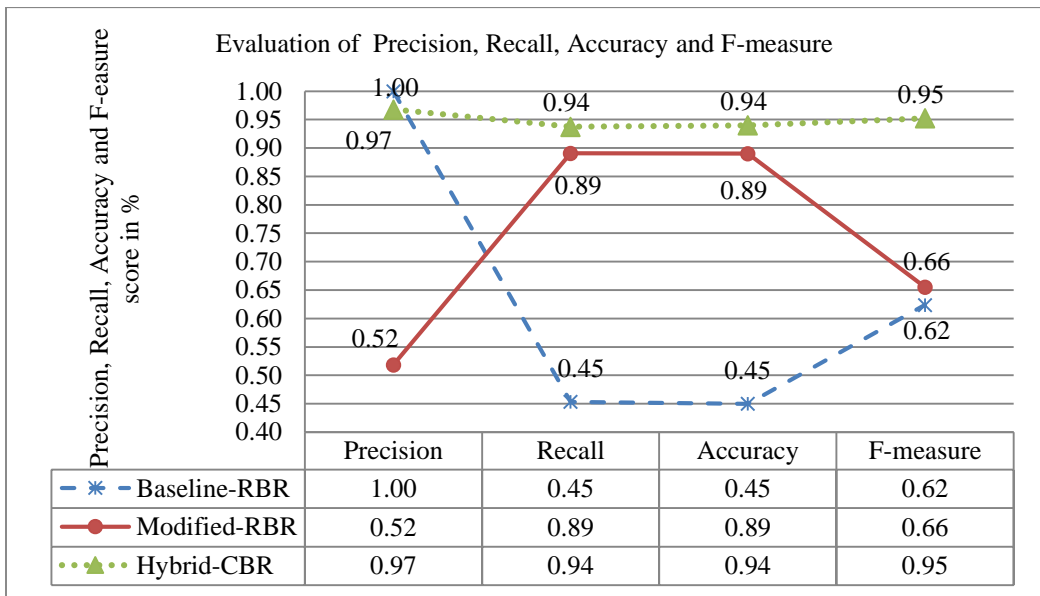
User ID	New case (METs value)	Retrieved cases (METs value)	Suggested physical activity recommendations	
1	6.5	6.5	i.	climbing hills with 0 to 9 lb load.
		6.5	ii.	race walking; rock or mountain climbing
		6.3	iii.	climbing hills; no load
2	7.6	7.3	i.	climbing hills with 10 to 20 lb load
		7.5	ii.	bicycling; general
		7.8	iii.	backpacking; hiking or organized walking with a daypack
3	7.8	7.8	i.	backpacking; hiking or organized walking with a daypack
		8	ii.	running; training; pushing a wheelchair or baby carrier
		8	iii.	running; marathon

User ID	New case (METs value)	Retrieved cases (METs value)	Suggested physical activity recommendations	
4	8.1	8	i.	running; training; pushing a wheelchair or baby carrier
		8	ii.	running; marathon
		8	iii.	carrying 25 to 49 lb load; upstairs
5	7.6	7.3	i.	climbing hills with 10 to 20 lb load
		7.5	ii.	bicycling; general
		7.8	iii.	backpacking; hiking or organized walking with a daypack
6	6.4	6.3	i.	climbing hills; no load
		6.5	ii.	climbing hills with 0 to 9 lb load
		6.5	iii.	race walking; rock or mountain climbing
7	8.5	8.5	i.	bicycling;
		8.5	ii.	bicycling; mountain; general
		8.5	iii.	bicycling; 12 mph; seated; hands on brake hoods or bar drops; 80 rpm
8	7.3	7	i.	walking; 4.5 mph; level; firm surface; very; very brisk
		7.3	ii.	climbing hills with 10 to 20 lb load
		7.5	iii.	bicycling; general
9	7.3	7	i.	walking; 4.5 mph; level; firm surface; very; very brisk
		7.3	ii.	climbing hills with 10 to 20 lb load
		7.5	iii.	bicycling; general
10	7.7	7.5	i.	bicycling; general
		7.8	ii.	backpacking; hiking or organized walking with a daypack
		8	iii.	bicycling; 12-13.9 mph; leisure; moderate effort

Table 7.12 shows that for each query case, the top three most relevant physical activity recommendations are provided, which fulfills the user's specific requirements. For the query age group = Adults and METs = 8.1, baseline-RBR failed to generate recommendations (see Table 7.10) and modified-RBR produced 59 possible recommendation options, but CBR produced only three recommendations (Table 7.12). The difference between the required METs values of the query case and the one using the rules is only 0.1, which is negligible; however, baseline-RBR fails to generate recommendations. This clearly shows the effectiveness of the proposed CBR methodology in HRM.

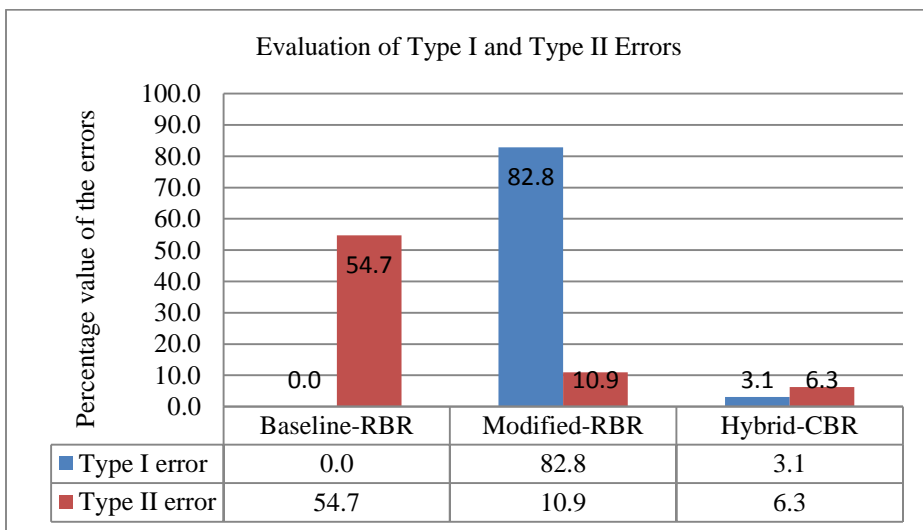
Moreover, to fully evaluate the CBR methodology, we apply the whole 'TCB' to generate recommendations. The results are shown in Figure 7.7 and Figure 7.8. These

results are significantly improved compared with those of the baseline-RBR and modified-RBR methodologies.



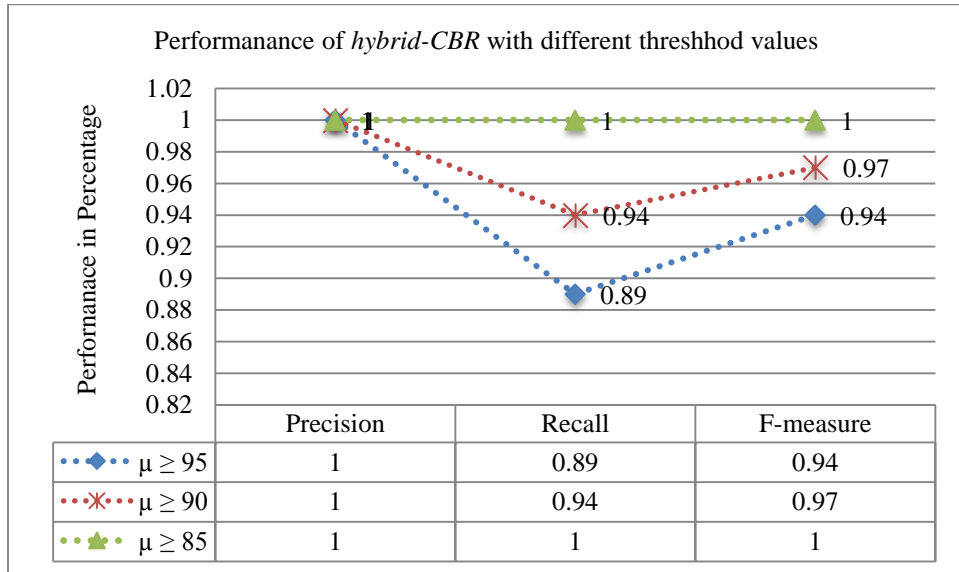
**Figure 7.7.** Comparison of baseline-RBR, modified-RBR and hybrid-CBR system

The green line at the top of the graph in Figure 7.7 shows the performance of hybrid-CBR, which is superior to the other two approaches.



**Figure 7.8.** Comparison of baseline-RBR, modified-RBR and hybrid-CBR

Figure 7.8 pictorially shows that hybrid RBR/CBR has improved Type I and Type II error results compared with the other experiments. To present the results of hybrid-CBR with different threshold values i.e.,  $\mu \geq 95$ ,  $\mu \geq 90$  and  $\mu \geq 85$ , we applied the ‘TCB’ and calculated the results, which are shown in Figure 7.9.



**Figure 7.9.** Performance of hybrid-CBR for different thresholds

Figure 7.9 shows that the proposed hybrid-CBR model produces 100% results for precision, recall, and F-score when the threshold  $\mu$  is taken as 85.

#### 7.5.4. Comparison of hybrid-CBR with jColibri

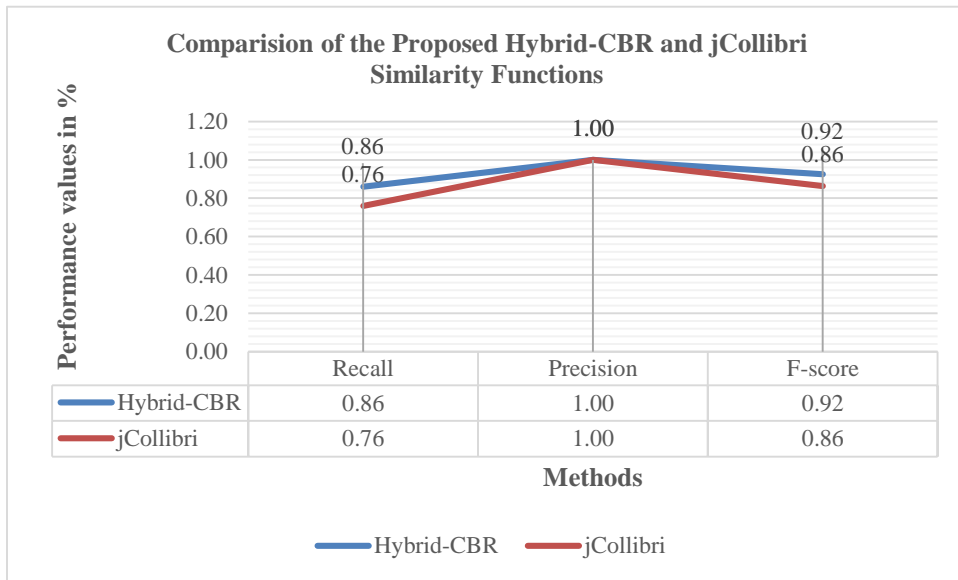
To compare results of the proposed hybrid-CBR with state-of-the-art CBR system, we selected jColibri 2.0, a case-based reasoning framework [152], which provides a reference implementation of the most commonly used similarity functions for building CBR systems used in different applications. We are motivated to jColibri as a comparison system because its similarity functions have open implementation and are more similar to our proposed hybrid-CBR’s local and global similarity functions [203]. We compare the equal and interval similarity functions, shown in equations of the jColibri with the hybrid-CBR functions defined in equation 11, 12 and 13.

$$\text{EqualSim}_1(nC_{ai}, eC_{ai}) = \begin{cases} d_1(nC_{ai}, eC_{ai}) = 1, & \text{if } nC_{ai} = eC_{ai} \\ 0, & \text{otherwise} \end{cases} \quad (15)$$

$$\text{IntervSim}_1(nC_{ai}, eC_{ai}) = 1 - \frac{d_1(nC_{ai}, eC_{ai})}{(\text{interval})} \quad (16)$$

For the global similarity, weighted sum function, described in equation 13 is used. The weights used in this equation are manually set by utilizing domain knowledge describing importance of the attributes of case base. The value of  $\beta$  is 0.1 (i.e.,  $\beta = 0.1$ ), and the value of  $\gamma$  is 0.9 (i.e.,  $\gamma = 0.9$ ). In the equation 16, the interval value is set to maximum value of the attribute.

A test case base of fifty input cases is used to test against the case base, containing around 120 resolved cases, using both the hybrid-CBR and jColibri similarity functions shown in equations 15, 16. We measured the performance of both the systems using precision, recall and consequently the f-score measure whose results are shown in Figure 7.10.



**Figure 7.10.** Performance of hybrid-CBR and jColibri similarity functions

The interpretations or analysis of the results shown in Figure 7.10 can be done as follows. The proposed hybrid-CBR has improved performance of case retrieval in

terms of recall/accuracy as compared to the jCollibri retrieval functions. The equal match function, equation 15, of the jCollibri system returns 1(100% match) if input case's attribute has same value as the existing case's attributes value, otherwise zero. However, there are situations where a set of values of an attribute constitute a super set, which can equally likely be applicable for the values of the subset as well, as the scenario is shown in equation 10 and Figure 7.6. In this scenario, which is practical in real-world applications, jCollibri's equal match function, equation 15, returns 0 if the value of a new case (nC) is not found in its own set. This function has the limitations of considering the applicability of the sub-set of values. Due to this reason, the case value is not matched, which should be, and consequently the contributions of this value in the overall weighted sum reduces its score and thus miss the actual case to be retrieved. The second, jCollibri interval match function, shown in equation 16, performs the same as the proposed hybrid-CBR function, provided the interval used is provided correctly. The proposed hybrid-CBR equal similarity function, shown in equation 11 and Table 7.6, can resolve these situations and hence improves the performance of the CBR retrieval phase.

## **7.6. Limitation of the proposed hybrid-CBR classifier**

As shown by the experiments and results, the proposed hybrid-CBR classifier performs well as compared to the baseline-RBR and modified-RBR as well as state-of-the-art jCollibri system, however it suffers from a number of limitations, as described below.

1. Without past successful/resolved cases and the availability of domain knowledge, accurate case authoring is challenging issue. The proposed guideline-enabled rule-based method used for the creation of training and new test cases is based on expert's rigorous inspection which is not efficient. A semi-automatic or automatic case preparation method is required to device.
2. By introducing new attribute in the case base, a new similarity function is required to be defined and tested prior to its real implementation in a real-world application.



3. For weighting the case attributes, i.e., quantifying preference levels of the attributes in the form of attributes weight, the presented method in this study is depended on domain experts' knowledge. We have not introduced any new method for automatically weighting the attributes, therefore a new weight estimation method is required.

## **7.7. Summary**

This chapter has presented the rationales behind the selection of hybrid-CBR classifier and its accurate design for the real-world application of physical activity recommendation generation. It also describes how effectively rule-based reasoning methodology is integrated with the CBR methodology to facilitate its adoption and extension in other application domains. The hybrid-CBR classifier achieves the objective of precise and specific recommendation decisions according to the specific requirements of the input test case, which is the user's specific needs formulated in query. The key features of the proposed hybrid-CBR discussed in this thesis are real-time case creation using the rule-based reasoning methodology that consumes the domain knowledge and the accurate similarity function that guarantee the accuracy of the recommendation decisions. The proposed classifier is tested in a weight management scenario and significant results are generated, which are measured in terms of precision, recall, accuracy, and f-score. The performance achieved is 0.97% precision, 0.94% recall, 0.94% accuracy, and a 0.95% f-score on a test dataset of 64 cases. Similarly, the Type I and Type II errors are significantly reduced as compared to the classical rule-based recommendation models, i.e., baseline-RBR and modified-RBR system built during this research. Furthermore, the proposed hybrid-CBR classifier can be easily extended to other application areas, which will increase its worth.

#### 8.1. Conclusion

This thesis focused on empirical performance evaluation of classification algorithms for classifiers ranking, meta-learning based automatic classifier selection, design of rough-set and hybrid-CBR classifiers and the associated issues, such as data acquisition for real-world datasets and cases preparation, semantics-preserving discretization, and accurate case similarity functions definitions. In first part of the thesis, focus is on the selection of accurate classifier selection using multi-criteria decision making and meta-learning and reasoning approaches. In the multi-criteria decision making, the thesis has proposed and developed an accurate multi-metric decision making methodology (AMD) which correctly recommends suitable classification algorithm for structured and prepared dataset. In AMD, first, a concepts of algorithms' quality meta-metrics (QMM) is proposed which describes physical meanings of the evaluation criteria, and a classification model, referred as classifiers quality meta-metrics model, is developed for it. This model helps experts in the selection of suitable evaluation criteria for comparison of the classifiers. Based on the experts' consensus, expert's grouped-based decision making method is developed for the selection of suitable evaluation metrics from a large set of evaluation metrics. A set of suitable evaluation metrics are identified for the comparison of results of the heterogeneous classifiers from the perspectives of speed, accuracy and consistency. Furthermore, a relative criteria weighting technique is developed, based on the AHP method, for consistently weighting the evaluation metrics. The analysis of the performance of classification algorithms is performed using statistical significance test the fitness evaluation function. The algorithms are ranked by computing the relative closeness value of all the algorithms with respect to the ideal ranking, using

the AHP-based estimated weights and local and global constraints on the evaluation criteria.

In the meta-learning based classifier selection method, a CBR-based meta-learning and reasoning (CBR-MLR) framework is proposed for accurate classifier selection using data characteristics, called meta-features, and classifiers characteristics, called performance metrics. The relationship of data and classifiers characteristics is represented as cases to form a training dataset, called Case-Base, for a CBR classifiers. The recommendations of an accurate classifier for a new case or test dataset is performed using the CBR multi-view, multi-level reasoning methods, developed as part of the proposed framework. In this approach, a set of four view of data characteristics are introduced and represented. These are: general characteristics, basic statistical, advanced statistical and information-theoretic families. These families represents the datasets from multiple aspects and are thus a good representative set of characteristics for building a model. The candidate nine decision tree classifiers, considered for this study, are taken from Weka environment with their default settings. In the online recommendation part, the CBR standard methodology is enhanced with accurate similarity functions and a post processing classifier conflict resolution methods to recommend the most appropriate classifier for a given new dataset or learning problem. The methodology is tested on 52 test datasets, taken from UCI/OpenML repositories, which has produced overall accuracy of 94%.

In the second part of the thesis, expert's heuristics based approach is used for selection of the classifiers and two accurate rough-set and hybrid-CBR classifiers are designed. The rough set classifier is developed for a real-world application scenario of the diabetes mellitus where the data is scattered in patients notes. Domain specific guidelines-enabled approach is used for structuring the data in the rough set information system format. The discretization phase is enhanced by introducing a semantics-preserving discretization scheme that has preserved the semantics of actual data in the rules. The rough set classifier's selection is made based on its capabilities of building an accurate comprehensible and interpretable model with the best approximation capabilities of the rough boundaries of different classes in the dataset.

Furthermore, the thesis has also built an accurate and precise hybrid-CBR classifier. The proposed hybrid-CBR classifier is supported with an enhanced rule-based mechanism for case preparation, which consumes the domain specific guidelines for preparing accurate resolved cases. Accurate similarity functions are defined which increase the accuracy of matching new case against the existing cases.

The evaluation results and comparison of the AMD and CBR-MLR methodologies, rough set and hybrid-CBR classifiers, with state-of-the-art methods, have shown that the proposed methods perform significantly better than the existing methods. The AMD has achieved 0.97 Spearman's rank correlation coefficient ( $R_s$ ) on 15 test datasets using 35 classification algorithms and CBR-MLR has achieved 94% accuracy of correct classifier recommendation, in the scope of top  $k=3$  best classifiers, using 100 training and 52 test datasets from the UCI/OpenML repositories. Similarly, rough set classifier has achieved 0.95% classification accuracy on a diabetes dataset and hybrid-CBR has achieved 0.97% precision, 0.94% recall, 0.94% accuracy, and 0.95% f-score on a physical activity dataset.

## 8.2. Future Directions

The thesis has presented four methods, the first two solely for the selection of accurate classifiers and the last two for the both selection and design of accurate classifiers. The first two are based on automatic methods AMD and CBR-MLR, while the last two are based on rough-set and hybrid-CBR methods. The expected future extensions in each of these method is given below.

### 8.2.1. Future perspective of AMD method

- **Automatic criteria selection:** The proposed method has provided minimum support for the automatic criteria selection. A partially automatic solution, in the form of classifiers quality meta-metric classification model, is provided, however it is not enough to reduce the experts' efforts and time. To resolve this issue an advanced method is required to minimize the experts' time and efforts by introducing a semi-automatic analysis method for analyzing the classifiers

performance metrics against the goal and constraints defined by the end user for his/her application.

- **New method for criteria weighting:** The AMD methodology uses relative criteria weighting mechanism which is a semi-automatic way requiring experts' preferences for quantifying their opinion in the form of weights. However, experts' availability is not always guaranteed, therefore, in future, we plan to introduce and design new methods for estimating criteria weights.
- **Exhaustive search:** The proposed method is based on exhaustive search mechanism to rank algorithms and finally select a single one for the application in hand. In future, we plan to introduce a hierarchical searching mechanism with multi-level filtration to filter-out the most unfit classifier from the competition and reduce the search scope for recommending suitable algorithm.

### 8.2.2. Future perspective of CBR-MLR method

- **Finding an optimum and suitable set of meta-features:** the process of finding right classifier for a dataset using a machine learning model that is based only on the datasets global features is not enough and may lead to a wrong decision. The reason is that the proposed 29 features for the selection of classifier does not represents the whole meta-feature space of the datasets and thus cannot be declared as the final optimum list of features. Therefore, in future we plan to perform an extensive set of experiments using statistical, information-theoretic, landmarking, model-based and complexity-based meta-characteristics and find an optimum set of meta-characteristics for best estimating data qualities, required for automatic selection of classifiers.
- **Classifiers performance analysis for finding class label:** while creating the successful cases, the proposed method analyses the performance results of the candidates classifiers using predictive accuracy and standard deviation, however this evaluation is application dependent. The users may interested in other characteristics of the classifiers to be selected. In that case, the proposed Case-

Base may not work well for them and need to be updated according to their application requirements, which is an exhaustive experimental work. To overcome this issue, in future we plan to introduce an efficient method to automatically or semi-automatically perform this analysis and produce the class label of the resolved cases.

- **Ranking classifiers with similar performance results (tie cases):** while analyzing the performance results of the classifiers for finding the best classifiers to make them class-labels, we perform the process of ranking the classifier. However, in case of small datasets most of the classifier perform with equal performance and are thus ranked same. This makes the process more complex because each dataset has the list of almost all the candidates' classifiers as the class labels, which makes the problem of classifier selection as a multi-label learning problem. However, the correct solution has no such strategy to properly address this situations. We simply create multiple cases with the same problem description part (i.e., meta-features list) and different class labels, each for a classifier with same rank. In future we plan to design sophisticated multilevel analysis of the classifiers and introduce multilevel learning.

### 8.2.3. Future perspective of rough-set classifier

- **Automatic data extraction:** In future the plan is to introduce an automatic data extraction method to extract information from the patients' charts, in order to eliminate the dependency on domain experts.
- **Conflict resolution of approximate rules:** approximate rules generated by the rough set classifier needs further processing using domain knowledge to reach accurate and more correct decision without any ambiguity. The future plan is to introduce a second level of evaluation of the conflicting rules by utilizing domain knowledge or some priority scheme, based on the rule conditions attributes.

### 8.2.4. Future perspective of hybrid-CBR classifier

- **Semi-automatic or automatic case preparation:** Without past successful/resolved cases and the availability of domain knowledge, accurate case authoring is challenging issue. The proposed guideline-enabled rule-based method used for the creation of training and new test cases is based on expert's rigorous inspection which is not efficient. We plan to introduce a semi-automatic or automatic case preparation method to reduce the experts' time and efforts in preparing these cases.
- **Automatic weight assignment:** For weighting the case attributes, i.e., quantifying preference levels of the attributes in the form of attributes weight, the presented method is depended on domain experts' knowledge. Our future plan is to introduce a new method for automatically or semi-automatically weighting the attributes, based on the relative score rather than absolute values of weights.

## Bibliography

---

1. Neslihan, D.; Zuhail, T., A comparative analysis of classification algorithms in data mining for accuracy, speed and robustness. *Inf. Technol. and Management* **2013**, 14, (2), 105-124.
2. Lavesson, N.; Boeva, V.; Tsiorkova, E.; Davidsson, P., A method for evaluation of learning components. *Automated Software Engineering* **2014**, 21, (1), 41-63.
3. Lavesson, N.; Davidsson, P. In *Analysis of multi-criteria methods for algorithm and classifier evaluation*, Proceedings of the 24th annual workshop of the Swedish Artificial Intelligence Society, 2007; Linköping Bors, Sweden: 2007; pp 11-22.
4. Kou, G.; Lu, Y.; Peng, Y.; Shi, Y., Evaluation of classification algorithms using MCDM and rank correlation. *International Journal of Information Technology & Decision Making* **2012**, 11, (01), 197-225.
5. Lemnaru, E. C. Strategies for dealing with real world classification problems. Technical University of Cluj-Napoca, 2012.
6. Wolpert, D. H.; Macready, W. G., No free lunch theorems for optimization. *Evolutionary Computation, IEEE Transactions on* **1997**, 1, (1), 67-82.
7. Pawlak, Z. a., Rough sets. *International Journal of Computer & Information Sciences* **1982**, 11, (5), 341-356.
8. Pawlak, Z. a.; Skowron, A., Rough sets: some extensions. *Information sciences* **2007**, 177, (1), 28-40.
9. Muthuraman, T.; Sankaran, G., A framework for personalized decision support system for the healthcare application. **2014**.
10. Soni, S.; Vyas, O. P., Fuzzy Weighted Associative Classifier: A Predictive Technique For Health Care DataMining. *International Journal of Computer Science, Engineering and Information Technology (IJCEIT)* **2012**, 2, (1), 11-22.
11. Bonikowski, Z.; Bryniarski, E.; Wybraniec-Skardowska, U., Extensions and intentions in the rough set theory. *Information sciences* **1998**, 107, (1), 149-167.
12. Kumar, P.; Sehgal, V. K.; Chauhan, D. S., A benchmark to select data mining based classification algorithms for business intelligence and decision support systems. *arXiv preprint arXiv:1210.3139* **2012**.
13. Luo, G., A review of automatic selection methods for machine learning algorithms and hyper-parameter values. In Available at [http://pages.cs.wisc.edu/~gangluo/automatic\\_selection\\_review.pdf](http://pages.cs.wisc.edu/~gangluo/automatic_selection_review.pdf): 2015.
14. Berrer, H.; Paterson, I.; Keller, J. r. In *Evaluation of machine-learning algorithm ranking advisors*, In Proceedings of the PKDD-2000 Workshop on



- DataMining, Decision Support, Meta-Learning and ILP: Forum for Practical Problem Presentation and Prospective Solutions, 2000; Citeseer: 2000.
15. Caruana, R.; Niculescu-Mizil, A. In *An empirical comparison of supervised learning algorithms*, Proceedings of the 23rd international conference on Machine learning, 2006; ACM: 2006; pp 161-168.
16. Fawcett, T., An introduction to ROC analysis. *Pattern Recognition Letters* **2006**, 27, (8), 861-874.
17. Saaty, T. L., Decision-making with the AHP: Why is the principal eigenvector necessary. *European journal of operational research* **2003**, 145, (1), 85-91.
18. Soares, C.; Costa, J.; Brazdil, P. In *A simple and intuitive measure for multicriteria evaluation of classification algorithms*, ECML2000 Workshop on Meta-Learning: Building Automatic Advice Strategies for Model Selection and Method Combination, 2000; Citeseer: 2000; pp 87-96.
19. Andersson, A.; Davidsson, P.; Lindén, J., Measure-based classifier performance evaluation. *Pattern Recognition Letters* **1999**, 20, (11), 1165-1173.
20. Elazmeh, W.; Japkowicz, N.; Matwin, S., A framework for measuring classification difference with imbalance (technical report ws-06-06). In AAAI press, Menlo Park: 2006.
21. Wang, G.; Song, Q.; Zhang, X.; Zhang, K., A Generic Multilabel Learning-Based Classification Algorithm Recommendation Method. *ACM Transactions on Knowledge Discovery from Data (TKDD)* **2014**, 9, (1), 7.
22. Zavadskas, E. K.; Zakarevicius, A.; Antucheviciene, J., Evaluation of ranking accuracy in multi-criteria decisions. *Informatica* **2006**, 17, (4), 601-618.
23. Hwang, C.-L.; Yoon, K., *Multiple attribute decision making: methods and applications a state-of-the-art survey*. Springer Science & Business Media: 2012; Vol. 186.
24. Freitas, A. A., A critical review of multi-objective optimization in data mining: a position paper. *ACM SIGKDD Explorations Newsletter* **2004**, 6, (2), 77-86.
25. Aha, D. W. In *Generalizing from case studies: A case study*, Proc. of the 9th International Conference on Machine Learning, 1992; 1992; pp 1-10.
26. Brodley, C. E. In *Addressing the selective superiority problem: Automatic algorithm/model class selection*, Proceedings of the Tenth International Conference on Machine Learning, 1993; Citeseer: 1993; pp 17-24.
27. Gama, J.; Brazdil, P., Characterization of classification algorithms. In *Progress in Artificial Intelligence*, Springer: 1995; pp 189-200.
28. Lindner, G.; Studer, R., AST: Support for algorithm selection with a CBR approach. In *Principles of data mining and knowledge discovery*, Springer: 1999; pp 418-423.
29. Alexandros, K.; Melanie, H., Model selection via meta-learning: a comparative study. *International Journal on Artificial Intelligence Tools* **2001**, 10, (04), 525-554.

30. Smith, K. A.; Woo, F.; Ciesielski, V.; Ibrahim, R., Matching data mining algorithm suitability to data characteristics using a self-organizing map. In *Hybrid information systems*, Springer: 2002; pp 169-179.
31. Lim, T.-S.; Loh, W.-Y.; Shih, Y.-S., A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms. *Machine Learning* **2000**, 40, (3), 203-228.
32. Brazdil, P. B.; Soares, C.; Da Costa, J. P., Ranking learning algorithms: Using IBL and meta-learning on accuracy and time results. *Machine Learning* **2003**, 50, (3), 251-277.
33. Ali, S.; Smith-Miles, K. A., A meta-learning approach to automatic kernel selection for support vector machines. *Neurocomputing* **2006**, 70, (1), 173-186.
34. Romero, C.; Olmo, J. L.; Ventura, S. In *A meta-learning approach for recommending a subset of white-box classification algorithms for Moodle datasets*, EDM, 2013; 2013; pp 268-271.
35. Ali, S.; Smith, K. A., On learning algorithm selection for classification. *Applied Soft Computing* **2006**, 6, (2), 119-138.
36. Singh, P.; Verma, A.; Chaudhari, N. S., Performance Evaluation of Classifier Combination Techniques for the Handwritten Devanagari Character Recognition. In *Information Systems Design and Intelligent Applications*, Springer: 2016; pp 651-662.
37. Singh, A.; Singh, M. L., Performance evaluation of various classifiers for color prediction of rice paddy plant leaf. *Journal of Electronic Imaging* **2016**, 25, (6), 061403-061403.
38. Perveen, S.; Shahbaz, M.; Guergachi, A.; Keshavjee, K., Performance Analysis of Data Mining Classification Techniques to Predict Diabetes. *Procedia Computer Science* **2016**, 82, 115-121.
39. Kandhasamy, J. P.; Balamurali, S., Performance analysis of classifier models to predict diabetes mellitus. *Procedia Computer Science* **2015**, 47, 45-51.
40. Song, Q.; Wang, G.; Wang, C., Automatic recommendation of classification algorithms based on data set characteristics. *Pattern recognition* **2012**, 45, (7), 2672-2689.
41. Reif, M.; Shafait, F.; Goldstein, M.; Breuel, T.; Dengel, A., Automatic classifier selection for non-experts. *Pattern Analysis and Applications* **2014**, 17, (1), 83-96.
42. Gayen, A. K., The frequency distribution of the product-moment correlation coefficient in random samples of any size drawn from non-normal universes. *Biometrika* **1951**, 38, (1/2), 219-247.
43. Khanmohammadi, S.; Rezaeiahari, M., AHP based Classification Algorithm Selection for Clinical Decision Support System Development. *Procedia Computer Science* **2014**, 36, 328-334.
44. Tzeng, G.-H.; Huang, J.-J., *Multiple attribute decision making: methods and applications*. CRC press: 2011 Jun 22.

45. Figueira, J.; Greco, S.; Ehrgott, M., *Multiple criteria decision analysis: state of the art surveys*. Springer Science & Business Media: 2005; Vol. 78.
46. Wolpert, D. H.; Macready, W. G., No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation* **1997**, 1, (1), 67-82.
47. Aha, D. W. In *Generalizing from Case studies: A Case Study*, Ninth International Conference on Machine Learning, 1992; Citeseer: 1992; pp 1-10.
48. Smith-Miles, K. A., Cross-disciplinary perspectives on meta-learning for algorithm selection. *ACM Computing Surveys (CSUR)* **2008**, 41, (1), 1-25.
49. Brazdil, P.; Gama, J. o.; Henery, B. In *Characterizing the applicability of classification algorithms using meta-level learning*, European Conference on Machine Learning: ECML-94, 1994; Springer: 1994; pp 83-102.
50. Bernado-Mansilla, E.; Ho, T. K., Domain of competence of XCS classifier system in complexity measurement space. *Evolutionary Computation, IEEE Transactions on* **2005**, 9, (1), 82-104.
51. Bache, K.; Lichman, M., UCI Machine Learning Repository [<http://archive.ics.uci.edu/ml>]. In Irvine, CA: University of California, School of Information and Computer Science.: 2013.
52. Sun, M.; Wong, D.; Kronenfeld, B., A heuristic multi-criteria classification approach incorporating data quality information for choropleth mapping. *Cartography and Geographic Information Science* **2016**, 1-13.
53. Yukun Chen, M. S.; Warren Clayton, M. D. In *Type 2 Diabetes Risk Forecasting from EMR Data using Machine Learning*, AMIA Annu Symp, 2012; 2012; pp 606–615.
54. Zolfaghari, R., Diagnosis of Diabetes in Female Population of Pima Indian Heritage with Ensemble of BP Neural Network and SVM. *International Journal of Computational Engineering & Management* **2012**, 15, (4).
55. Stahl, F. Diabetes Mellitus Glucose Prediction by Linear and Bayesian Ensemble Modeling. Sweden, 2012.
56. Narasingarao, M.; Manda, R.; Sridhar, G.; Madhu, K.; Rao, A., A clinical decision support system using multilayer perceptron neural network to assess well being in diabetes. *The Journal of the Association of Physicians India* **2009**, 57, 127-133.
57. Thirugnanam, M.; Kumar, P.; Srivatsan, S. V.; Nerlesh, C. R., Improving the Prediction Rate of Diabetes Diagnosis Using Fuzzy, Neural Network, Case Based (FNC) Approach. *Procedia Engineering* **2012**, 38, 1709-1718.
58. Chen, H.; Tan, C., Prediction of type-2 diabetes based on several element levels in blood and chemometrics. *Biological trace element research* **2012**, 147, (1-3), 67-74.
59. Sood, A.; Diamond, S.; Wang, S., Type 2 Diabetes Mellitus Classification. In Department of Computer Science, Stanford University: 2012.
60. Pobi, S., A study of machine learning performance in the prediction of juvenile diabetes from clinical test results. **2006**.

61. Ali, R.; Siddiqi, M. H.; Idris, M.; Kang, B. H.; Lee, S., Prediction of Diabetes Mellitus Based on Boosting Ensemble Modeling. In *Ubiquitous Computing and Ambient Intelligence. Personalisation and User Adapted Services*, Springer: 2014; pp 25-28.
62. Hashemi, R. R.; Jelovsek, F. R.; Razzaghi, M., Developmental toxicity risk assessment: a rough sets approach. *Methods of information in medicine* **1993**, 32, (1), 47-54.
63. Tsumoto, S. In *Automated knowledge acquisition from clinical databases based on rough sets and attribute-oriented generalization*, Proceedings of the AMIA Symposium, 1998; American Medical Informatics Association: 1998; p 548.
64. Paterson, G. I., A rough sets approach to patient classification in medical records. *Medinfo. MEDINFO* **1994**, 8, 910-910.
65. Komorowski, J.; Āhrn, A., Modelling prognostic power of cardiac tests using rough sets. *Artificial Intelligence in Medicine* **1999**, 15, (2), 167-191.
66. Āhrn, A., Discernibility and rough sets in medicine: tools and applications. **2000**.
67. Lichman, M., UCI Machine Learning Repository [<http://archive.ics.uci.edu/ml>]  
In Irvine, CA: University of California, School of Information and Computer Science: 2013.
68. Breault, J. L. In *Data mining diabetics databases: Are rough sets a useful addition*, Proceedings of the 33rd Symposium on Interface, Computing Science and Statistics, 2001; 2001.
69. Hassanien, A. E.; Abdelhafez, M. E.; Own, H. S., Rough sets data analysis in knowledge discovery: a case of kuwaiti diabetic children patients. *Advances in fuzzy Systems* **2008**, 8, 2.
70. Deja, R., Accuracy evaluation of the system of type 1 diabetes prediction. In *Rough Sets and Knowledge Technology*, Springer: 2011; pp 321-326.
71. Stepianiuk, J., Rough Set Data Mining of Diabetes Mellitus Data. **1999**.
72. Abdul-Ghani, M. A.; Abdul-Ghani, T.; Stern, M. P.; Karavic, J.; Tuomi, T.; Bo, I.; DeFronzo, R. A.; Groop, L., Two-step approach for the prediction of future type 2 diabetes risk. *Diabetes care* **2011**, 34, (9), 2108-2112.
73. Guasch-Ferre, M.; Bullo, M.; Costa, B.; Martinez-Gonzalez, M. A.; Ibarrola-Jurado, N.; Estruch, R.; Barrio, F.; Salas-Salvado, J.; Investigators, P.-P., A risk score to predict type 2 diabetes mellitus in an elderly Spanish Mediterranean population at high cardiovascular risk. *PLoS one* **2012**, 7, (3), e33437.
74. Collins, G. S.; Mallett, S.; Omar, O.; Yu, L.-M., Developing risk prediction models for type 2 diabetes: a systematic review of methodology and reporting. *BMC medicine* **2011**, 9, (1), 103.

75. Tabaei, B. P.; Herman, W. H., A multivariate logistic regression equation to screen for diabetes development and validation. *Diabetes care* **2002**, 25, (11), 1999-2003.
76. Shen, J.; Xing, J.; Xu, M. In *Research on CBR-RBR Fusion Reasoning Model and its Application in Medical Treatment*, Proceedings of the 21st International Conference on Industrial Engineering and Engineering Management 2014; Springer: 2014; pp 431-434.
77. Rossille, D.; Laurent, J.-F. o.; Burgun, A., Modelling a decision-support system for oncology using rule-based and case-based reasoning methodologies. *International journal of medical informatics* **2005**, 74, (2), 299-306.
78. Khandelwal, K.; Sharma, D. P., Hybrid Reasoning Model for Strengthening the problem solving capability of Expert Systems. *International Journal of Advanced Computer Science and Applications (IJACSA)* **2013**, 4, (10).
79. Park, H.-J.; Oh, J.-S.; Jeong, D.-U.; Park, K.-S., Automated sleep stage scoring using hybrid rule-and case-based reasoning. *Computers and Biomedical Research* **2000**, 33, (5), 330-349.
80. Petot, G. J.; Marling, C.; Sterling, L., An artificial intelligence system for computer-assisted menu planning. *Journal of the American Dietetic Association* **1998**, 98, (9), 1009-1014.
81. Montani, S.; Bellazzi, R., Supporting decisions in medical applications: the knowledge management perspective. *International journal of medical informatics* **2002**, 68, (1), 79-90.
82. Evans-Romaine, K.; Marling, C. In *Prescribing exercise regimens for cardiac and pulmonary disease patients with CBR*, Workshop on CBR in the health sciences at 5th international conference on case-based reasoning (ICCBR-03), 2003; Citeseer: 2003; pp 45-62.
83. Berka, P., NEST: a compositional approach to rule-based and case-based reasoning. *Advances in Artificial Intelligence* **2011**, 2011, 4.
84. Bichindaritz, I.; Siadak, M. F.; Jocom, J.; Moinpour, C.; Kansu, E.; Donaldson, G.; Bush, N.; Chapko, M.; Bradshaw, J. M.; Sullivan, K. M. In *CARE-PARTNER: a computerized knowledge-support system for stem-cell post-transplant long-term follow-up on the World-Wide-Web*, Proceedings of the AMIA Symposium, 1998; American Medical Informatics Association: 1998; p 386.
85. Lieber, J.; D'Aquin, M.; Bey, P.; Bresson, B.; Croissant, O.; Falzon, P.; Lesur, A.; Julien; Mollo, V.; Napoli, A. In *The kasimir project: knowledge management in cancerology*, 4th International Workshop on Enterprise Networking and Computing in Health Care Industry-HealthCom 2002, 2002; 2002; pp 125--127.
86. Van Meteren, R.; Van Someren, M. In *Using content-based filtering for recommendation*, Proceedings of the Machine Learning in the New Information Age: MLnet/ECML2000 Workshop, 2000; 2000; pp 47-56.

87. Davis, D. A.; Chawla, N. V.; Christakis, N. A.; Barabási, A.-L. s., Time to CARE: a collaborative engine for practical disease prediction. *Data Mining and Knowledge Discovery* **2010**, 20, (3), 388-415.
88. Hasan, S.; Duncan, G. T.; Neill, D. B.; Padman, R. In *Towards a collaborative filtering approach to medication reconciliation*, AMIA Annu Symp Proc, 2008; 2008; pp 288-92.
89. Satzger, B.; Endres, M.; Kießling, W., A preference-based recommender system. In *E-Commerce and Web Technologies*, Springer: 2006; pp 31-40.
90. Wang, W. M.; Cheung, C. F.; Lee, W. B.; Kwok, S. K., Knowledge-based treatment planning for adolescent early intervention of mental healthcare: a hybrid case-based reasoning approach. *Expert Systems* **2007**, 24, (4), 232-251.
91. Montani, S.; Magni, P.; Bellazzi, R.; Larizza, C.; Roudsari, A. V.; Carson, E. R., Integrating model-based decision support in a multi-modal reasoning system for managing type 1 diabetic patients. *Artificial Intelligence in Medicine* **2003**, 29, (1), 131-151.
92. Nguyen, L.; Sun, Z.; Stranieri, A.; Firmin, S. In *CWDM: A case-based diabetes management web system*, 24th Australasian Conference on Information Systems (ACIS), 2013; RMIT University: 2013; pp 1-11.
93. Ali, R.; Hussain, J.; Siddiqi, M. H.; Hussain, M.; Lee, S., H2RM: A Hybrid Rough Set Reasoning Model for Prediction and Management of Diabetes Mellitus. *Sensors* **2015**, 15, (7), 15921-15951.
94. Ali, R.; Siddiqi, M. H.; Lee, S.; Kang, B. H. In *KARE: A Hybrid Reasoning Approach for Promoting Active Lifestyle*, 9th International Conference of Ubiquitous Information Management and Communication ICUIMC (IMCOM) Bali, Indonesia, 8-10 January, 2015; Bali, Indonesia, 2015.
95. Fahim, M.; Idris, M.; Ali, R.; Nugent, C.; Kang, B.; Huh, E.-N.; Lee, S., ATHENA: A Personalized Platform to Promote an Active Lifestyle and Wellbeing Based on Physical, Mental and Social Health Primitives. *Sensors* **2014**, 14, (5), 9313-9329.
96. Nassabi, M. H.; op den Akker, H.; Vollenbroek, M. In *An ontology-based recommender system to promote physical activity for pre-frail elderly*, Mensch & Computer, 2014; Walter de Gruyter GmbH & Co KG: 2014; p 181.
97. Marling, C.; Petot, G.; Sterling, L. In *A CBR/RBR hybrid for designing nutritional menus*, Multimodal Reasoning: Papers from the 1998 AAAI Spring Symposium, 1998; AAAI Press, Menlo Park: 1998.
98. Wang, D.; Yu, M.; Low, C. B.; Arogeti, S., *Model-based health monitoring of hybrid systems*. Springer: 2013.
99. Aamodt, A.; Plaza, E., Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI communications* **1994**, 7, (1), 39-59.
100. Marling, C.; Rissland, E.; Aamodt, A., Integrations with case-based reasoning. *The Knowledge Engineering Review* **2005**, 20, (03), 241-245.

101. Bentley, J. L., *Writing efficient programs*. Prentice-Hall, Inc.: 1982.
102. Witten, I. H.; Frank, E., *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann: 2005.
103. Han, J.; Kamber, M.; Pei, J., *Data mining: concepts and techniques*. Elsevier: 2011.
104. Alpaydin, E., *Introduction to machine learning*. MIT press: 2014.
105. Kotsiantis, S. B.; Zaharakis, I.; Pintelas, P., Supervised machine learning: A review of classification techniques. In 2007; pp 3-24.
106. Apte', C.; Weiss, S., Data mining with decision trees and decision rules. *Future generation computer systems* **1997**, 13, (2), 197-210.
107. Rokach, L., Ensemble-based classifiers. *Artificial Intelligence Review* **2010**, 33, (1-2), 1-39.
108. Kolodner, J., *Case-based reasoning*. Morgan Kaufmann: 2014.
109. Pohekar, S. D.; Ramachandran, M., Application of multi-criteria decision making to sustainable energy planning—a review. *Renewable and sustainable energy reviews* **2004**, 8, (4), 365-381.
110. Jahanshahloo, G. R.; Lotfi, F. H.; Izadikhah, M., Extension of the TOPSIS method for decision-making problems with fuzzy data. *Applied Mathematics and Computation* **2006**, 181, (2), 1544-1551.
111. Lemke, C.; Budka, M.; Gabrys, B., Metalearning: a survey of trends and technologies. *Artificial intelligence review* **2015**, 44, (1), 117-130.
112. Vilalta, R.; Giraud-Carrier, C.; Brazdil, P., Meta-learning-concepts and techniques. In *Data mining and knowledge discovery handbook*, Springer: 2009; pp 717-731.
113. Brazdil, P.; Carrier, C. G.; Soares, C.; Vilalta, R., *Metalearning: Applications to data mining*. Springer Science & Business Media: 2008.
114. Anderson, M. L.; Oates, T., A review of recent research in metareasoning and metalearning. *AI Magazine* **2007**, 28, (1), 12.
115. Lichman, M., UCI machine learning repository [<http://archive.ics.uci.edu/ml>]. Irvine, CA: University of California, School of Information and Computer Science. In 2013.
116. Stone, M., Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society. Series B (Methodological)* **1974**, 111-147.
117. Freitas, A. A., Are we really discovering interesting knowledge from data. *Expert Update (the BCS-SGAI Magazine)* **2006**, 9, (1), 41-47.
118. Buciluă, C.; Caruana, R.; Niculescu-Mizil, A. In *Model compression*, Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, 2006; ACM: 2006; pp 535-541.
119. Pinto, M. r. P. *Microsoft Excel 2010*, 14.0.4760.1000 (34-bit), ; Microsoft Corporation: 2010.

120. Freitas, A. In *On objective measures of rule interestingness*, Second european symposium on principles of data mining & knowledge discovery, 1998; 1998.
121. Gaines, B. R., Transforming rules and trees into comprehensible knowledge structures. *Advances in Knowledge discovery and Data mining* **1996**, 205-226.
122. Bouckaert, R. R.; Frank, E.; Hall, M. A.; Holmes, G.; Pfahringer, B.; Reutemann, P.; Witten, I. H., WEKA---Experiences with a Java Open-Source Project. *The Journal of Machine Learning Research* **2010**, 11, 2533-2541.
123. Ferri, C.; Hernández-Orallo, J.; Modroiu, R., An experimental comparison of performance measures for classification. *Pattern Recognition Letters* **2009**, 30, (1), 27-38.
124. Sprinkhuizen-Kuyper, I. G.; Smirnov, E. N.; Nalbantov, G. I. In *Reliability yields information gain*, Proceedings of the 14th Annual Machine Learning Conference of Belgium and the Netherlands: BENELEARN, 2005; Citeseer: 2005; pp 89-96.
125. Han, J.; Kamber, M.; Pei, J., *Data mining: concepts and techniques: concepts and techniques*. Elsevier: 2011.
126. Vanderlooy, S.; Hüllermeier, E., A critical analysis of variants of the AUC. *Machine Learning* **2008**, 72, (3), 247-262.
127. Tsiporkova, E.; Tourwé, T.; Boeva, V. In *A collaborative decision support platform for product release definition*, 2010 Fifth International Conference on Internet and Web Applications and Services, 2010; IEEE: 2010; pp 351-356.
128. Bouyssou, D., Building criteria: a prerequisite for MCDA. In *Readings in multiple criteria decision aid*, Springer: 1990; pp 58-80.
129. Gallagher, M.; Hares, T.; Spencer, J.; Bradshaw, C.; Webb, I., The nominal group technique: a research tool for general practice? *Family Practice* **1993**, 10, (1), 76-81.
130. Majumder, M., Multi Criteria Decision Making. In *Impact of Urbanization on Water Shortage in Face of Climatic Aberrations*, Springer: 2015; pp 35-47.
131. Saaty, T. L., How to make a decision: the analytic hierarchy process. *European journal of operational research* **1990**, 48, (1), 9-26.
132. Saaty, T. L., The Analytical Hierarchy Process: Planning, Setting Priorities, Resource Allocation. In McGraw-Hill International Book Co., New York: 1980.
133. Garcia-Cascales, M. S.; Lamata, M. T., On rank reversal and TOPSIS method. *Mathematical and Computer Modelling* **2012**, 56, (5), 123-132.
134. Van Rijn, J. N.; Bischl, B.; Torgo, L.; Gao, B.; Umaashankar, V.; Fischer, S.; Winter, P.; Wiswedel, B.; Berthold, M. R.; Vanschoren, J., OpenML: A collaborative science platform. In *Machine learning and knowledge discovery in databases*, Hendrik, B., Ed. Springer: 2013; pp 645-649.



135. Soares, C.; Brazdil, P.; Costa, J., Measures to evaluate rankings of classification algorithms. In *Data Analysis, Classification, and Related Methods*, Springer: 2000; pp 119-124.
136. Brazdil, P. B.; Soares, C., A comparison of ranking methods for classification algorithm selection. In *Machine Learning: ECML 2000*, Springer: 2000; pp 63-75.
137. Neave, H. R.; Worthington, P. L., *Distribution-free tests*. Unwin Hyman London: 1988.
138. Neave, H.; Worthington, P., *Distribution-Free Tests*. London: Routledge: 1992.
139. Zar, J. H., Significance testing of the Spearman rank correlation coefficient. *Journal of the American Statistical Association* **1972**, 67, (339), 578-580.
140. Saaty, T. L., Rank from comparisons and from ratings in the analytic hierarchy/network processes. *European journal of operational research* **2006**, 168, (2), 557-570.
141. Opricovic, S.; Tzeng, G.-H., Compromise solution by MCDM methods: A comparative analysis of VIKOR and TOPSIS. *European journal of operational research* **2004**, 156, (2), 445-455.
142. Leyva Lopez; Carlos, J., Multicriteria decision aid application to a student selection problem. *Pesquisa Operacional* **2005**, 25, (1), 45-68.
143. Goicoechea, A.; Hansen, D. R.; Duckstein, L., Multiobjective decision analysis with engineering and business applications. *Wiley Online Library* **1982**.
144. Insua, D. R.; French, S., A framework for sensitivity analysis in discrete multi-objective decision-making. *European journal of operational research* **1991**, 54, (2), 176-190.
145. Friedman, M., A comparison of alternative tests of significance for the problem of m rankings. *The Annals of Mathematical Statistics* **1940**, 11, (1), 86-92.
146. Rice, J. R., The algorithm selection problem. *Advances in computers* **1976**, 15, 65-118.
147. Sun, Q., Integrated Fantail library. In 2014 ed.; GitHub: 2014.
148. Sarkheyli, A.; Sa'ffker, D., Case indexing in Case-Based Reasoning by applying Situation Operator Model as knowledge representation model. *IFAC-PapersOnLine* **2015**, 48, (1), 81-86.
149. Rokach, L.; Maimon, O., Decision trees. In *Data mining and knowledge discovery handbook*, Springer: 2005; pp 165-192.
150. Freitas, A. A., Comprehensible classification models: a position paper. *ACM SIGKDD explorations newsletter* **2014**, 15, (1), 1-10.
151. Rokach, L.; Maimon, O., *Data mining with decision trees: theory and applications*. World scientific: 2014.
152. Bello-Tomájs, J. J.; GonzÁlez-Calero, P. A.; DÁaz-Agudo, B. n., Jcolibri: An object-oriented framework for building cbr systems. In *Advances in case-based reasoning*, Springer: 2004; pp 32-46.

153. Rahman, A. M., Sadiq, Automatic-algorithm-selector. In April, 2016 ed.; GitHub: 2016.
154. Wang, W.; Krishnan, E., Big data and clinicians: a review on the state of the science. *JMIR medical informatics* **2014**, 2, (1).
155. Vesin, A.; Azoulay, E.; Ruckly, S.; Vignoud, L.; RusinovÃ , K.; Benoit, D.; Soares, M.; Azevedo-Maia, P.; Abroug, F.; Benbenishty, J., Reporting and handling missing values in clinical studies in intensive care units. *Intensive care medicine* **2013**, 39, (8), 1396-1404.
156. Singh, K.; Thakur, S. S.; Lal, M., Vague rough set techniques for uncertainty processing in relational database model. *Informatica* **2008**, 19, (1), 113-134.
157. Abu-Donia, H. M., Multi knowledge based rough approximations and applications. *Knowledge-Based Systems* **2012**, 26, 20-29.
158. Zierler-Brown, S.; Brown, T. R.; Chen, D.; Blackburn, R. W., Clinical documentation for patient care: models, concepts, and liability considerations for pharmacists. *American Journal of Health-System Pharmacy* **2007**, 64, (17), 1851-1858.
159. Yao, Y., Semantics of fuzzy sets in rough set theory. In *Transactions on Rough Sets II*, Springer: 2005; pp 297-318.
160. Grzymala-Busse, J. W.; Hu, M. In *A comparison of several approaches to missing attribute values in data mining*, Rough sets and current trends in computing, 2001; Springer: 2001; pp 378-385.
161. Markus, H.; Ralf, K., *RapidMiner: Data Mining Use Cases and Business Analytics Applications*. Chapman & Hall/CRC: 2013; p 525.
162. Chen, Y. S.; Chang, J. F.; Cheng, C. H., Forecasting IPO returns using feature selection and entropy-based rough sets. *International Journal of Innovative Computing, Information and Control* **2008**, 4, (8), 1861-1875.
163. Ali, R.; Siddiqi, M.; Lee, S., Rough set-based approaches for discretization: a compact review. *Artificial Intelligence Review* **2015**, 1-29.
164. World Health, O., BMI classification 2013. *World Health Organization*. URL: <http://apps.who.int/bmi/index.jsp>.
165. Pickering, T. G.; Hall, J. E.; Appel, L. J.; Falkner, B. E.; Graves, J.; Hill, M. N.; Jones, D. W.; Kurtz, T.; Sheps, S. G.; Roccella, E. J., Recommendations for blood pressure measurement in humans and experimental animals part 1: blood pressure measurement in humans: a statement for professionals from the Subcommittee of Professional and Public Education of the American Heart Association Council on High Blood Pressure Research. *Hypertension* **2005**, 45, (1), 142-161.
166. National High Blood Pressure Education, P., The seventh report of the Joint National Committee on prevention, detection, evaluation, and treatment of high blood pressure. **2004**.
167. Chobanian, A. V.; Bakris, G. L.; Black, H. R.; Cushman, W. C.; Green, L. A.; Izzo Jr, J. L.; Jones, D. W.; Materson, B. J.; Oparil, S.; Wright Jr, J. T., The seventh report of the joint national committee on prevention, detection, evaluation,

- and treatment of high blood pressure: the JNC 7 report. *Jama* **2003**, 289, (19), 2560-2571.
168. American Diabetes, Association: Diagnosis and classification of diabetes mellitus. *Diabetes care* **2011**, 33, (Supplement 1), S62-S69.
  169. American Diabetes, Association: Standards of medical care in diabetes--2008. *Diabetes care* **2008**, 31, S12.
  170. Type 1 diabetes: Diagnosis and management of type 1 diabetes in children, young people and adults NICE guidelines, CG15. **2004**.
  171. NICE guidelines [CG87], Type 2 diabetes: The management of type 2 diabetes **2009**.
  172. Expert Panel on Detection, E., Executive summary of the third report of the National Cholesterol Education Program (NCEP) expert panel on Detection, Evaluation, and Treatment of high blood cholesterol in adults (Adult Treatment Panel III). *Jama* **2001**, 285, (19), 2486.
  173. Henry N Ginsberg, M., Are the new ADA guidelines for Lipids optimal for individuals with diabetes mellitus. *Columbia University College of Physicians and Surgeons*.
  174. Vashist, R.; Garg, M. L., Rule generation based on reduct and core: A rough set approach. *International Journal of Computer Application* **2011**, 29, (9), 1-5.
  175. Predki, B.; Wilk, S., Rough set based data exploration using ROSE system. In *Foundations of Intelligent Systems*, Springer: 1999; pp 172-180.
  176. Predki, B. o.; SÅ,owiÅ„ski, R.; Stefanowski, J.; Susmaga, R.; Wilk, S. In *ROSE-software implementation of the rough set theory*, Rough Sets and Current Trends in Computing, 1998; Springer: 1998; pp 605-608.
  177. Stefanowski, J., On rough set based approaches to induction of decision rules. *Rough sets in knowledge discovery* **1998**, 1, (1), 500-529.
  178. McDonald, J. H., *Handbook of biological statistics*. Sparky House Publishing Baltimore, MD: 2009; Vol. 2.
  179. Pu, P.; Chen, L.; Hu, R., Evaluating recommender systems from the userâ€™s perspective: survey of the state of the art. *User Modeling and User-Adapted Interaction* **2012**, 22, (4-5), 317-355.
  180. Brodersen, K. H.; Ong, C. S.; Stephan, K. E.; Buhmann, J. M. In *The balanced accuracy and its posterior distribution*, Pattern Recognition (ICPR), 2010 20th International Conference on, 2010; IEEE: 2010; pp 3121-3124.
  181. Velez, D. R.; White, B. C.; Motsinger, A. A.; Bush, W. S.; Ritchie, M. D.; Williams, S. M.; Moore, J. H., A balanced accuracy function for epistasis modeling in imbalanced datasets using multifactor dimensionality reduction. *Genetic epidemiology* **2007**, 31, (4), 306-315.
  182. Banos, O.; Bilal Amin, M.; Ali Khan, W.; Afzal, M.; Ahmad, M.; Ali, M.; Ali, T.; Ali, R.; Bilal, M.; Han, m.; Hussain, J.; Hussain, M.; Hussain, S.; Hur, T. H.; Bang, J. H.; Huynh-The, T.; Idris, M.; Kang, D. W.; Park, S. B.; Siddiqui, M.; Vui, L. B.; Fahim, M.; Khattak, A. M.; Kang, B. H.; Lee, S. In *An Innovative*

- Platform for Person-Centric Health and Wellness Support*, Bioinformatics and Biomedical Engineering, Granada, Spain, 2015; Ortuño, F.; Rojas, I., Eds. Springer International Publishing: Granada, Spain, 2015; pp 131-140.
183. Banos, O.; Bilal Amin, M.; Ali Khan, W.; Afzal, M.; Ali, T.; Kang, B. H.; Lee, S. In *The Mining Minds Platform: a Novel Person-Centered Digital Health and Wellness Framework*, 9th International Conference on Pervasive Computing Technologies for Healthcare, Istanbul, Turkey, 2015; Istanbul, Turkey, 2015.
  184. Ainsworth, B. E.; Haskell, W. L.; Herrmann, S. D.; Meckes, N.; Bassett Jr, D. R.; Tudor-Locke, C.; Greer, J. L.; Vezina, J.; Whitt-Glover, M. C.; Leon, A. S., 2011 Compendium of Physical Activities: a second update of codes and MET values. *Medicine and science in sports and exercise* **2011**, 43, (8), 1575-1581.
  185. Nassim, D.; Marie-Christine, J., Clinical Practice Guidelines Formalization for Personalized Medicine. *Int. J. Appl. Evol. Comput.* **2013**, 4, (3), 26-33.
  186. World Health, O., BMI classification *World Health Organization*. URL: <http://apps.who.int/bmi/index.jsp> **2013**.
  187. Robinson, J. D.; Lupkiewicz, S. M.; Palenik, L.; Lopez, L. M.; Ariet, M., Determination of ideal body weight for drug dosage calculations. *American Journal of Health-System Pharmacy* **1983**, 40, (6), 1016-1019.
  188. Singhal, S.; Gordon, L. I.; Tallman, M. S.; Winter, J. N.; Evens, A. O.; Frankfurt, O.; Williams, S. F.; Grinblatt, D.; Kaminer, L.; Meagher, R., Ideal rather than actual body weight should be used to calculate cell dose in allogeneic hematopoietic stem cell transplantation. *Bone marrow transplantation* **2006**, 37, (6), 553-557.
  189. Jette, M.; Sidney, K.; BlÅ¼mchen, G., Metabolic equivalents (METs) in exercise testing, exercise prescription, and evaluation of functional capacity. *Clinical cardiology* **1990**, 13, (8), 555-565.
  190. World Health; Organization *Global recommendations on physical activity for health*; 2010.
  191. UK; Guidelines *Technical Report: Physical Activity Guidelines in the UK: Review and Recommendations*; 2010.
  192. Haskell, W. L.; Lee, I. M.; Pate, R. R.; Powell, K. E.; Blair, S. N.; Franklin, B. A.; Macera, C. A.; Heath, G. W.; Thompson, P. D.; Bauman, A., Physical activity and public health: updated recommendation for adults from the American College of Sports Medicine and the American Heart Association. *Circulation* **2007**, 116, (9), 1081.
  193. US; Guidelines, Physical activity guidelines advisory committee report, 2008. *Washington, DC: US Department of Health and Human Services* **2008**, 2008, A1-H14.
  194. Baget, J.-F. In *Improving the Forward Chaining Algorithm for Conceptual Graphs Rules*, International Conference on Principles of Knowledge Representation and Reasoning, 2004; 2004; pp 407-414.
  195. Cover, T.; Hart, P., Nearest neighbor pattern classification. *Information Theory, IEEE Transactions on* **1967**, 13, (1), 21-27.

196. Neyman, J., Outline of a theory of statistical estimation based on the classical theory of probability. *Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences* **1937**, 236, (767), 333-380.
197. Altman, D. G.; Gore, S. M.; Gardner, M. J.; Pocock, S. J., Statistical guidelines for contributors to medical journals. *British medical journal (Clinical research ed.)* **1983**, 286, (6376), 1489-1493.
198. Fisher, R. A., *Statistical methods for research workers*. Genesis Publishing Pvt Ltd: 1925.
199. Han, M.; Lee, Y.-K.; Lee, S., Comprehensive context recognizer based on multimodal sensors in a smartphone. *Sensors* **2012**, 12, (9), 12588-12605.
200. Banos, O.; Bang, J.; Hur, T.; Siddiqi, M. H.; Hyunh-The, T.; Vui, L. B.; Khan, A.; W., A. T.; Villalonga, C.; Lee, S. In *Mining Human Behavior for Health Promotion*, Proceedings of the 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC 2015) Milano, Italy, August 25-29, 2015; Milano, Italy, 2015.
201. Microsoft *Microsoft Excel 2010 [computer software]*, Microsoft Corporation: 2010.
202. Pu, P.; Chen, L.; Hu, R., Evaluating recommender systems from the user's perspective: survey of the state of the art. *User Modeling and User-Adapted Interaction* **2012**, 22, (4-5), 317-355.
203. Asanchezrg; belend; jarecio; pagoncal, jCOLIBRI: CBR Framework. In 2004 ed.; sourceforge: 2013.

## Appendix

### List of Publications

---

#### International Journals Papers:

1. **Rahman Ali**, Muhammad Afzal, Maqbool Hussain, Maqbool Ali, Muhammad Hameed Siddiqi, Byong Ho Kang, Sungyoung Lee. "Multimodal hybrid reasoning methodology for personalized wellbeing services". Computers in biology and medicine (SCI, IF: 1.24). 2016 Feb 1;69:10-28.
2. **Rahman Ali**, Jamil Hussain, Muhammad Hameed Siddiqi, Maqbool Hussain, Sungyoung Lee, "H2RM: A Hybrid Rough Set Reasoning Model for Prediction and Management of Diabetes Mellitus," Journal of Sensors (SCIE, IF: 2.245), Vol. 15, Issue 7, pp. 15921–15951, 2015.
3. **Rahman Ali**, Muhammad Hameed Siddiqi, Muhammad Idris, Shujaat Hussain, Eui-Nam Huh, Taqdir Ali, Byong Ho Kang, Sungyoung Lee, "GUDM: Automatic Generation of Unified Datasets for Learning and Reasoning in Healthcare," Journal of Sensors (SCIE, IF: 2.245), Vol. 15, Issue 7, pp. 15772–15798, 2015.
4. **Rahman Ali**, Muhammad Hameed Siddiqi, and Sungyoung Lee, "Rough Set-based Approaches for Discretization: A Compact Review", Artificial Intelligence Review, (SCI, IF: 2.111), 2015 Aug 1;44(2):235-63.
5. Muhammad Hameed Siddiqi, **Rahman Ali**, Adil Mehmood Khan, Young-Tack Park, Sungyoung Lee, "Human Facial Expression Recognition using Stepwise Linear Discriminant Analysis and Hidden Conditional Random Fields," IEEE Transaction on Image Processing (SCI, IF: 3.625), 2015.
6. Maqbool Hussain, Muhammad Afzal, Taqdir Ali, **Rahman Ali**, Wajahat Ali Khan, Arif Jamshed, Sungyoung Lee, Byeong Ho Kang and Khalid Latif,

- "Data-driven knowledge acquisition, validation, and transformation into HL7 Arden Syntax", *Artificial Intelligence in Medicine* (SCI, IF:2.109), 2015
7. Ishtiaq Ahmed, **Rahman Ali**, Donghai Guan, Sungyoung Lee, Yongkoo Lee and TaeChoong Chung, "Semi-supervised learning using frequent itemset and ensemble learning for SMS classification", *Expert Systems with Applications* (SCIE, IF:1.965), Volume 42, Issue 3, pp.1065-1073, 2015
  8. Muhammad Hameed Siddiqi, **Rahman Ali**, Adil Mehmood Khan, Eun Soo Kim, Min Cheol Whang, Gerard Junghyun Kim, Sungyoung Lee, "Facial Expression Recognition using Active Contour-based Face Detection, Facial Movement-based Feature Extraction, and Non-Linear Feature Selection," *Multimedia Systems* (SCI, IF: 0.619), DOI: 10.1007/s00530-014-0400-2, 2014.
  9. Muhammad Hameed Siddiqi, **Rahman Ali**, Muhammad Idris, Adil Mehmood Khan, Eun Soo Kim, Min Cheol Whang, Sungyoung Lee, "Facial Expression Recognition using Curvelet Feature Extraction and Normalized Mutual Information Feature Selection," *Multimedia Tools and Applications* (SCIE, IF:1.346), DOI: 10.1007/s11042-014-2333-3, 2014.
  10. Muhammad Hameed Siddiqi, **Rahman Ali**, Abdul Sattar, Adil Mehmood Khan, Sungyoung Lee, "Depth Camera-based Facial Expression Recognition System using Multilayer Scheme," *IETE Technical Review* (SCIE, IF: 0.888), volume. 31, issue no. 4, pp. 277 – 286, 2014.
  11. Muhammad Hameed Siddiqi, **Rahman Ali**, Md. Sohel Rana, Een-Kee Hong, Eun Soo Kim, Sungyoung Lee, "Video-based Human Activity Recognition using Multilevel Wavelet Decomposition and Stepwise Linear Discriminant Analysis," *Journal of Sensors* (SCIE, IF: 2.245), volume. 14, issue no. 4, pp. 6370 – 6392, 2014.
  12. Muhammad Fahim, Muhammad Idris, **Rahman Ali**, Christopher Nugent, Byeong Kang and Sungyoung Lee, "ATHENA: A Personalized Platform to Promote Active Lifestyle and Wellbeing based on Physical, Mental and Social Health Primitives", *Sensors* (SCIE, IF:1.953), Vol. 14, No 5, pp.9313–9329, 2014

**International Conferences Papers:**

1. **Rahman Ali**, Muhammad Hameed Siddiqi, Byeong Ho Kang and Sungyoung Lee, "KARE: A Hybrid Reasoning Approach for Promoting Active Lifestyle", ACM-IMCOM (ICUIMC) 2015, Jan 8-10, 2015.
2. **Rahman Ali**, Muhammad Hameed Siddiqi, Muhammad Idris, Byeong Ho Kang, and Sungyoung Lee, "The Prediction of Diabetes Mellitus Based on Boosting Ensemble Modeling", 8th International Conference on Ubiquitous Computing & Ambient Intelligence (UCAmI 2014), Belfast, United Kingdom, December 2-5, 2014.
3. Wajahat Ali Khan, Muhammad Bilal Amin, Oresti Banos, Taqdir Ali, Maqbool Hussain, Muhammad Afzal, Shujaat Hussain, Jamil Hussain, **Rahman Ali**, Maqbool Ali, Dongwook Kang, Jaehun Bang, Tae Ho Hur, Bilal Ali, Muhammad Idris, Asif Razzaq, Sungyoung Lee and Byeong Ho Kang, "Mining Minds: Journey of Evolutionary Platform for Ubiquitous Wellness", 12th International Conference on Ubiquitous Healthcare (u-Healthcare 2015), Osaka, Japan, Nov 30- Dec 02, 2015.
4. Wajahat Ali Khan, Muhammad Idris, Taqdir Ali, **Rahman Ali**, Shujaat Hussain, Maqbool Hussain, Muhammad Bilal Amin, Asad Masood Khattak, Yuan Weiwei, Muhammad Afzal, Sungyoung Lee and Byeong Ho Kang, "Correlating Health and Wellness Analytics for Personalized Decision Making", 17th International Conference on E-health Networking, Application & Services (Healthcom 2015), Oct 14-17, 2015.
5. Banos, O., Bilal Amin, M., Ali Khan, W., Afzel, M., Ahmad, M., Ali, M., Ali, T., **Ali, R.**, Bilal, M., Han, M., Hussain, J., Hussain, M., Hussain, S., Hur, T. H., Bang, J. H., HuynhThe, T., Idris, M., Kang, D. W., Park, S. B., Siddiqi, M., Vui, L. B., Fahim, M., Khattak, A. M., Kang, B. H. and Lee, S, "An Innovative Platform for Person-Centric Health and Wellness Support", Proceedings of the International Work-Conference on Bioinformatics and Biomedical Engineering (IWBBIO 2015), pp. 131–140, Granada, Spain, April 15–17, 2015.



6. Muhammad Hamed Siddiqi, **Rahman Ali**, Byeong Ho Kang, Sungyoung Lee, “A New Feature Extraction Technique for Human Facial Expression Recognition Systems using Depth Camera”. Proc. Of the 6<sup>th</sup> International Work-conference on Ambient Assisted Living (IWAAL’14), pp. 131 – 138, UK, 2014.
7. Muhammad Hameed Siddiqi, **Rahman Ali**, Ibrahiem M. M. El Emary, Sungyoung Lee, “An Unsupervised and Robust Technique for Human Face Detection and Extraction”, Proc. of IEEE International Conference on Information Science, Electronics and Electrical Engineering (ISEEE’14), pp. 1756 – 1760, Sapporo City, Hokkaido, Japan, 2014.