

DEVELOPMENT OF FEATURE SELECTION AND LEARNING MECHANISM FOR FUZZY RULE-BASED SYSTEM

Thesis Submitted for the Award of the Degree of

DOCTOR OF PHILOSOPHY

in

Computer Science and Engineering

By

Ankur Kumar

Registration Number: 41800879

Supervised By

Dr. Avinash Kaur (14557)

Department of Computer Science and Engineering (Professor)

Lovely Professional University



L OVELY
P ROFESSIONAL
U NIVERSITY

Transforming Education Transforming India

LOVELY PROFESSIONAL UNIVERSITY, PUNJAB

2025

DECLARATION

I, hereby declared that the presented work in the thesis entitled “Development of Feature Selection and Learning Mechanism for Fuzzy Rule Based System” in fulfilment of degree of **Doctor of Philosophy (Ph. D.)** is outcome of research work carried out by me under the supervision of Dr. Avinash Kaur, working as Professor, in the School of Computer Science and Engineering of Lovely Professional University, Punjab, India. In keeping with general practice of reporting scientific observations, due acknowledgements have been made whenever work described here has been based on findings of other investigator. This work has not been submitted in part or full to any other University or Institute for the award of any degree.

Ankur Kumar

41800879

School of Computer Science and Engineering

Lovely Professional University,

Punjab, India

CERTIFICATE

This is to certify that the work reported in the Ph. D. thesis entitled “Development of Feature Selection and Learning Mechanism for Fuzzy Rule Based System” submitted in fulfillment of the requirement for the award of degree of **Doctor of Philosophy (Ph.D.)** in the School of Computer Science and Engineering, is a research work carried out by Ankur Kumar, 41800879, is bonafide record of his original work carried out under my supervision and that no part of thesis has been submitted for any other degree, diploma or equivalent course.

Dr. Avinash Kaur
Professor
School of Computer Science and Engineering
Lovely Professional University,
Punjab, India

ABSTRACT

Uncertainty and ambiguity are inherent in real-world problems, which can be easily handled by fuzzy methods – which are considered to be the best practice for dealing with uncertainties. Fuzzy Rule-Based Systems (FRBS) are widely recognized for their ability to handle uncertainty and model complex systems using human-understandable rules. In recent years, fuzzy set theory and fuzzy logic have been successfully applied in real-life application areas including health practices, especially diagnosis, bioinformatics, speech recognition, medicine, diseases, the Internet, pattern recognition, control systems, decision support systems, and many more.

While considering real-world data, most of them are high dimensional. FRBS faces several challenges while dealing with high-dimensional data. Key challenges include the curse of dimensionality, which makes fuzzy systems design huge and complex, due to noise in data. This leads to poor generalization, overfitting, and increased computational complexity, including parameters such as accuracy, training time, memory usage, model complexity, and inference time. Other than the above challenges, high dimensional data have issues of feature redundancy and data sparsity. Moreover, fuzzy systems are valued for their interpretability, high dimensional datasets make the model less interpretable. Thus, a compromise was made between interpretability and accuracy.

This thesis uses feature selection techniques to improve state-of-the-art methods by reducing dimensionality and removing unnecessary features. Furthermore, membership functions are learned and optimized using Genetic Algorithm (GA) and Particle Swarm Optimization (PSO). These methods effectively mitigate overfitting, computational costs, and feature redundancy. Furthermore, the balance between interpretability and accuracy is carefully managed to ensure that the fuzzy system remains both accurate

and interpretable. While fuzzy systems are valued for their interpretability, the increasing complexity of high-dimensional data often diminishes the model's transparency, leading to a trade-off between interpretability and accuracy. The proposed methods aim to resolve this trade-off while enhancing the overall performance of the fuzzy system.

A feature selection hybrid approach MISFS has been developed to perform the dimensionality reduction and eliminate irrelevant features. This MISFS is a new hybrid feature selection method that uses MI, the SFS Method, and 10-fold CV to choose the best features. It combines filter and wrapper techniques. The work considered is applied to the six datasets, considered from the area of medicine. The datasets considered are both binary and multi-class. According to the experiment's findings, the MISFS method can reduce features in large feature datasets by over 75%. Furthermore, the algorithm's accuracy on the datasets under consideration is either higher than or comparable to that of existing algorithms. In addition to the aforementioned outcomes, this algorithm also resolves the majority of issues raised in the literature that the earlier work did not take into account. These challenges include increasing accuracy, decreasing the number of features, lowering the computational complexity, lowering the computational cost, and reducing the storage space.

After applying this feature selection approach, the number of features received, a fuzzy rule-based system is designed. The proposed LT_FRBS method fine-tunes the membership functions, aiming to strike an optimal balance between accuracy and interpretability. The algorithm learns and tunes the MFs using optimizing methods Genetic Algorithm and Particle Swarm Optimization. Complexity-based interpretability and semantic-based interpretability are two types of interpretability as per the literature, which can be analyzed based on a rule-base label and fuzzy level partition. The study shows that there are numerous parameters to study the interpretability of FRBS, out of which most of the previous studies specifically focussed on two parameters viz. number of rules and number of membership functions. The proposed algorithm focuses on 10 parameters to study the interpretability and further minimize the trade-off between accuracy and interpretability. The parameters studied

in the research work for interpretability are the number of rules, conditions, membership functions, features, consistency of the rule base, the number of rules fired simultaneously, completeness, normalization, distinguishability, complementarity. The results also demonstrate that both GA and PSO can enhance the performance of FRBS by refining membership functions. This leads to notable improvements in accuracy while maintaining or even enhancing system interpretability.

The results of LT_FRBS are calculated in terms of accuracy and the interpretability of the datasets considered. Also, the results are compared with the previous studies in terms of both accuracy and interpretability. Also, focusing on how the trade-off is balanced between interpretability and accuracy.

Furthermore, the findings indicate that most of the challenges are effectively addressed. Specifically, feature reduction is more effective for high-dimensional data, while a moderate reduction rate is observed for low-dimensional data. This helps reduce storage space requirements and minimizes the ratio of selected features. Importantly, the trade-off between accuracy and interpretability is minimized, as evidenced by improved interpretability on key parameters like total rules, conditions per rule, and the consistency and comprehensiveness of the rule base. Comparing the results with previous studies, shows that, most of the earlier research either focus on accuracy more promptly and silent on interpretability or discuss only one or two parameters of interpretability. Thus, not justifying the trade-off issues. This issue is fully explored in this research work by considering multiple parameters of interpretability and justifying how trade-off is minimized between interpretability and accuracy.

The complexity (rules, variables) and semantics (completeness, distinguishability) of the fuzzy set partitions both affect how interpretable fuzzy logic systems (FLSs) are. Although optimization methods increase efficiency, they frequently change important parameters, which compromises interpretability. To improve performance without sacrificing interpretability, this study examines tuning an FLS's Membership Function (MF) while maintaining constant rules, antecedents, and consequents. The study contrasts conventional optimization techniques with GA and PSO. According to experimental findings, interpretability can be maintained by optimizing particular FLS

components Specifically, feature reduction is achieved more effectively for high-dimensional data, while a moderate reduction rate is observed for low-dimensional data.. To assess model performance, the application of interpretability indices is investigated.

Extending experiments with various rule generation techniques and datasets, as well as expanding optimization capabilities, will be the main goals of future research. Also, to explore newer or domain-specific optimization techniques, multi-objective optimization methods will be explored. Further, more research areas are to be explored to measure the performance of the proposed work.

ACKNOWLEDGEMENTS

I take this opportunity to express my sincere thanks to everyone who helped me in various capacities to carried out this research and prepare the report.

I am delighted to thank our respected supervisor Dr. Avinash Kaur who has offered tremendous support in the completion of this research. Their unparalleled knowledge, judgment, and moral fiber were together with their expertise.

I acknowledge the Department of Computer Science and Engineering, Lovely Professional University for providing me the appropriate resources and financial support to pursue the doctoral degree. I am grateful to the administration staff at the Centre for Research Degree Programmes for the numerous applications.

I would also thank my Parents, Gurus, Wife, Son, Friends, and Contemporaries for their cooperation and compliance. I cannot cherish a greater fortune than having them in life. Their care and love are indispensable for my achievements.

Ankur Kumar

CONTENTS

Declaration	ii
Certificate	iii
Abstract	iv
Acknowledgement	viii
List of Tables	xiv
List of Figures	xvi
List of Abbreviations	xviii
1. Introduction	1
1.1. Introduction to Fuzzy Logic and Fuzzy Rule Based System	1
1.1.1. Types of FRBS	2
1.1.2. Features of Fuzzy Modelling	4
1.1.3. Areas of Fuzzy Modelling	4
1.1.4. System Modelling	6
1.1.5. Tasks performed by Fuzzy Logic	7
1.2. Feature Selection	8
1.2.1. Feature Selection Approaches	8
1.3. Challenges in FRBS	9
1.3.1. Challenges in Dimensionality Reduction	9
1.3.2. Challenges related to Accuracy and Interpretability	9

1.4.	Research Issues	10
1.5.	Research Objectives	11
1.6.	Research Methodology	12
1.7.	Thesis Contribution	12
1.8.	Thesis Organization	14
2.	Literature Review	16
2.1.	Introduction	16
2.2.	Feature Selection	17
2.3.	Feature Selection Approaches	17
2.4.	Benefits and Drawbacks of the Feature Selection Approach	18
2.5.	Unsupervised Feature Selection	18
2.6.	Semi-supervised feature selection methods	21
2.6.1.	Graph- based methods	21
2.6.2.	SVM based methods	22
2.6.3.	Self-training based	22
2.6.4.	Co-training based	23
2.6.5.	Filter Methods and Graph-Based Methods	24
2.6.6.	Wrapper methods and Self Training Based or Graph Based or Co-Training Based	24
2.6.7.	SVM Based	24
2.7.	Findings from the literature	26
2.8.	Critical insights from unsupervised and semi-supervised feature Selection methods	33
2.9.	Summary	36
3.	MISFS Algorithm for Feature Selection	38
3.1.	Introduction	38
3.2.	Background	40
3.2.1.	Dimensionality Reduction	40
3.2.2.	Model for feature selection	41
3.3.	Related Work	42

3.4.	Research Methodology	46
3.4.1.	MISFS Approach	46
3.4.2.	Abstract View of MISFS and Comparison methods	47
3.4.3.	Phases in MISFS	47
3.4.4.	Method and Algorithm	48
3.4.5.	Workflow of the MISFS	51
3.5.	Experimental Evaluation and Comparative Analysis	52
3.5.1.	Experimental Setup	52
3.5.1.1.	Datasets Used	53
3.5.1.2.	Performance Metrics	55
3.5.1.3.	Comparison Techniques	56
3.5.2.	Results and Discussion	56
3.5.2.1.	Features Selected by MISFS	56
3.5.2.2.	Performance on different datasets	58
3.5.2.3.	Comparative Study of MISFS with LR, KNN and PCC	59
3.5.2.4.	Comparison of the algorithm MISFS with existing work	62
3.5.2.5.	Discussion and Insights	64
3.6.	Summary	66
4.	Learning and optimization techniques for FRBS	68
4.1.	Introduction	68
4.2.	Related Work	69
4.3.	Learning of FRBS	71
4.4.	Learning Algorithms Fuzzy Rule-Based Systems	72
4.4.1.	Adhoc Covering Methods	73
4.4.2.	Neural Network Based Learning Methods	74
4.4.3.	Genetic Learning Fuzzy Rule Based Systems	75
4.5.	Tuning Fuzzy Rule-Based Systems	77
4.5.1.	Rule-Base optimization methods	77
4.5.2.	Membership function optimization	78
4.5.3.	Adaptive Rule-Based Systems	78
4.5.4.	Interpretability-Accuracy Trade-off	78

4.6.	Experimental Evaluation and comparative analysis of FRBS	
	Learning and Tuning methods	78
4.6.1.	Advantages and disadvantages of FRBS learning and tuning methods	79
4.6.2.	Comparative performance of learning and tuning methods Across different metrics	79
4.6.3.	Comparative study of FRBS Learning Methods and Tuning Methods	80
4.7.	Summary	81
5.	Optimizing FRBS using GA and PSO	83
5.1.	Introduction	83
5.2.	Trade-off between Interpretability and Accuracy	84
5.3.	Background	85
5.3.1.	Interpretability in Fuzzy Systems and Machine Learning	85
5.3.2.	Types of Interpretabilities in FRBS	85
5.3.3.	Interpretability Index	89
5.3.4.	Relationship between Accuracy and Interpretability in FRBS	89
5.3.5.	Learning and Tuning in FRBS	89
5.3.6.	Genetic Algorithm	92
5.3.7.	Particle Swarm Optimization	94
5.4.	Related Work	95
5.5.	Methodology	98
5.5.1.	Overview of the LT_FRBS Approach	98
5.5.2.	Abstract view of LT_FRBS	99
5.5.3.	Method and Algorithm of LT_FRBS	99
	5.5.3.1.Tuning of FRBS through GA	103
	5.5.3.2.Tuning of FRBS through PSO	105
5.6.	Experimental Evaluation and Comparative Analysis	105
5.6.1.	Experimental Setup	105
	5.6.1.1. Datasets	106
	5.6.1.2. Performance Metrics	106

5.6.1.3. Comparison Techniques	106
5.6.2. Results and Discussions	107
5.6.2.1. Accuracy measurements on Different Datasets	107
5.6.2.2. Interpretability measurement on different datasets	109
5.6.2.3. Comparative analysis with existing algorithm	114
5.6.2.4. Trade-off between Interpretability and Accuracy is minimized	117
5.6.2.5. Convergence and Local optima considerations	118
5.7. Summary	118
6. Conclusion and Future Directions	120
6.1. Conclusion and Discussion	120
6.2. Future Directions	121
References	123
List of Publications (Journals)	145
List of Conferences	146

LIST OF TABLES

Table 2.1:	Positives and negatives of the feature selection approach	19
Table 2.2:	Semi-supervised method classification	25
Table 2.3:	Literature Survey	26
Table 2.4:	Comparative analysis of UFS methods with their category	35
Table 2.5:	Advantages and disadvantages of semi-supervised feature selection methods based on learning methods	36
Table 3.1:	Overview of Data Samples	54
Table 3.2:	Optimized Features Selected	57
Table 3.3:	Performance of Algorithm MISFS	58
Table 3.4:	Comparison of performance of proposed algorithm MISFS with LR, KNN, and PCC	59
Table 3.5:	Comparing MISFS with existing work on Pima Indian diabetes	62
Table 3.6:	Comparing MISFS with existing work on Spectfheart	62
Table 3.7:	Comparing MISFS with existing work on Breast Cancer dataset	62
Table 3.8:	Comparing MISFS with previous work on Parkinson Dataset	63
Table 3.9:	Comparing MISFS with existing work on PCOS dataset	63
Table 3.10:	Comparing MISFS with existing work on Cleaveland Dataset	63
Table 3.11:	Summary of MISFS accuracy compared with PCC and datasets considering all features	64
Table 4.1:	Benefits and drawbacks of different learning and tuning methods	79

Table 4.2:	Suitability and Performance metrics comparison of three approaches	80
Table 4.3:	Learning Methods and Tuning Methods Comparison	81
Table 5.1:	Key differences between interpretability in FRBS and ML	86
Table 5.2:	Interpretability parameters definition	87
Table 5.3:	Learning and Tuning Techniques in FRBS	91
Table 5.4:	Existing work on different metrics of FRBS	95
Table 5.5:	Summary of the current work on the interpretability of FRBSs	97
Table 5.6:	Parameters for GA	104
Table 5.7:	Accuracy of the LT_FRBS Algorithm	108
Table 5.8:	Interpretability measures of PIMA dataset	109
Table 5.9:	Interpretability measures of Spectfheart dataset	110
Table 5.10:	Interpretability measures of Breast Cancer dataset	111
Table 5.11:	Interpretability measures of Parkinson dataset	111
Table 5.12:	Interpretability measures of PCOS dataset	112
Table 5.13:	Interpretability measures of Cleaveland dataset	112
Table 5.14:	Interpretability results of all datasets through the proposed LT_FRBS algorithm	113
Table 5.15:	Comparison of LT_FRBS with existing work on PIMA Indian Diabetes	115
Table 5.16:	Comparison of LT_FRBS with existing work on Spectfheart	115
Table 5.17:	Comparison of LT_FRBS with Breast Cancer	115
Table 5.18:	Comparison of LT_FRBS with existing work on Parkinson	116
Table 5.19:	Comparison of LT_FRBS with existing work on PCOS	116
Table 5.20:	Comparison of LT_FRBS with existing work on Cleaveland	116

LIST OF FIGURES

Figure 1.1:	Basic structure of the Mamdani fuzzy rule-based system	2
Figure 1.2:	Linguistic and Precise Fuzzy Modelling	6
Figure 1.3:	Various Fuzzy Logic Tasks	7
Figure 1.4:	Research methodology flowchart	13
Figure 1.5:	Chapter-wise thesis organization	15
Figure 2.1:	Basic taxonomy-based semi-supervised methods of feature selection	22
Figure 2.2:	Learning method-based semi-supervised methods of feature selection	22
Figure 3.1:	Dimensionality Reduction	40
Figure 3.2:	A general framework of the feature selection process	41
Figure 3.3:	Abstract view of MISFS working and result comparison	47
Figure 3.4:	The flow of the proposed methodology	52
Figure 3.5:	K-fold cross-validation in Sequential Forward Feature Selection	53
Figure 3.6:	Feature selected by the proposed algorithm MISFS on each dataset	57
Figure 3.7:	Accuracy of each dataset on the algorithm MISFS with classifier LR and KNN	59
Figure 3.8:	Comparison of MISFS accuracy with accuracy achieved with all features and PCC on datasets considered	61
Figure 4.1:	Learning and Tuning of Fuzzy rule-based methods	71
Figure 4.2:	Design of FRBSs	73

Figure 4.3:	Graphical representation of a uniform fuzzy partition	74
Figure 4.4:	Deriving Mamdani rules from TSK rules	74
Figure 4.5:	Structure of ANFIS	75
Figure 4.6:	Learning through SLAVE	77
Figure 5.1:	A taxonomy to analyze the interpretability of FRBS	87
Figure 5.2:	Flowchart of the Genetic Algorithm	92
Figure 5.3:	Flow of the proposed algorithm LT_FRBS	100
Figure 5.4:	Triangular membership function	100
Figure 5.5:	Triangular membership function for each feature weight w	101
Figure 5.6:	Membership value for any feature with weight w	101
Figure 5.7:	Flowchart for tuning of MF using PSO	107
Figure 5.8:	Mean Accuracy of each dataset on the proposed algorithm after tuning with GA and PSO	108

LIST OF ABBREVIATIONS

ACC	Precision Metrics
FRBS	Fuzzy Rule-Based System
FS	Feature Selection
GA	Genetic Algorithm
KB	Knowledge Base
KNN	k-Nearest Neighbour
LFM	Linguistic Fuzzy Modelling
LR	Logistic Regression
MF	Membership Function
MI	Mutual Information
MSE	Mean Squared Error
NMI	Normalized Mutual Information
PCA	Principal Component Analysis
PFM	Precise Fuzzy Modelling
PSO	Particle Swarm Optimization
SFS	Sequential Forward Feature Selection
SVM	Support Vector Machine
UFS	Unsupervised Feature Selection

CHAPTER 1

INTRODUCTION

1.1. Introduction to Fuzzy Logic and Fuzzy Rule Based Systems

The process which is related to the social, economic and environmental decisions is called sustainable development. Parameters defined by sustainable decision making are complex and ill-defined having high degree of uncertainty. Fuzzy Logic is considered to be a standard tool for the appraisal of sustainability.

Fuzzy Logic is one of the tools of artificial intelligence and it has gained popularity in recent years. Fuzzy Logic deals with uncertainty i.e. it deals with the information that is imprecise and vague. The term “fuzzy” comes from this vagueness. It is a mathematical model of representing vague and imprecise information. Fuzzy models are capable of handling and processing data that is ambiguous or uncertain, enabling the recognition, representation, manipulation, and interpretation of vague information. Models based on Fuzzy Logic deals with sources of vagueness and uncertainty that are nonstatistical in nature. The foundation of Fuzzy Logic is fuzzy set theory proposed by Zadeh [1] in 1965. It is a current and innovative methodology that modifies conventional system design with engineering expertise. Anything designed using conventional techniques can also be designed using Fuzzy Logic [2].

The rule-based systems developed using fuzzy logic is called ‘Fuzzy Rule Based Systems (FRBS)’. These systems represent one of the most significant applications of fuzzy logic. Unlike classical rule-based systems, FRBS extend traditional logic by incorporating fuzzy rules instead of classical logic rules. They are successfully applied to a wide range of problems from different areas presenting uncertainty and vagueness in different ways.

There are two main components in FRBS. The first component is inference system and other is the knowledge base (KB). The KB stores the knowledge relevant to the problem at hand, which is represented by a set of fuzzy rules. Meanwhile, the inference system

carries out the fuzzy inference, using the input provided to generate an output from the FRBS.

1.1.1. Types of FRBS

FRBS is considered for improvement of accuracy and interpretability, but improving one provides sacrifice of the other. Following are the different types of FRBS:

- i. Mamdani System
- ii. Sugeno models (TSK method)
- iii. Tsukamoto models

Mamdani[3, 4], Takagi and Sugeno (TSK method)[5, 6], and Tsukamoto models[7] are three kinds of FRBS, that distinguish themselves based on how they express the consequent of the fuzzy rules within their KB. In the Mamdani system, the consequent involves a linguistic variable. In TSK fuzzy rules, the consequent is represented by polynomial functions of the inputs while in the Tsukamoto model, the consequent is depicted as a fuzzy set with a monotonic membership function. The final output in the Tsukamoto model is determined by calculating the weighted average of the outputs from each rule.

Mamdani first translates Zadeh's foundational work in the control system application as fuzzy rules. This system, known as Mamdani fuzzy rule-based system, is sometimes called fuzzy logic controllers, due to their first application as described in the landmark paper by Mamdani and Assilian [8, 9].

The elementary model of Mamdani FRBS consists of an inference system, KB, and the fuzzification and defuzzification interfaces depicted in Figure 1.1.

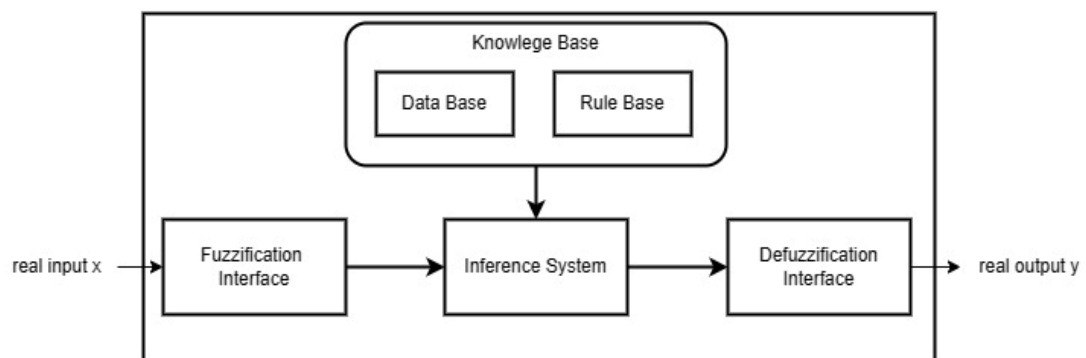


Figure 1.1. Basic structure of the Mamdani fuzzy rule-based system [10]

The Mamdani model serves the following key features:

- It is suitable for real-world applications because it can handle both real-valued inputs and outputs.
- It offers a natural framework for incorporating expert knowledge through linguistic rules and allows for easy integration of rules automatically generated from data sets that represents system behavior.
- It provides flexibility in selecting the fuzzification and defuzzification components, as well as the inference system, allowing for the design of a more tailored FRBS for specific problems.

The fuzzy rules in a Mamdani FRBS consist of linguistic variables for both input and output, which take values from a linguistic terms that holds real information. As a result, every rule represents a IF -THEN statement easily interpreted by humans. Thus making Mamdani FRBSs well-suited for linguistic modeling, a subfield of FM where interpretability is a key feature. An extension of Mamdani FRBSs involves altering the structure of the linguistic rules to increase flexibility. This modified FRBS uses disjunctive normal form (DNF) to design fuzzy rules as shown in Equation 1.1:

$$IF X_1 \text{ is } \tilde{A}_1 \text{ and } \dots X_n \text{ is } \tilde{A}_n \text{ THEN } Y \text{ is } B \quad (1.1)$$

where each variable X_i is associated with a referential set U_i and can take values within a finite domain $D_i, i = 1, 2, \dots, n$. Similarly, the referential set for Y is V and its domain is F . The value of the output variable Y is denoted as B , where $B \in F$, and the value of the variable X_i is represented as $\tilde{A}_i$, where $\tilde{A}_i \in P(D_i)$. Here, $P(D_i)$ represents the power set of D_i . Consequently, the complete syntax (Equation 1.2) for the antecedent of the rule is expressed as

$$X_1 \text{ is } \tilde{A}_1 = \{A_{1l} \text{ or } \dots \text{ or } A_{1ln}\} \text{ and } \dots \text{ and } X_n \text{ is } \tilde{A}_n = \{A_{nl} \text{ or } \dots \text{ or } A_{nln}\}. \quad (1.2)$$

An example of this kind of rule is shown in Equation 1.3. Considering three inputs, M_1 , M_2 , and M_3 and an output Y having the following linguistic term for each input

$$\begin{aligned} D_1 &= \{P_{11}, P_{12}, P_{13}\}, D_2 = \{Q_{21}, Q_{22}, Q_{23}, Q_{24}, Q_{25}\}, \\ D_3 &= \{R_{31}, R_{32}\}, F = \{B_1, B_2, B_3\}. \end{aligned} \quad (1.3)$$

The rule is formed using Disjunctive Normal Form (Equation 1.4) as

$$\begin{aligned} \text{If } M_1 \text{ is } \{P_{11} \text{ or } P_{13}\} \text{ and } M_2 \text{ is } \{Q_{23} \text{ or } Q_{25}\} \text{ and } M_3 \text{ is } \{Q_{31} \text{ and } Q_{32}\} \text{ then} \\ Y \text{ is } B_2 \end{aligned} \quad (1.4)$$

Fuzzy Modelling (FM) [11] is used to model a system using fuzzy logic concepts with fuzzy predicates. FM involves creating models in the form of FRBSs, which are constructed through parametric system identification methods. Fuzzy systems are mostly capable for control modelling, and classification, having numerous applications in this area. These modelling are highly successful due to their ability of decision making similar to human expert knowledge - providing information regarding many real systems, described by imprecise and vague statements. This makes them highly valuable in research, as they represent systems easily understandable to humans.

1.1.2. Features of Fuzzy Modelling

Fuzzy Modelling considered a system modelling discipline, is characterized by two important features that assess the quality of the obtained fuzzy models:

- i. *Interpretability*: It is a subjective property states that how much a fuzzy system is capable of expressing the behavior of a system understandably. The more the system is understandable more the system is interpretable. Several factors influence interpretability [12, 13], including the model's structure, total input variables, rules available, linguistic terms, and the structure of fuzzy systems. Different criteria for interpretability include compactness, completeness, consistency, or transparency.
- ii. *Accuracy*: Fuzzy systems have the ability to represent the modelled system closed to the real system. Closer the modelled system to real system, then the modelled system is called highly accurate. Closeness refers to similarity between the output of modelled system to the real system. Fuzzy model is sometimes called fuzzy approximation model due to the term approximation sometimes used for accuracy [12, 14].

1.1.3. Areas of Fuzzy Modelling

Zadeh stated the Principle of Incompatibility [15], "*as the complexity of a system increases, our ability to make precise and yet significant statements about its behaviour*

diminishes until a threshold is reached beyond which precision and significance (or relevance) become almost mutually exclusive characteristics.”

It means that obtaining high degrees of accuracy and interpretability at the same time is not possible, due to the inverse relationship among them. A highly interpretable system is less accurate and vice-versa. Fuzzy Modelling can be classified into two areas based on what requirement (accuracy or interpretability) is mainly needed:

- i. *Linguistic fuzzy modelling (LFM)*–This modelling prioritizes interpretability in fuzzy systems.
- ii. *Precise fuzzy modelling (PFM)*–This modelling prioritizes accuracy in fuzzy systems.

The main objectives in the field of FM are interpretability and accuracy. Ideally, both metrics should achieve a higher degree. But, due to their contradiction to each other, it is not possible to obtain both at a high degree. So, according to the nature of the problem, it is decided to give more priority to one objective, leaving the other one in the background. Thus, according to the problem, which objective is considered in priority, fuzzy modelling approaches can be of two types: Linguistic Fuzzy Modelling (interpretability) and Precise Fuzzy Modelling (accuracy).

Figure 1.2 shows the linguistic and precise fuzzy modelling with a trade-off. The general scheme available in the literature to perform fuzzy modelling is:

- i. Specific model is defined to handle the objectives interpretability and accuracy, called Fuzzy Modelling approach.
- ii. Different mechanism is used to improve the modelling process. Thus, obtaining the desired ratio of interpretability and accuracy.

Researchers have done a significant effort for compensating the initial difference by proposing improvement mechanism.

Classification is the process of determining the category or class to which a new observation belongs, based on a set of data with predefined categories. Classification problems are classified as binary classification and multiclass classification. Binary classification problems have two classes with yes or no choice. For example, a person may have cancer or not having cancer. Multiclass problems have k-classes. As an example, computer equipment's may be normal or abnormal. Abnormal may have

either fault 1, fault 2, or fault 3. Thus, computer equipment has four classes (normal, fault 1, fault 2 and fault 3). To solve multiclass problems, we need more decision functions.

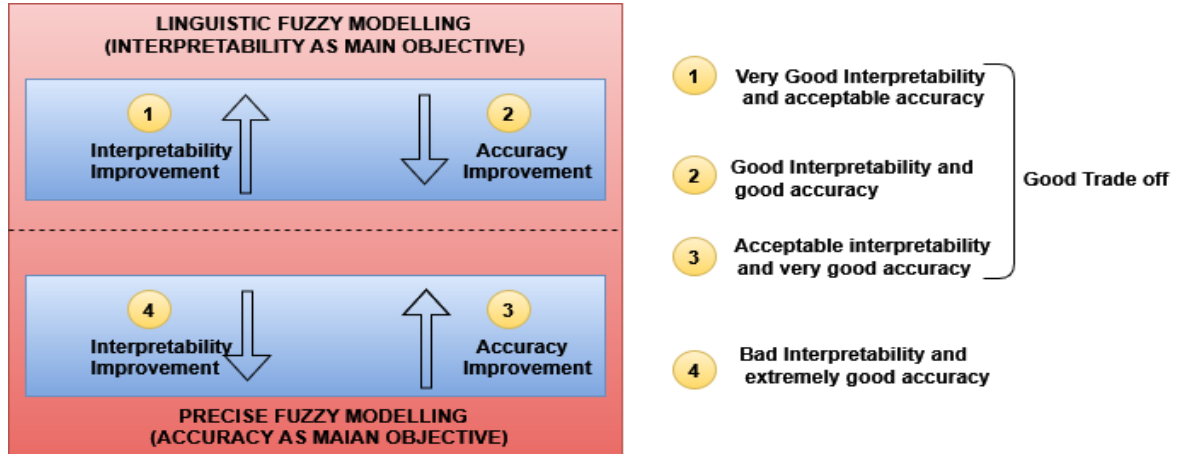


Figure 1.2: Linguistic and Precise Fuzzy Modelling[12]

When it comes to solving classification problems, FRBS are particularly useful due to their ability to handle uncertainty and imprecision in data. In classification tasks, fuzzy rules are generated to map input data and class labels, offering a flexible approach to decision-making. FRBS can efficiently model complex decision boundaries, making worthy for multiclass problems. By leveraging fuzzy sets, these systems can account for overlapping class boundaries and provide a degree of membership to each class, rather than rigid, crisp decisions. This allows FRBS to solve classification problems in a way that closely mimics human reasoning, especially in cases where the data is uncertain or vague.

1.1.4. System Modelling

Fuzzy system modelling depicts system behaviour using fuzzy set rules and fuzzy variables. Fuzzy models based on Mamdani-type FRBS, are called linguistic models, as it has been seen that it provides interpretable, comprehensible model of the behaviour of the system. Other than that, where TSK or approximate Mamdani-type FRBS is used to model the behaviour of the system, is called the fuzzy modelling. These model focuses on accuracy compared to interpretability and descriptive power. Depending on

the requirements of the problem studied, the accuracy-interpretability trade-off is handled.

1.1.5. Tasks performed by Fuzzy Logic:

There are various tasks that could be performed on the basis of fuzzy logic. These tasks are shown in Figure 1.3 and short description is as follows:

- i. *Ranking analysis* is ordering alternatives (or solutions), based on set of criteria, from best to worst.
- ii. *Clustering* is a technique used to group inputs into meaningful subsets, known as clusters. The goal is to organize data in such a way that similar items are grouped together.

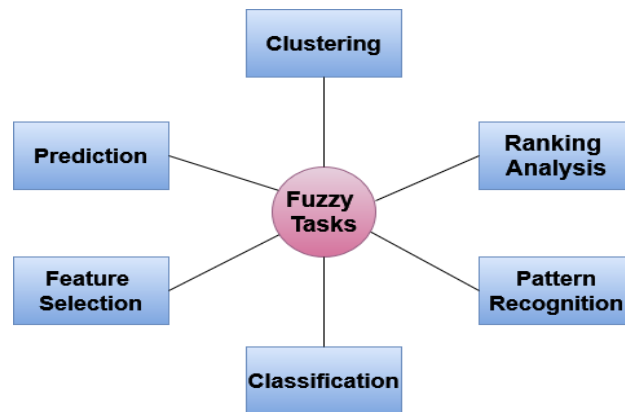


Figure 1.3: Various Fuzzy Logic Tasks

- iii. *Predictive analysis* is used to make future predictions. It is the branch of advanced analytics. Various techniques from machine learning, statistics, data mining, artificial intelligence, and modelling to generate forecasts based on current or historical data. This process helps in anticipating trends and making informed decisions.
- iv. *Classification* is done based on groups for the available classes.
- v. *Pattern recognition* is used for the identification of patterns and to check data is regular.
- vi. *Feature selection* is a methodology used to identify and remove irrelevant or redundant features in the feature space. It helps improve the accuracy and efficiency of the decision-making process, enhancing the performance of models and algorithms.

1.2. Feature Selection

It is a crucial step in optimizing both the efficiency and accuracy of models, as well as in reducing the risk of overfitting. It can be carried out by experts, either with or without the assistance of specific techniques designed for the task. Selecting relevant features is a key challenge in data mining and pattern recognition.

A feature is defined as a measurable property of an observable object, process, or phenomenon. By using a set of features, a machine-learning algorithm can perform classification. The feature selection problem can be viewed as searching for the optimal subset of features that minimize redundancy and maximize predictive capability. This process enhances the learning mechanisms of the rule-based system. Pattern recognition, machine learning, data mining, and statistical analysis are some of the fields where feature selection plays a critical role by improving model performance and decision-making accuracy.

1.2.1. Feature Selection Approaches

The techniques used for the selection of features can be divided into three main approaches:

- *Filter Method*: This approach uses the intrinsic properties of the features to select the optimized feature subset, instead of relying on a learning algorithm. Evaluation criteria are used to calculate the score of each feature (or feature set). Then, the features having the best scores are chosen. Univariate or multivariate measures can be used as evaluation criteria. Features are evaluated independently in univariate methods, while multivariate methods consider interactions between two or more features within the feature set. These methods are fast and scalable.
- *Wrapper Method*: In this approach, a learning algorithm is used to estimate the efficiency of different feature subsets. The objective is to determine feature set that enhance performance of the predefined learning algorithm. While effective, wrapper methods tend to have high computational costs compared to other methods, as they require repeatedly training the learning algorithm on different subsets. Additionally, these methods are typically influenced by the choice of learning algorithm (classification, regression, etc.) and are not limited to clustering tasks.

- *Hybrid Method*: This method combines the benefits of both filter and wrapper methods by leveraging the strengths of each. There are two approaches to hybridizing these methods. The first approach involves two stages: the feature set is first minimized using the filter method, and afterwards the final subset is selected from the reduced set using the wrapper method. The second approach uses a local search mechanism, applying the filter method within the wrapper method (or vice versa) to enhance the feature selection process.

1.3. Challenges in FRBS

The key challenges associated with learning and tuning FRBS include a) Dimensionality Reduction, b) Rule Optimization, c) Handling Uncertainty, d) Scalability, e) Membership Functions and parameter tuning, f) Overfitting, g) Interpretability vs Accuracy (Complexity).

1.3.1. Challenges in Dimensionality Reduction

When selecting a dataset for work, an idea about which features are important and which features are noisy, irrelevant, or redundant. Most of the real-life datasets are high-dimensional data with noisy, irrelevant, or redundant attributes (or features). These high-dimensional data create a challenge to existing learning methods to maintain the performance of the system. Firstly, due to a large number of features and a relatively small training set, leads to overfitting, thus performance degrades. Secondly, noisy, irrelevant, and redundant features also decrease the performance of FRBS in terms of computational time, memory requirements, accuracy, as well as interpretability of the system. Thirdly, storing and processing high-dimensional data is challenging.

1.3.2. Challenges related to Accuracy and Interpretability

In fuzzy modelling, the primary goals are interpretability and Accuracy. However, due to their inherently conflicting nature, it is often challenging to satisfy both simultaneously. In most cases depending on the problem to be addressed, more priority is given to one of them, leaving the other in the background. Thus, a trade-off exists between these two parameters.

1.4. Research Issues

The proposed work explores mechanism for dimensionality reduction and performance enhancement in FRBS. While FRBS have been effectively applied in various decision-making and classification tasks, several open research issues persist in improving their scalability, interpretability, and learning efficiency. This thesis explores feature selection methodology and learning mechanisms aimed at optimizing the overall performance of FRBS.

- **Feature Selection** - Feature selection is a crucial step in enhancing efficiency, accuracy and model generalization while preventing overtraining. It helps reduce computational complexity, storage requirements, and redundancy, while improving performance, create more generalized models. However, most existing approaches struggle to achieve a balance between dimensionality reduction and interpretability. Therefore, there is a need for an efficient feature selection mechanism that not only reduces dimensionality but also preserves meaningful information for FRBS learning.

The following issues are addressed:

- How to handle irrelevant, noisy, and redundant features?
 - How to transform high-dimensional datasets into compact, informational subsets?
 - How to prevent overfitting?
 - How to reduce computational complexity and memory usage during processing?
- **Learning Mechanism** – This determines how FRBS are trained and adjusted, including optimization, tuning of membership functions, and rule adaptation. This learning process is crucial because it shapes how the system interprets input data and makes decisions. However, current learning methods often face difficulties in automating rule generation, handling uncertainty, and achieving optimal membership tuning.

Key research issues include:

- How to automate fuzzy rule generation?
- How to handle uncertainty and imprecision in real-world data?
- How to adjust membership functions during learning?

- How to eliminate overlapping and redundant rules while preserving interpretability?
- **Tuning of FRBS** - Tuning of FRBS is essential to optimize parameters for achieving a balance between accuracy and interpretability. The following challenges are identified:
 - How to optimize membership functions for accurate input representation?
 - How to streamline the rule base for efficiency?
 - How to balance accuracy and generalization to avoid overfitting?
 - How to simplify the rule base for better interpretability?
 - How to enhance computational efficiency?

Identified Research Gap

Despite substantial progress in FRBS design and optimization, several key gaps remain:

- Most feature selection methods for FRBS focus on accuracy improvement but neglect interpretability and scalability.
- Learning and tuning processes often suffer from high computational cost and convergence to local optima.
- There is a lack of integrated frameworks that combine efficient feature selection, automated rule generation, and membership tuning under a unified model.
- Comparative analysis across diverse datasets to validate generalizability is still limited.

Addressing these gaps highlight the need for a hybrid and efficient feature selection and learning and tuning framework to achieve improved performance, interpretability, and computational efficiency.

1.5. Research Objectives

In view of the research issues identified above, the present work aims to develop a hybrid feature selection and learning framework for fuzzy rule-based systems (FRBS) that enhances both accuracy and interpretability while maintaining computational efficiency. Therefore, the researcher summarizes the following objectives:

1. To perform the systematic review on fuzzy rule-based system.
2. To design the feature selection mechanism for fuzzy rule-based system.
3. To develop the learning mechanism of fuzzy rule-based system for better performance.
4. To evaluate the proposed model with existing models on different parameters such as interpretability, accuracy, etc.

1.6. Research Methodology

The corollary of working on this Problem Statement is to achieve an optimized view in designing of the Fuzzy systems, which holds the significance of improvising the utmost parameters of interpretability and accuracy and providing us with a refined system. A flowchart as depicted in Figure 1.4 lists the activities performed to inculcate the stated parameters as amelioration to the design of an efficient Fuzzy Systems.

1.7. Thesis Contributions

The present study addresses the defined research questions by contributing to the enhancement of feature selection and learning mechanisms within Fuzzy Rule-Based Systems (FRBS). The major contributions of this thesis are summarized as follows:

- ***Comprehensive Literature Survey and Taxonomy Development***
 - a. A detailed survey and taxonomy of existing feature selection and learning mechanisms for FRBS have been developed to identify current research trends and gaps.
 - b. Feature selection methods have been systematically classified into supervised, semi-supervised, and unsupervised categories, as well as filter, wrapper, and hybrid approaches.
 - c. The taxonomy provides a structured understanding of various methods and their respective advantages, limitations, and application domains.
- ***Design of an Efficient Hybrid Feature Selection Approach***
 - a. An efficient hybrid algorithm has been designed and developed for handling high-dimensional datasets.

- b. The proposed method integrates filter and wrapper strategies to reduce data dimensionality, minimize computational complexity, and enhance classification performance.
- c. The algorithm has been validated on benchmark datasets and compared with existing feature selection methods to demonstrate its superiority in terms of feature reduction and accuracy.

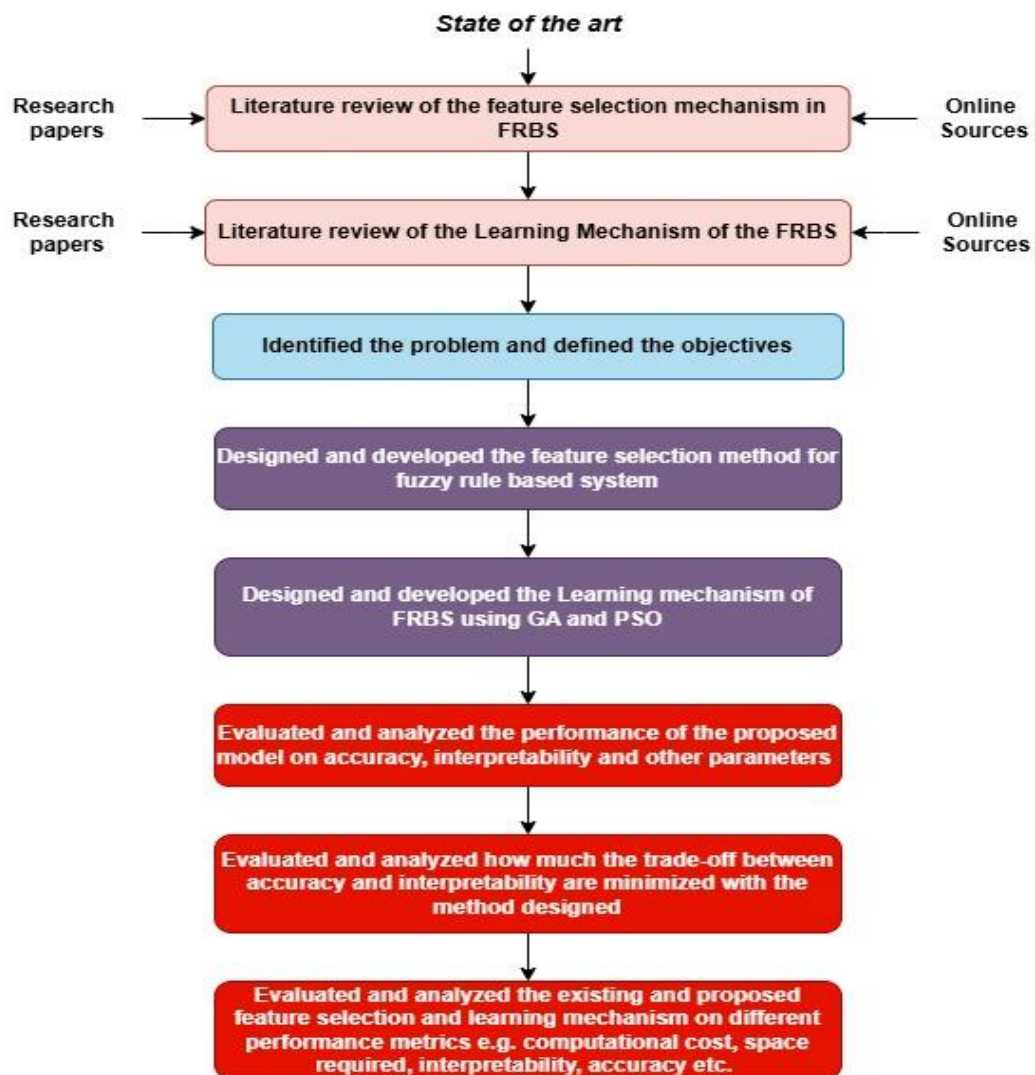


Figure 1.4: Research methodology flowchart

- ***Development of Robust Learning and Optimization Mechanisms for FRBS***
 - a. A robust learning mechanism for FRBS has been designed using Genetic Algorithms (GA) and Particle Swarm Optimization (PSO) to optimize fuzzy membership functions.

- b. Comparative analyses confirm that the integration of GA and PSO accelerates convergence and enhances the performance of FRBS across various benchmark datasets.

1.7. Thesis Organization

The overall organization of the thesis and its interdependencies are illustrated in Figure 1.5. Each chapter builds upon the preceding one, providing a logical progression from the literature survey to the development, optimization, and evaluation of the proposed frameworks. The chapters are organized as follows:

- Chapter 2 provides the methodological survey and taxonomy of feature selection methods and their learning techniques needed for FRBS. This chapter is partially derived from:
 - Kumar, A., & Kaur, A. (2023). Literature Review on Development of Feature Selection and Learning Mechanism for Fuzzy Rule-Based System. *Recent Advances in Computer Science and Communications (Formerly: Recent Patents on Computer Science)*, 16(4), 42-62. (Published, Indexed: Scopus).
- Chapter 3 covers a methodology to achieve Feature Selection on High Dimensional Datasets. This chapter is partially derived from:
 - A. Kumar, A. Kaur, P. Singh, Maha Driss, and Wadii Boulila, Efficient Multiclass Classification Using Feature Selection in High-Dimensional Datasets. *Electronics*. 2023; 12(10):2290. <https://doi.org/10.3390/electronics12102290> (Published, Indexed: Scopus, SCIE, Impact Factor: 2.6)
- Chapter 4 explains the learning and tuning methods devised for achieving the accuracy, interpretability, and other parameters, and to optimize the balance between the accuracy and interpretability of the FRBS. This chapter is partially derived from:
 - Kumar, A. & Kaur, A. (2021), Study of Various Fuzzy Approaches for Disease Diagnosis, 11th International Conference on Advances in Information Technology and Mobile Communication, AIM 2021. (Published, Indexed: Scopus)

- A. Kumar, A. Kaur, P. Singh, Advancement in Learning and Optimization Techniques for Fuzzy Rule-Based System (Hinweis International Conference on Recent Trends in Engineering and Technology, RTET 2024, Dec 2024. (Published)

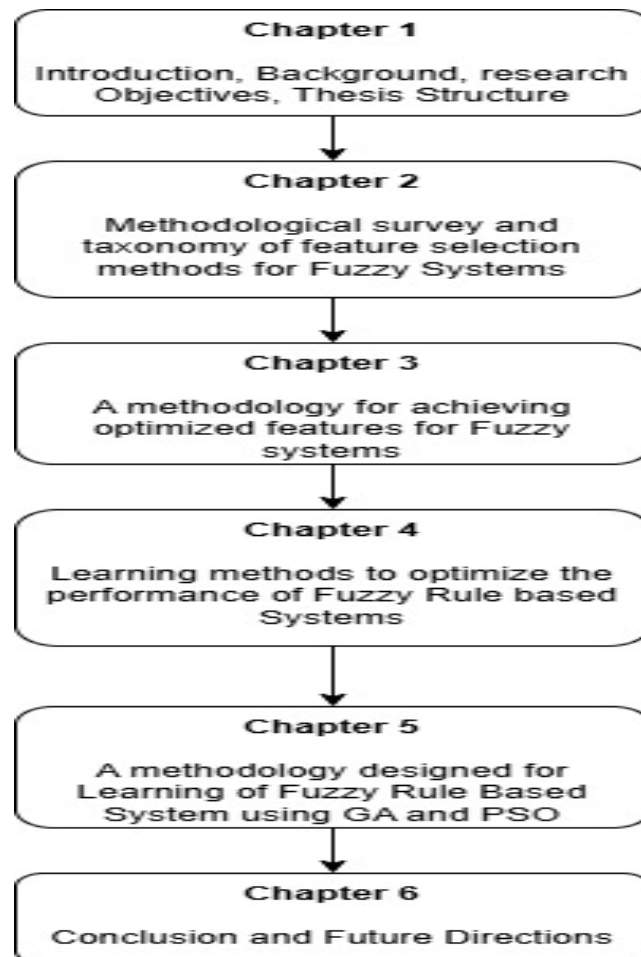


Figure 1.5: Chapter-wise thesis organization

- Chapter 5 covers the Learning mechanism of FRBS using GA and PSO.
 - Ankur Kumar, Avinash Kaur, Parminder Singh, Evolutionary-Swarm Optimization of FRBS for High-Dimensional Interpretability-Accuracy Balance (Accepted, International Journal of Fuzzy Systems, Indexing: Scopus, SCIE, Impact Factor: 3.6)
- Chapter 6 presents the overall conclusion of the thesis and introduce the possible future directions.

CHAPTER 2

LITERATURE REVIEW

2.1. Introduction

Feature selection (FS) [12, 13, 14] is an important step for optimizing effectiveness and accuracy and mitigating overtraining. Feature selection methods are not always used by experts for selecting optimized features. The selection of relevant features is an important problem of data mining and pattern recognition. A feature is a discrete assessable property of an observable object, process, or phenomenon. Using a set of features, a machine-learning algorithm can perform classification. The problem of selecting features can be formulated as a search for the optimal subset of features with the minimum redundancy and maximum predictive capability. Thus increasing the learning mechanism of the rule-based system. Pattern recognition [19], machine learning [20], data mining and statistical analysis [21] are some of the areas that have appearance of feature selection. Description of objects, which are under study in these areas, includes insignificant and unwanted features [22], specifically affecting the data analysis, resulting in bigotry or sometimes false models. FS keeps the most suitable ones to design a low dimensional space, while feature extraction combines the initial features transforming the real input space to another space having low dimensions. Furthermore, feature selection, other than decreasing the data dimensionality, which simplifies its visualization and interpretation, also contributes to building more robust models with improved generalization capabilities [23]. Feature selection and feature extraction are well-known techniques in data pre-processing (also called dimensionality reduction). The approaches related to extraction of features include Principal Component Analysis (PCA) [24], Linear Discriminant Analysis (LDA) [25], and Singular Value Decomposition (SVD) [26]. Moreover, supervised methods, unsupervised methods, and semi-supervised methods are employed for feature selection

methods depending on the information available in datasets. If the features in the dataset are large enough, the computational complexity is to be reduced to decrease the required storage and generalizable models are to be built, following which feature selection is needed. Feature extraction can be used for the same purpose. However, in case of data where readability and interpretability are important, since, we must keep the original features in reduced space, feature extraction is most likely to be used [27].

2.2. Feature Selection

The process of eliminating features that are insignificant and unwanted and have a harmful impact on model designing is called feature selection. A predefined criterion is used to estimate the optimality of the selected feature subset. It is applied in diverse tasks such as association rules, classification, and regression due to its better readability and interpretability, which it gets by keeping the meaning of original features as preferred. In the early stage of feature selection, methods were popular and focused especially in the field of classification of supervised learning tasks [24, 25], as explained and analysed by researchers. Feature selections are classified most of the time in three scenarios: supervised, semi-supervised, and unsupervised feature selection. As per the research, the following steps are to be applied in implementing feature selection: selection, evaluation, stopping criterion, and validation [30]. First, an already-defined search strategy (e.g., complete search, sequential search, and sequential floating search) is used to select a feature subset. Second, the initial criterion is established, and then the chosen features (subsets) are assessed based on this criterion.. Third, the subset that has the finest evaluation, after meeting the stop criterion, is considered. Lastly, the validation metric is applied to validate the chosen subset.

2.3. Feature Selection Approaches

Several methods have been developed to select the most optimal features. Supervised, semi-supervised, and unsupervised feature selection are further classified into three types depending on the stratagem used for feature selection:

1. *Filter Method*: This approach uses the general characteristics of data to select the optimized feature subset instead of a learning algorithm. The evaluation of features

is based on the inherent properties of the data. Evaluation criteria are applied to calculate the score of each feature (or subset). Then, the features with highest scores are chosen. This method generally uses statistical methods for the evaluation of a subset of features. Univariable (aka univariate) or multivariable (aka multivariate) measures can be used for evaluation criteria. Features are evaluated independently in univariable methods, while multivariable methods consider multiway relationships within the feature set. These methods are speedy and scalable.

2. *Wrapper Method*: A learner is required to evaluate the goodness of possible feature subsets. A better feature subset is obtained through the wrapper, which enhances the performance of the predefined learning algorithm. It has high computational costs compared to other methods and is usually not used in combination with a specific clustering algorithm. Sometimes, cross-validation is used for the evaluation of features.
3. *Hybrid Method*: This method tries to take advantage of both the above methods by exploiting the qualities inherited from filter and wrapper methods. There are two hybridization approaches to hybridize filter and wrappers method. The first method has two stages in which the feature set is reduced using the filter approach, and then a final set is obtained from the reduced set using the wrapper method. The second method as a local search mechanism is using a filter (wrapper) approach in a wrapper (filter) approach.

2.4. Benefits and Drawbacks of the Feature Selection Approach

The following Table 2.1 shows the general benefits and drawbacks of feature selection approaches.

2.5. Unsupervised Feature Selection

UFS methods are categorized into filter, wrapper, and hybrid methods based on the stratagem used for feature selection.

1. *Filter Approach*: Based on this approach, UFS methods are classified as univariate and multivariate [31] filter methods. Univariate filter methods can be categorized as methods that select the features on how relevant they are depending on the

concepts of information theory [32] and others are evaluating features based on manifold learning (also called spectral analysis) [29, 30] using how much the objects are similar among themselves.

1.1. Univariate Filter Methods: These are also known as UFS methods based on ranking. These methods get an ordered list (ranking) of features using some criteria for evaluating each feature. This order is then used to select the final feature subset. These methods are suitable for identifying and removing irrelevant features. Due to not considering the possible dependencies among features, redundant features are not removed by these methods, which is the disadvantage of this method. Univariate methods can be categorized as

- a. Information-Based methods: Feature evaluation is most frequently done with these methods i.e., for instance, entropy, divergence, and information gain. In this method, the idea of measuring the quantity of data dispersed using methods such as entropy, J-measure, and information gain to find cluster structures in data.

Table 2.1. Positive and negatives of the feature selection approach

Methods	Positive	Negatives
Filter	This method is fast compared to other approaches.	Interaction with clustering algorithms is ignored.
	This approach is scalable.	These methods do not remove multicollinearity.
	It deals with data parallelly.	It might fail to find the best subset of features on many occasions.
	A clustering algorithm is not used here.	
Wrapper	This approach depends on the clustering algorithm.	Overfitting risk is more in the wrapper approach.
	Feature dependencies can be modeled in this approach.	The computational cost is high.
	Can always provide the best feature set.	The selection of the clustering algorithm is specific.
Hybrid	This approach is dependent on the clustering algorithm.	The selection of the clustering algorithm is specific.
	Feature dependencies can be modeled in this approach.	
	Time consumption is less.	

- b. Spectral Similarity Based method: Moreover, called manifold learning or spectral analysis. A similarity matrix, is calculated, and then from that

matrix, the eigenvalues of the eigen system of Laplacian or normalized Laplacian matrices for modeling the data structure is calculated.

- 1.2. Multivariate Filter Methods: These methods jointly evaluate the relevancy of features compared to the individual relevancy of features. Unwanted and insignificant features are easily handled by these methods. Learning algorithms, in most cases, provide better-constrained accuracy when applied to the feature subset chosen by multivariate methods compared to that achieved by using univariate methods. Multivariate methods can be categorized as
 - a. Statistical/Information-Based Methods: These consist of an unsupervised feature selection method that selects features through statistical measures like information gain, mutual correlation, variance, etc. among others.
 - b. Bioinspired-based methods: These unsupervised feature selection methods find a good feature subset by satisfying some criterion of quality using stochastic search strategies such as Monte Carlo Search, and Tabu Search.
 - c. Spectral/Sparse Learning: These methods are spectral analysis and sparse learning based. In this learning, a constrained regression model is used for optimizing the feature selected at the time of the learning process. Thus, this method is sometimes known as the embedded method.
2. *Wrapper Approach:* Wrapper methods are categorized as sequential, bio-inspired, and iterative when classified in terms of features searching using some methodology. The addition or removal of features is done in sequential order in the first approach. Sequential search methods are fast and implemented easily. Bio-inspired methods incorporate fickleness into the method of searching to avoid local bests. The iterative approach handles the UFS problem by trying to avoid combinatorial search by framing it as an estimation problem.
3. *Hybrid Approach:* This approach tries to take maximum benefit from the filter and wrapper approaches by selecting the best out of them. In the filter stage of this approach, the ranking of features is done by applying some measures depending on the data's inherent properties, while in the wrapper, particular clustering algorithms

are applied to evaluate certain features for finding the best one. Hybrid approaches are categorized into ranking-based methods and non-ranking based methods.

2.6. Semi-supervised feature selection methods

Most of real-world application has unlabelled data compared to labelled data. Supervised feature selection methods deal with labelled data, while unsupervised feature selection methods deal with unlabelled data. No approach was there which could deal with both types of data simultaneously. This motivated the researchers to develop an approach, called semi-supervised learning, which uses both data to evaluate the feature relevance. In a study [35], the semi-supervised approach was initially developed that utilized both supervised and unsupervised data for selecting features.

Semi-supervised learning should have a minimum supervised data and maximum unsupervised data for its learning [32, 33, 34]. There are a few smoothness assumptions, mainly cluster assumption [39] and manifold assumption which should be met while dealing with semi-supervised learning [32, 36]. Cluster supposition says that any sample will belong to that class only if they are part of the same cluster [40]. Manifold assumption states that the high-dimensional data lie on a low-dimensional manifold formed by the data [33, 36, 37, 38].

These methods are classified as generative models, self-training, co-training, and Transductive SVM. Here, the two most prominent taxonomies to classify feature selection methods of semi-supervised learning are discussed. The foremost perspective categorizes them on the basis of how they interact with the learning algorithms (also called feature selection basic taxonomy). The other perspective categorizes these into several types depending on the learning procedure. Figures 2.1 and 2.2 shown below classify these taxonomies. Depending on the above taxonomies, some of the important methods are:

2.6.1. *Graph-based methods*: Here, samples of training data are used to construct a graph. Graph vertices represent the labelled and unlabelled samples of training data. The similarity between two samples is represented by the undirected edges between two vertices. These methods are capable of preserving the geometrical structure of the graph, and thus construct a neighbourhood graph to identify the relevant features. Some of the graph-based methods include Laplacian score,

Fisher criterion, pairwise constraints, sparse models, ensemble learning, and spectral graph theory & cluster assumption.

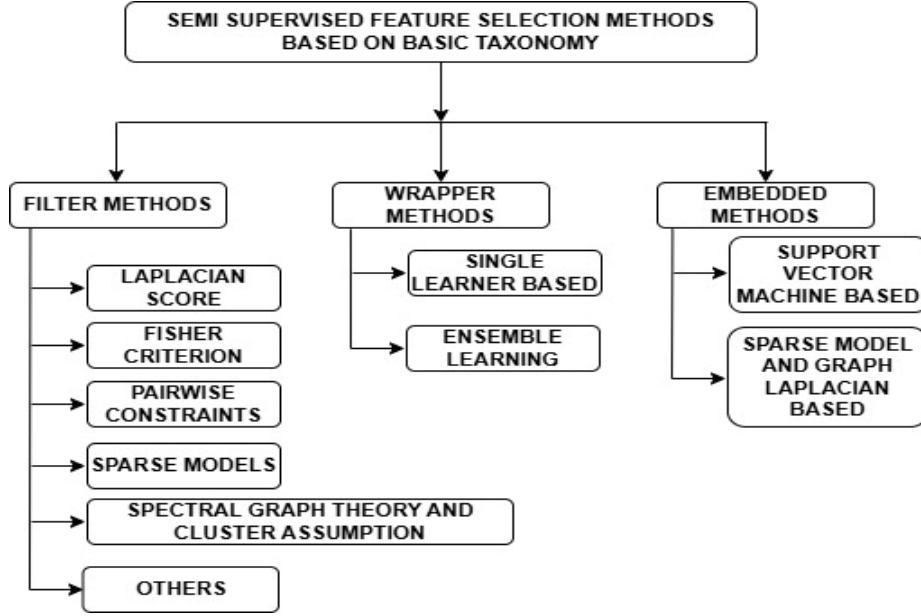


Figure 2.1. Basic taxonomy-based semi-supervised methods of feature selection

2.6.2. *SVM-based methods*: These methods principally measure the distance between two observations using kernels as the fundamental mechanism. SVM based methods include sparse models, feature recursive elimination based, and manifold regularization methods.

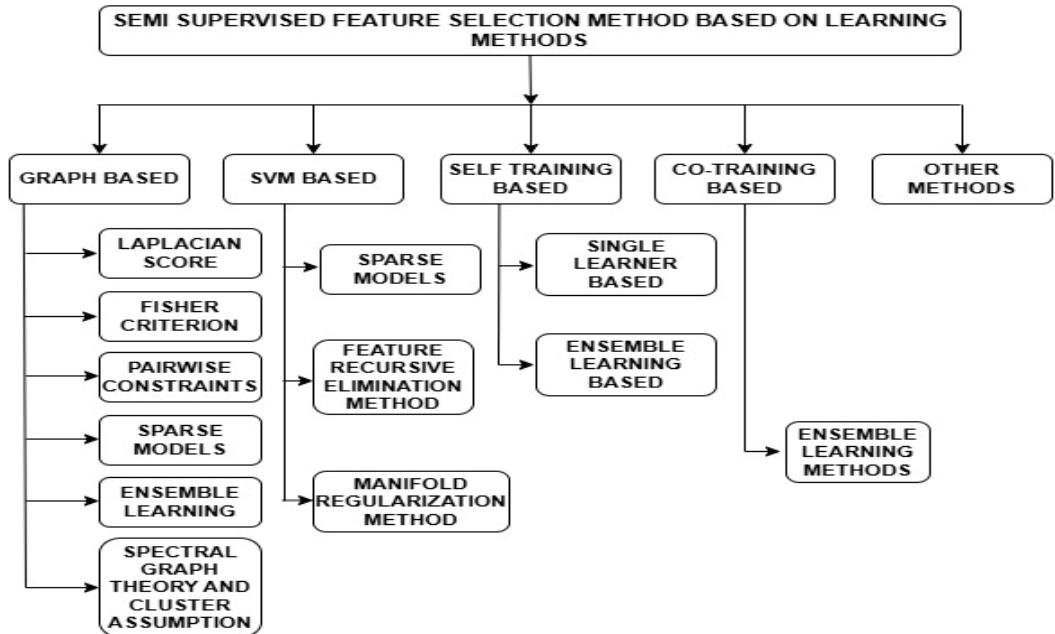


Figure 2.2. Learning method-based semi-supervised methods of feature selection

- 2.6.3. *Self-training based:* In this method, the training of a classifier is done first on labelled data to apply it to unlabelled data for prediction of labels. By this the most confident data which is unlabelled, is extracted. A training set is then created from the subsets of this predicted labelled data. Then the classifier trains on the new data set, and the method is continued. Self-training is very well understood by the fact that the classifier teaches itself by using its own predictions. Self-training-based methods include single-learner-based and ensemble learning-based methods.
- 2.6.4. *Co-training based:* As the names suggest, it trains two different classifiers. For this, it takes the labelled data and selects a different feature set for training of these classifiers and then these classifiers are used to classify unlabelled data. Ensemble learning is one of the methods based on this training.
- 2.6.5. *Filter Methods and Graph-Based Methods*
- a. Laplacian Score: This unsupervised method is also used in semi-supervised methods of selecting features. It computes the Laplacian Score of each feature reflecting its ability to preserve locality [43]. Laplacian score is based on the observation that if two data points are nearby, they are likely to belong to the same class and thus the feature is well-thought-out to be a good one. Most of the problems based on learning, such as classification, give importance to the local structure of the data space compared to its global structure. The nearest neighbor model is constructed to capture the local geometric structure of the data space. It pursues those features that admire this graph structure. The features are also capable of preserving the global geometrical structure of the data in addition to the local geometrical structure of the data space, thus making it available for semi-supervised feature selection methods.
 - b. Fisher Criterion or Fisher Score: A supervised method is used for calculating the ratio between classes to class. It seeks features having the finest discrimination ability. The methodology of the Fisher criterion involves calculating the score of each feature based on the data samples. For this, the distance of data samples in the different classes is maximized, and also within the same class is minimized by calculating the mean and variance. Once a score

is assigned to each feature, the features with the highest Fisher score are selected.

- c. Pairwise Constraints: This method can simply specify that data inputs must be grouped together (must-link constraints) or cannot be grouped together (cannot link constraints). It simply means that a pairwise constraint between two items indicates whether they belong to the same class or not.
- d. Sparse Models: This method is used to select features that are not dense and judicial.

2.6.6. *Wrapper Methods and Self Training Based or Graph Based or Co-Training Based*

- a. Single Learner Based: This learning technique is semi-supervised and the methodology is wrapper based. It is also self-training or co-training classified according to the second taxonomy of semi-supervised methods. In this, any of the supervised selection processes can be applied to generating the feature subsets. Then, the classifier is trained on the selected features by envisaging the labels of unlabelled data. Afterward, a new training data set is formed by randomly selecting a subset of unlabelled data along with the predicted labels and adding it to the labelled data. The selection of features is done on this new training set using the supervised FS method and the learning model. This process of selecting a subset of features is repeated until the size of the subset of features reaches a predefined number.
- b. Ensemble Learning: It is a semi-supervised wrapper feature selection method and a self-learning-based method. It is also called a multiple classifier system. This learning technique can strategically generate several models such as classifiers or experts and combines them to solve computational intelligence problems. This technique mostly improves the performance (classification, prediction, function approximation, etc) of the model or minimizes the likelihood of an unformulated selection of a less important one. (Ensemble method solutions have higher accuracy compared to a single model-based). The advantages of ensemble learning are performance and robustness.

2.6.7. *SVM Based*

- a. Manifold Regularization methods: It is a technique for using the shape of a dataset to constrain the functions that should be learned on that dataset.

Table 2.2. Semi-supervised method classification

			The semi-supervised method based on the selection of features											References
			Graph-Based					Self-training		Co-training	SVM			
			Graph-based and cluster	Laplacian score	Pairwise constraints	Fisher criterion	Sparse Models	Ensemble learning	Single learner	Ensemble learning	Manifold regularization	Feature recursive elimination	Sparse models	Other models
Semi-supervised learning based on basic taxonomy	Filter methods	Graph theory and cluster	Y											[35]
		Laplacian score		Y										[44]
				Y										[45]
				Y										[46]
				Y										[47]
		Pairwise constraints				Y								[48]
					Y									[49]
					Y									[50]
					Y									[51]
		Fisher criterion				Y								[52]
						Y								[53]
						Y								[54]
					Y								[55]	
	Sparse models											Y	[57]	
	Other models											Y	[58]	
	Wrapper methods	Single learner based						Y						[59]
		Based on ensemble learning					Y							[59]
									Y					[60]
									Y					[60]
	Embedded methods	Sparse models and graph Laplacian					Y							[61]
							Y							[62]
							Y							[63]
							Y							[64]
		Support vector machines (SVM)										Y		[65]
											Y			[66]
												Y		[67]
												Y		[68]

On the basis of the above taxonomies, a detailed classification shown in Table 2.2. The table shows that the majority of the literature on semi-supervised methods are filter-based methods, and many of these filter-based methods are based on the graph. Few research works have concentrated on the wrapper and embedded methods. Some research has explored self-training, co-training, and SVM, with a few of these also utilizing graphs.

2.7. Findings from the literature

The research has been done on various unsupervised methods. This literature survey covers the research in chronological order (Table 2.3).

Table 2.3. Literature Survey

References	Method	Indexing of the journal	Main findings	Remarks
[28]	Univariate Filter – Information-based method Sequential backward selection method for Unsupervised Data (SUD)	IEEE Computer Society	The classifier used is C4.5. Tenfold cross-validation is used for validating error rate.	Ranking of feature from most relevant to least relevant
[69]	Univariate Filter – Information-based method Singular Value Decomposition Entropy	Bioinformatics journal	Used QC for validation, Clustering algorithm is 100 times, and the average of the Jaccard score is taken	Leave one out comparison method is used
[70]	Univariate Filter – Information-based method Adaptive floating Search Method	IEEE Transactions	K- means is applied to calculate the objective function.	Solve the drawbacks of SVD – Entropy. The methodology used tries to remove a few pitfalls of Singular Value Decomposition entropy.
[71]	Univariate Filter – Information-based method	Springer	C4.5 decision tree classifier is used for working	Tenfold cross-validation is used to find the error in classification and clustering.
[72]	Univariate Filter – Information-based method	Elsevier	Fuzzy C-means clustering algorithm is used. Sammon's Mapping is used for dimensionality reduction.	To solve the drawbacks of Singular Value Decomposition entropy. Features having the same constant value are distinguished by this methodology which could

References	Method	Indexing of the journal	Main findings	Remarks
				not be handled by Singular Value Decomposition entropy.
[73]	Univariate Filter – Information-based method	Journal of Physics	Genetic Algorithm and information gain is used for feature selection	Dataset used is KDD CUP 99 Detection rate is 99.7% and false positive rate is 0.2%
[74]	Univariate Filter-spectral similarity-based method	NIPS	K-means Laplacian Score is used for dimensionality reduction.	NMI & ACC are used as validation measure
[75]	Univariate Filter-spectral similarity-based method SPEC (spectrum decomposition)	Elsevier	1 Nearest Neighbour classifier is used for classifying objects. 10 trials are used for classification accuracy.	It is a generalization of Laplacian Score
[76]	Univariate Filter-spectral similarity-based method	IEEE	5 nearest neighbour classifiers are used for classification Average classification accuracy is calculated using tenfold cross-validation.	Laplacian++ (Extended version of Laplacian score) Features are evaluated globally, not locally
[77]	Univariate Filter-spectral similarity-based method USF Spectral Method for mixed data- USFSM	Elsevier	Classifiers used are 3 Nearest Neighbour, Support Vector Machines, and Naïve Bayes, and the algorithm used is the K-prototypes clustering algorithm	Tenfold cross-validation is used to calculate classification accuracy. Redundancy rate
[78]	Multivariate filter-Statistical/Information based methods	IEEE	None	Correctness of the selected features is done.
[79]	Multivariate filter-Statistical/Information based methods	Intelligent Data Analysis	Selected features that are highly correlated COBWEB clustering algorithm is used.	Two-fold cross-validation is used to calculate the average error rate.
[80]	Multivariate filter-Statistical/Information based methods FSFS (Feature Selection using Feature Similarity)	IEEE Transactions	K-nearest neighbor and Bayes classifier were used.	Clustering is done.
[81]	Multivariate filter-Statistical/Information based methods	Springer	Naive Bayes classifier	Tenfold cross-validation is used for calculating classification error.

References	Method	Indexing of the journal	Main findings	Remarks
[82]	Multivariate filter-Statistical/Information based methods	ACM	3NN classifier is used for classification. It is a hierarchical method.	For validation, they used FFEI values.
[83]	Multivariate filter-Statistical/Information based methods	IEEE Computer Society	No classifier was used.	Unwanted features are removed.
[84]	Multivariate filter-Statistical/Information based methods MPMR (maximum projection and minimum redundancy)	Elsevier	Not specified	Validation measure NMI and ACC were used.
[85]	Multivariate filter-Statistical/Information based methods RRFS (relevance redundancy feature selection)	Elsevier	SVM, Naïve Bayes and 3 Nearest Neighbour classifiers are used	Features selection in two steps, namely relevance measure and feature similarity measure
[86]	Multivariate filter-Statistical/Information based methods GA for extracting features	Springer	KDD99 benchmark dataset is used	Accuracy of 99.27% is found. 7 out of 41 features are selected using genetic algorithm
[87]	Multivariate filter – bio-inspired method Use ant colony optimization for feature selection	Elsevier	SVM, decision trees, and Naïve Bayes classifiers	Random train/test partitions were done to calculate the average error rate.
[88]	Multivariate filter – bio-inspired method RR-FSACO- feature selection through relevance redundancy	Elsevier	SVM, decision trees, and Naïve Bayes classifiers	Random train/test partitions were done to calculate the average error rate.
[89]	Multivariate filter – bio-inspired method Microarray gene selection based using ACO	Elsevier	SVM, decision trees, and Naïve Bayes classifiers	Random train/test partitions were done to calculate the average error rate.
[90]	Multivariate filter – bio-inspired method Probabilistic feature selection using ACO is used (UPFS)	Elsevier	SVM	Random train/test partitions were done to calculate the average error rate.
[91]	Multivariate filter – bio-inspired method	Elsevier	Convolution Neural network is used for feature selection	Accuracy received is 97.7%

References	Method	Indexing of the journal	Main findings	Remarks
[92]	Multivariate filter – bio-inspired method	Elsevier	KDD CUP 99 is used as dataset. K means algorithm is used for intrusion detection.	Accuracy received is 98.6% & FPR is 0.625%.
[93]	Multivariate filter – bio-inspired method	IEEE Access	NSL KDD dataset is considered for working. Modified cuckoo search algorithm is used for feature selection	Accuracy received 98.81%
[94]	Multivariate filter – bio-inspired method	Elsevier	Particle Swarm Optimization is used for feature selection.	Detection rate: 4% to 6% (increased) False Alarm Rate: 1% to 5% (reduced)
[95]	Multivariate filter-Spectral/sparse learning methods Feature selection by minimum spectral redundancy	IEEE	Spectral Clustering Method spectrum decomposition-based feature selection algorithm used	Clustering error is calculated.
[96]	Multivariate filter-Spectral/sparse learning methods Feature elimination is done using LDA	ACM	Classifier used in nearset mean	Mean error rate is found here, which is over 100.
[97]	Multivariate filter-Spectral/sparse learning methods	ACM	K-means clustering is used for clustering.	Random train/test partitions were done to calculate the average error rate.
[98]	Multivariate filter-Spectral/sparse learning methods Multi-cluster FS algorithm is used	ACM	K-means clustering is used for clustering	NMI
[99]	Multivariate filter-Spectral/sparse learning methods	ACM	SVM classifier	Average score is calculated using Jaccard, and classification accuracy is calculated.
[100]	Multivariate filter-Spectral/sparse learning methods UDFS (Unsupervised Discriminative Feature Selection Algorithm)	ACM	K-means clustering is used for clustering.	NMI & ACC are used as validation measure

References	Method	Indexing of the journal	Main findings	Remarks
[101]	Multivariate filter-Spectral/sparse learning methods NDFS (non-negative discriminative feature selection)	ACM	K-means clustering is taken.	NMI & ACC are used as validation measure
[97, 98]	Multivariate filter-Spectral/sparse learning methods JESLR	IEEE Transactions	K-means clustering is used for clustering.	NMI & ACC are used as validation measure
[104]	Multivariate filter-Spectral/sparse learning methods SPFS (Similarity preserving feature selection)	IEEE Transactions	Linear SVM classifier is used for classification.	Classification accuracy is calculated.
[105]	Multivariate filter-Spectral/sparse learning methods Clustering Guided Sparse Structural Learning	IEEE Transactions	K-means clustering is used for clustering	NMI & ACC are used as validation measure
[106]	Multivariate filter-Spectral/sparse learning methods RUFFS (Robust Unsupervised feature selection)	ACM	K-means clustering is used for clustering	NMI & ACC are used as validation measure
[107]	Multivariate filter-Spectral/sparse learning based	IEEE	K-means clustering is used for clustering	For validation, NMI & ACC are used.
[108]	Multivariate filter-Spectral/sparse learning methods CDLFS (Couple Dictionary Learning)	ACM	K-means clustering is used for clustering	Validation is done through NMI & ACC.
[109]	Multivariate filter-Spectral/sparse learning methods SOGFS (Structural Optimal Graph Feature Selection)	ACM	K-means clustering	ACC is used for validating
[110]	Multivariate filter-Spectral/sparse learning methods	Springer	5 Nearest Neighbour classifier is used for classifying objects	Classification accuracy is calculated

References	Method	Indexing of the journal	Main findings	Remarks
	GRRSR (Graph Regularized Non-negative self-representation)			
[111]	Multivariate filter-Spectral/sparse learning methods	Elsevier	Manifold structure is uncovered by the algorithm	Accuracy metrics are used for validation
[112]	Multivariate filter-Spectral/sparse learning methods RUFMS (Robust Unsupervised Feature Selection via Matrix Factorization)	Elsevier	Clustering performance is improved	NMI and ACC are used as validation measures
[113]	Multivariate filter-Spectral/sparse learning methods Regularized Self-representation model is proposed	Elsevier, ACM	For clustering K -means is used	Normalized Mutual Information and accuracy metrics are used as authentication measures
[114]	Multivariate filter-Spectral/sparse learning methods FSR	IEEE	Most representative features are identified, which helps in constructing other features.	Normalized Mutual Information and accuracy metrics are used as authentication measures
[115]	Multivariate filter-Spectral/sparse learning methods LRSL	Elsevier	Clustering is done through the k-means algorithm	NMI & ACC are used as a validation measure
[116]	Multivariate filter-Spectral/sparse learning methods	Elsevier	Clustering is done through the k-means algorithm	NMI & ACC are used as a validation measure
[117]	Multivariate filter-Spectral/sparse learning methods DSRMR	Elsevier	Clustering is done through the k-means algorithm	NMI & ACC are used as a validation measure
[118]	Multivariate filter-Spectral/sparse learning methods $l_{2,1}$ – UFS ($l_{2,1}$ based graph regularized UFS method)	Elsevier	K-means clustering is used for clustering	Normalized Mutual Information and accuracy metrics are used as authentication measures

References	Method	Indexing of the journal	Main findings	Remarks
[119]	Multivariate filter-Spectral/sparse learning methods	IEEE Transactions	K-means clustering is used for clustering	Normalized Mutual Information and accuracy metrics are used as authentication measures
[120]	Multivariate filter-Spectral/sparse learning methods	IEEE Transactions	K-means clustering is used for clustering	NMI and Precision metrics (ACC) are used as validation measure
[121]	Wrapper- sequential methods	ACM	COBWEB clustering algorithm based on a hierarchical concept is used.	Accurate prediction of the class labels of unknown data
[122]	Wrapper- sequential methods	Elsevier	Expectation Maximization and K-means algorithm are used	Class error rate is calculated, and Naïve Bayes is used as a classifier
[123]	Wrapper- sequential methods	IEEE	Clustering is done through the k-means algorithm	Class error
[124]	Wrapper- sequential methods	Elsevier	Clustering is done through the k-means algorithm	Adjusted rand index Precision recall and F-measure
[125]	Wrapper-bioinspired method	Intelligent Data Press (IOS)	K-means & EM is used for clustering.	F-score method is used to calculate accuracy.
[126]	Wrapper-bioinspired method	Intelligent Data Press	Partitioning clustering algorithms are used	Missing features data are easily handled Accuracy metrics are used for validation measure
[127]	Wrapper – bioinspired method	Elsevier	Multi parent mutation with SVM is used for feature selection	Accuracy of 99.3%
[128]	Wrapper-bioinspired method	Neural Information Processing System (NIPS)	Expectation Maximization clustering technique is used.	Stability of data partitions
[129]	Wrapper-bioinspired method	IEEE Transactions	Expectation Maximization clustering technique is used.	Gaussian mixture-clustering is solved
[130]	Wrapper-bioinspired method	IEEE Transactions	None	Validation is done through accuracy metrics
[131]	Wrapper-bioinspired method	ACM	For clustering, K-means algorithm is used	Normalized Mutual Information (NMI) and Accuracy metrics (ACC) are used as validation measures
[132]	Wrapper – Iterative method	IEEE	For clustering, K-means algorithm is used	ACC is used for validation measure

References	Method	Indexing of the journal	Main findings	Remarks
[133]	Wrapper – Iterative method	AAAI	For clustering, K-means algorithm is used	ACC and NMI are used for validation measure
[134]	Hybrid- based on ranking	Springer	For clustering, K-means algorithm is used	Feature ranking Impurity
[135]	Hybrid-non ranking based	IEEE	For clustering, K-means algorithm is used	Class error
[136]	Hybrid – ranking	IEEE	3NN classification is used for classification. It is a hierarchical method.	Feature ranking is used for selecting features
[137]	Hybrid – non ranking	IEEE	Expectation Maximization clustering technique is used.	Classification error using tenfold
[138]	Hybrid – based on ranking	ACM	For clustering, K-means algorithm is used	Jaccard index and global silhouette Retention and Run-time

2.8. Critical Insights from Unsupervised and Semi-Supervised Feature Selection Methods

This section presents a comparative analysis of various unsupervised and semisupervised feature selection methods. The preceding sections reviewed these approaches according to their categories, types, and underlying learning mechanisms. Here, the key advantages and disadvantages of different methods are summarized to highlight their strengths, weaknesses, and practical considerations.

Table 2.4 provides a comparison of unsupervised feature selection techniques, focusing on their main benefits and limitations. Similarly, Table 2.5 presents advantages and disadvantages of semi-supervised feature selection methods based on learning methods. As it is seen from Table 2.4, no UFS method is complete in itself for every type of data. Every approach has its positive and negative aspects. However, with the literature review done in the section 2.8 and from Tables 2.3 and 2.4, some positive and negative aspects of different approaches with their categories are highlighted.

Table 2.4. Comparative Analysis of UFS methods with their category

Category/Subcategory	Method	Advantages	Disadvantages
Univariate-information based	Filter	It can represent linear and nonlinear relations.	Feature correlation is ignored

Category/Subcategory	Method	Advantages	Disadvantages
		This approach is neutral towards data dimension	
Univariate-spectral/similarity based	Filter	Provides a commanding outline for UFS.	Feature correlation is ignored
Multivariate-statistical/information based	Filter	- Dependencies among features are easily modelled - It takes less time compared to wrapper methods	Compared to univariate methods, it is less scalable and slower
Multivariate-bio-inspired	Filter	Feature dependencies are modelled by this approach	- More space(memory) is required. - It is slower compared to univariate methods.
Multivariate-spectral/sparse learning based	Filter	Redundant features can easily handle	Slower and less scalable compared to univariate methods
Sequential	Wrapper	Implementation is easy	- Overfitting risk - Local optima prone
Bio-inspired	Wrapper	Feature dependencies can be modelled	- Compared to sequential based methods, it has high risk of overfitting. - More space is required
Iterative	Wrapper	- Feature dependencies can be modelled. - Simultaneously, feature selection and clustering can be done	Overfitting risk
Ranking based	Hybrid	Relevant individual features are easily recognized and nominated for the feature grading(rank).	Performance in terms of quality is reduced when integrating filter and wrapper approaches.
Non-ranking based	Hybrid	Other approach ideas may also be exploited	Individual features which are relevant cannot be easily identified.

Studying the literature review in detail, we can find that wrapper methods are minimum in number compared to filter methods for UFS. The reason behind this is that for high dimensionality problems, wrappers are not so useful and hence seldomly used. If the balance between the efficiency and quality of the subsets of features that have been selected can be compromised, then the hybrid methods can be used compared to the wrapper ones.

The literature survey on UFS methods done in section 2.8 shows that filter methods are more focused compared to wrapper and hybrid methods. It has been observed that a lot of technological advancement has been done in the last few years, due to which a large volume of unlabelled data is engendered in various scientific areas such as data mining,

social networking, trespassing, and many more. For these areas, fast and scalable methods are helpful and the filter approach fits with this approach. In UFS filter approach, there is no use of clustering methods for evaluating features, they only depend on the fundamental properties of data. Thus, this approach is not influenced (biased) towards any specific learning algorithm. These methods can be easily designed, and understandable and are usually faster than other approaches, making them attractive for high-dimensional data.

There are a few prominently used norms for UFS methods that are available to determine the relevant features:

- i. The first criterion is picking those features that sustain the manifold structure of the original data. (UFS filter methods use these criteria in mostly spectral and sparse learning-based methods)
- ii. Second, the criterion uses a clustering algorithm to search for pseudo labels (or cluster indicators), thus reducing UFS to a supervised context (filter, wrapper, and hybrid approaches use this in multivariate learning, especially spectral or sparse learning).
- iii. The third criterion is based on feature dependency i.e. analysis of correlation among features is done by selecting a subset of features with the peak or deepest correlation among features. (filter approach-based Multivariate Statistical methods use it).

Feature correlation can also be used for defining feature redundancy. Two main approaches available for this are:

- i. The objective concept is not considered for quantifying redundancy.
- ii. The objective concept is considered for quantifying redundancy.

Performance evaluation for UFS methods can be done in three ways:

- i. The quality of each feature is evaluated on the feature subset of the exact supervised/unsupervised classifier.
- ii. The redundancy of the features that have been selected is evaluated.
- iii. The correctness of the features that have been selected is evaluated.

Table 2.5.: Advantages and disadvantages of semi-supervised feature selection methods based on learning methods

Process	Advantages	Disadvantages
Graph-based	• Its computational performance is high compared to other methods. • It is highly generalized.	• Individual features are evaluated and feature dependencies are ignored. • Interaction with the classifier is ignored.
Self-training, and co-training based	• It interacts with the classifier. • Features are evaluated combinedly, also dependency among these features is considered.	• Computation is expensive • Many models are built, which is a time-consuming process. • Overfitting is susceptible • Selection is classifier dependent.
SVM- based	• Classifier interaction is there. • Features are evaluated combinedly, and the dependencies among these features are considered • Computational complexity is small compared to self-training and co-training-based methods.	• Selection is classifier dependent.

2.9. Summary

Having their ability to select features from unlabelled data, researchers in different research areas have a large interest in UFS methods. This paper covers the survey of relevant unsupervised feature selection methods from scratch to the current work done. A detailed taxonomy of Unsupervised Feature Selection methods with their positives and negatives has been analysed. The survey shows that the maximum quality research time is spent in the area of filter approach due to its low computational cost and ease of working compared to wrapper and hybrids, making it suitable for high dimensionality datasets.

The previous decades also focussed on semi-supervised FS methods. In this study, different methods of selection of features based on semi-supervised are reviewed. This research focused on semi-supervised feature selection methods as two taxonomies, covering two different perspectives of this approach. The first classification of the semi-supervised approach is based on the feature selection approach and another is on learning approaches.

Filter, wrapper, and embedded methods are the classification of semi-supervised approaches according to the first arrangement. Moreover, it was observed that most research works are conducted on the filter approach rather than wrapper approaches. Filter approaches work fast as they do not depend on any classifier and have low computational cost. These methods can effortlessly gauge up to very high-dimensional data sets. The experimental results done by researchers show that this approach is decent for selecting features in vast applications. Graph-based approaches have low computational costs and can easily be generalized. This is the reason for using this feature selection method in many applications.

This chapter provided a detailed review of feature selection approaches, The analysis covered their advantages, limitations, and comparative effectiveness. From the reviewed studies, it is evident that while many algorithms have achieved improved classification accuracy, interpretability, scalability, and computational efficiency remain key challenges.

To address these challenges, the present research proposes hybrid feature selection and learning mechanism designed to enhance performance and interpretability while maintaining computational efficiency. The insights gained from this review directly inform the methodological design presented in Chapter 3 to 5.

CHAPTER 3

MISFS ALGORITHM FOR FEATURE SELECTION

3.1. Introduction

FS is used as a dimensionality reduction technique in most of the problems of different research areas, where many attributes are available due to improved collection methods [12, 134]. For defining a single sample, a large set of features increases the computational complexity and the manifold space dimension. Out of these many features, most of the features are not informative or sometimes misleading, thus degrading the classifier's performance. In many other works, datasets are presented with missing values. The strategies used for selecting features aim to find the optimal subset of features and discard redundant information. Methods for the selection of optimal features, which are exhaustive, are NP-hard. So, to perform these tasks, intelligent frameworks are designed. According to their working principle, these frameworks are divided into filter, wrapper, and embedded methods [140]. Filter methods use intrinsic properties of data to evaluate the features and are efficient and computationally faster. Wrapper methods evaluate the features at the time of the learning mechanism. These are more accurate than the filter methods. However, these methods sometimes cause overfitting. The overfitting could be overcome by using different techniques like the hold-out method, cross-validation, L1/L2 regularization, etc. Embedded methods try to use the strength of these two methods. These methods are computationally costlier than filter methods and are not smart to handle high – dimensional data.

Classification problems are those that are used to identify new data belonging to which known class. Classification could be classified into two types: (a) binary classification and (b) multi-classification. In binary classification, the output classes in total are two, i.e.; the output could be either yes or no; for example, in the case of cancer, a person has cancer or not. In multi-classification for the problem, the output comes in k -classes,

where k is an integer ranging from 2 to n . Depending on the output, the classification problem could be linearly separable or linearly inseparable. Multiple methods have been devised to solve these problems. Some of them include K- Nearest Neighbour [141], Random Forest [142], Gradient Boosting [143], SVM [144], and Ada Boost [145].

The methods designed for feature selection are used in dealing with many challenges such as reducing the data dimensionality, computational complexity, computational cost, storage space, increasing the accuracy, ratio of features selected [146]. A method designed for feature selection cannot cover all the existing challenges in one go. So, most of the existing methods deal with only accuracy. In existing methods, data with fewer features have average accuracy. Increasing the number of features (high-dimensional dataset), the accuracy abruptly decreases. Moreover, most of the existing challenges as computational cost, storage space, ratios of features selected, high classification rate with fewer features are not covered by previous work and they remain challenges for today also.

To address the limitations of previous work and current challenges, the researcher has proposed a hybrid algorithm. This algorithm aims to overcome issues that were not optimally addressed in prior studies. It combines filter and wrapper techniques for feature selection, using Mutual Information alongside a Sequential Forward Selection method (a greedy approach), referred to as MISFS. The algorithm also evaluates the classification accuracy of multiclass datasets. The algorithm in the first phase applied the Mutual Information method on the dataset to remove those features that are highly independent compared to the target. After removing those features, in the second phase, the Sequential Forward Feature Selection method is applied to the remaining features to select the best features having optimized accuracy. Selecting filter methods in the first phase is that it has less computational time. Taking mutual information as a filter approach is that it removes redundant data easily. Also, it finds how relevant the target class is to the input features and also how relevant individual features are to each other. Selecting the Sequential Forward Selection method in the second phase as a wrapper approach involves less computational time and is robust against overfitting. The proposed algorithm MISFS covers a few existing challenges, such as maximum reduction of features in high-dimensional datasets, higher classification rate, and less

computational time, with the limitation of not addressing about the computational space and other complexities.

3.2. Background

3.2.1. *Dimensionality Reduction*: When we have selected the dataset for work, we have no idea about which features are essential concerning dependent features, which are noisy, irrelevant, or redundant. Most of the real-life application areas include bioinformatics, speech recognition, the oil industry, text processing, the Internet, engineering applications, medicine, diseases, and many more. The datasets covered in these areas are high-dimensional data with noisy, irrelevant, and redundant attributes. Thus, the FS process is applied to selecting those attributes that result [140] in minimum computing time with high accuracy. Achieving these results includes numerous factors, such as:

- Is there any methodology to find the optimal features?
- Is there any evaluation process that could say the selected features are optimal?
- Is there a methodology used for feature selection application independent?

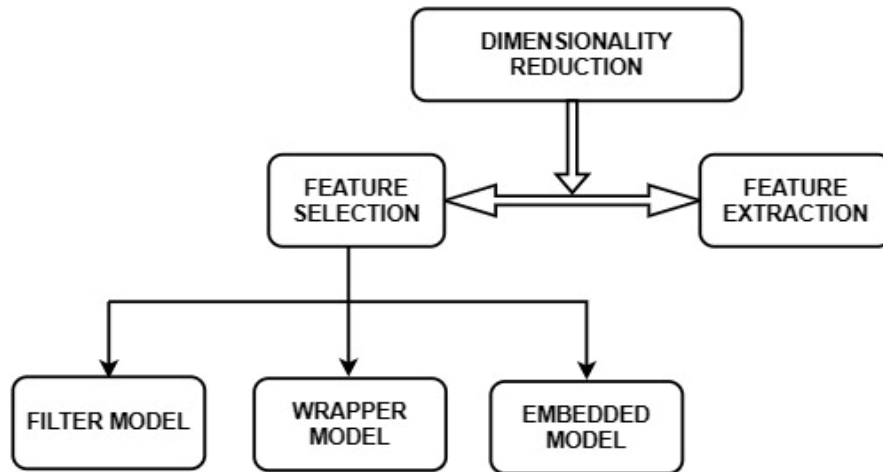


Figure 3.1. Dimensionality Reduction

Other motivations for performing feature selection or dimensionality reduction are general data reduction (minimum storage space with minimum execution time), simplifying the feature set (to save resources for subsequent data collection or utilization), performance improvement, and improving overall data understanding.

Figure 3.1 frames the dimensionality reduction. From research work [140], the general outline of the FS process, shown in Figure 3.2, could be categorized as follows:

- First, generate a feature subset using a specific methodology.
- After that, evaluate the feature subset.
- Then termination condition is applied and,
- Finally, the result is validated.

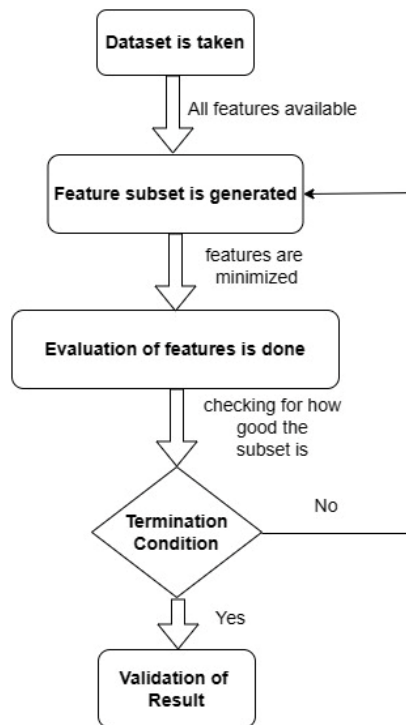


Figure 3.2. A general framework of the feature selection process

3.2.2. *Model for feature selection:* A different feature selection model includes a filter, wrapper, and embedded models [140]. Filter-based feature selection methods select the features using several scoring metrics in an unsupervised fashion, requiring no classifier in its core. Compared to the other two methods, they are computationally more efficient. But in terms of performance, wrapper methods are far better than filter methods. Wrapper methods select the optimal feature set by applying the learning algorithm at its core. The heart of this method is the classifier, whose purpose is to upsurge the performance of carefully chosen features at each iteration. The computational cost of the wrapper method is more compared to filter methods. Filter and wrapper methods differ mostly by evaluation criterion. Filter methods are sometimes acclimatized with feature

ranking methods. Similar to wrapper methods are embedded methods, using the inherent properties of the model as a metric during the learning phase. The more complex problems are handled easily via embedded methods in contrast with filter methods. However, in the case of larger training samples, embedded methods beat the filter methods. Hybrid methods also exist, which are a combination of filter and wrapper methods, first applying the filter method to rank features to generate a nested subset of features; these are computed via a learning mechanism, i.e., a wrapper approach.

Algorithms designed based on these methods are not perfect or suited for all real-life applications. Selecting an algorithm suitable for a given application area or all application areas is a tedious and complex task. The reason for this is that each application area is different from others. Moreover, the models or algorithms designed to solve a problem in a particular domain work differently depending on the data available. A small change in data could lead to different feature selections. As a result, an algorithm designed to solve a problem in one domain may not perform effectively in another domain. Also, every model could handle some of the challenges mentioned above. Most of the models available in the literature deal either with accuracy or the reduction in features. However, achieving both at the same time is not achieved at a higher rate by these algorithms. So, the purpose of this proposed methodology is to handle most of the challenges. It can be noticed that the MISFS method [147] achieves higher accuracy with with maximum reduction in features in both small and large data sets. Also, the computational efficiency is low compared to existing methods, reduce data dimensionality is achieved, and the ratio of features selected is also calculated. This approach's limitation is that it covers storage space to a very limited extent.

3.3. Related Work

The ability to acquire massive data in an existing period is a two-edged sword. Firstly, it allows us to analyze features more soundly due to the availability of more information about data. On the other hand, storing and processing this data is becoming more complex. Thus, techniques needed for dimensionality reductions are becoming more important because they help to remove features that are less needed without decreasing

the learner's performance. Features that fail to differentiate samples from different classes or clusters are considered irrelevant. The learning performance of the system is not hampered after removing such features. However, most of the time, the system's performance is improved if a suitable model is used for performing feature selection depending on the area and type of problem selected.

Most of the research has been done, and many overview papers have been designed to evaluate feature selection methods available in the literature [12, 135] in detail with their categorization. Major studies are in the area of classification problems, which include both binary and multi-class problems. Filter methods are used for counting the number of samples misclassified on two gene expression datasets [148]. Classification accuracy is calculated on artificial datasets [149] using different filters, wrappers, and embedded methods. Again, the microarray datasets are considered by [150] for calculating the classification accuracy using filter methods. Some reviewed the ensemble feature selection techniques for classification. In [151], the author used various strategies, such as considering sentiment data for classification and using ensemble feature selection with hesitant fuzzy sets to attain results. They selected the top k features based on a relevancy score. The existing algorithms do not address all the challenges. Mostly these methods cover only accuracy as a parameter. The weakness of these approaches is that they do not cover the number of features selected, computational time, and storage space. Additionally, they are often tuned to specific feature selection methods, and low accuracy in high dimensional datasets.

In the case of diseases, there are common symptoms for some diseases. Therefore, diagnosing them is sometimes a bit complex, so various optimization techniques are required to overcome this problem. Research has also been done on diseases caused by diabetes. The deep learning technique [152] is used to predict eye, kidney, and cardiovascular diseases caused by diabetes. They have used statistical methods to evaluate the performance. Most research has been done on the widely used PIMA Dataset available at the UCI repository. One of the research works in [153] predicted diabetes by employing a classification model based on the Synthetic Minority Oversampling Technique and DT, thus improving the accuracy of the estimation of diabetes by eliminating class imbalance. They achieved a classification rate of 94.70% on the dataset considered. However, these existing research works have weaknesses in

that they do not handle the existing issues such as reduction in data dimensionality, computational cost, and higher classification accuracy but somehow achieved the accuracy at par level. Most of the methods do not outline their weaknesses.

Another in [154] implemented Principal Component Analysis and Particle Swarm Optimization for feature selection in multiple combinations with classification algorithms on PIMA Indian Diabetes set and the Local dataset of diabetes (LDD) gathered from Ranchi, India and achieved the highest classification rate of 79.56% with PCA – LR combination for the PIMA and 95.58% on LDD with PCA – PSO – C4.5 DT. This showed that the efficiency of the proposed model shows different performances on two datasets. Moreover, there was a 50% reduction in features in both datasets via PCA and PSO. However, the authors remain silent on computational time, and storage space and do not cover the high dimensional data to evaluate the performance of the proposed method. Also, they do not guarantee that the same performance and the same percentage of reduction in features will be achieved on other datasets. A similar task was performed by [155] while calculating the accuracy of the PIMA Indian Diabetes Dataset. They compared the performance of the Support Vector Machine, ANN, Naïve Bayesian, J48 DT, and Bagging method and also added the performance of the Genetic Algorithm with SVM. Also, they analyzed the weakness and strengths of these methods. They concluded that statistical methods are not successful on categorical data and handling missing values, that's why machine learning methods are used. In [156], they used SOM optimization algorithm with four heuristic algorithms namely Particle Swarm Intelligence (PSO), newton-based self-organizing map PSO, self-organizing map Harmony Search Algorithm and self-organizing map Swarm. They implemented it on various bio-medical datasets. They compared the performance of these on PIMA Indian Diabetes Dataset, Appendicitis, Heart, Hepatitis, Mammographic, Wisconsin Breast Cancer and new Thyroid dataset. They achieved 91% accuracy on new Thyroid dataset using newton-based self-organizing map PSO and 97% accuracy on Wisconsin dataset through self-organizing map Harmony Search Algorithm. The method proves good quantization error for clustering and good accuracy with statistical measures. The authors proposed that multi-strategy SOM deep mapping learning could be adopted for multi-dimensional

unstructured big-data problems. However, they have an issue in dealing with big-data preprocessing.

Researchers have proposed various feature selection methodologies designed on the Mutual Information (MI) technique. It is an approach based on the filter method. The first classification approach using MI dates back to [157]. In that study, the authors used a Mutual Information-based feature selection method for text classification. The degree of correlation between the independent feature and dependent feature is calculated with Mutual Information and also helps in filtering out the irrelevant features. This method does not consider the relationship between the candidate and selected features. Moreover, in the case of heavily redundant data, the performance of Mutual Information is degraded. The issue of not considering the relationship between the candidate and selected features is considered in [158]. They proposed MI based on FS (MIFS) which evaluates the availability of unnecessary information between potential features and the selected ones.

The patient's records are maintained through biomedical datasets containing multiple features. Several methods based on the wrapper and hybrid filter wrapper are used as ideal methods. Hybrid methods built on Information Gain Genetic Algorithm [159] and Information Gain Particle Swarm Intelligence [160] showed the classification accuracy of combined filter and wrapper methods. In [159] they concluded that no classification technique outperforms all the other existing methods. They also concluded that Genetic Algorithm somewhat improves the classification performance of other classifiers. Artificial Bee Colony algorithm [161] used only seven features on the Cleveland heart disease dataset with a better classification accuracy compared to other feature selection methods. Feature selection is made by Artificial Bee Colony methodology, and the performance is measured using the KNN classifier, whose performance is excellent on Cleveland Heart Disease dataset [162]. There are a few other domains in which LSTM networks have been successfully applied in sequence classification. For example [163], used LSTM network for air-pollution outlook. The authors used data from air quality sensors for the same. They clean the raw data and fill in the missing data using edge-based components exploiting temporal and spatial information. The predicted accuracy was improved by 40.18% on the mean absolute percentage error.

This section on previous work shows that various feature selection methods tailored for classification tasks on different diseases. Most of the methods deal with classification accuracy and some deal with feature selection on different datasets and then calculated the accuracy. But it is noticed that most of the methods do not provide higher accuracy with maximum feature reduction. Also, most of the existing challenges, explained above, are not handled by the existing methods. This research has proposed a methodology MISFS which identifies the issues in the existing methods and tries to find the solution to most of the existing challenges. The proposed method uses the hybrid filter - wrapper method to cope with the issues mentioned in the literature. This combination takes benefit of both filter and wrapper methods, thus improving the classification accuracy, reduced feature set and low computational time, and low storage space. Mutual information (MI) is used as a filter method for ranking the features and then selecting the features with higher ranking from all features set, passing the higher ranked features to Sequential Forward Feature Selection (SFS) – a greedy-based wrapper method - for a further selection of features and calculating the accuracy. Thus, maximum features of less importance, redundant and noisy features are removed with achieving higher accuracy. As the approach is greedy based, thus decreasing the computational complexity as a whole. Thus, the proposed MISFS method covers existing challenges regarding data dimensionality, computational complexity, computational cost, storage space at some extent, model accuracy, ratio of features selected.

The proposed MISFS method covers the application area of the medical field but it could be used for different application areas.

3.4. Research Methodology

3.4.1. MISFS Approach

Here the authors have proposed an approach, MISFS, for selecting optimized features to obtain a good classification rate. A hybrid approach is designed for selecting optimized features. In this approach the strengths of both the filter and wrapper approaches by exploiting their diversified evaluation capabilities in both methodology phases. The process leveraged the advantages of both filter and wrapper methods. The filter approach, which considers only the data for evaluation, is algorithm-independent.

The wrapper approach considers both the data and the learning algorithm for evaluation and is algorithm-dependent. Separately, both have some disadvantages, which are overcome by the hybrid approach we propose.

3.4.2. Abstract View of MISFS and Comparison Methods

The researcher proposed a methodology called MISFS for feature selection and calculating accuracy. First of all, the dataset is divided into the training data and the test data. After that, MISFS is implemented on the training set to identify the optimal features. After finding the optimal features, both the training and testing accuracy are calculated using MISFS with KNN and LR. Finally, the results of the MISFS are compared with three different styles as shown in Figure 3.3.

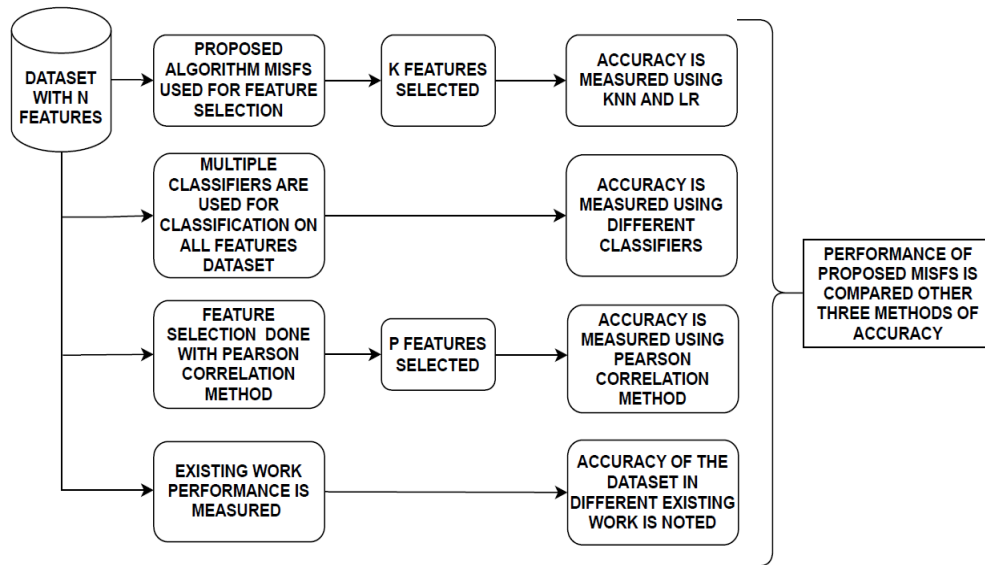


Figure 3.3. Abstract view of MISFS working and result comparison

3.4.3. Phases in MISFS

The approach MISFS designed for feature selection is a hybrid approach, as explained in Section 3.4.1, and is implemented in two phases viz. Phase 1, and Phase 2. The summary of each phase is as follows:

- a. First phase: The filter approach is implemented in the first phase, and in the second phase, we implemented the wrapper approach. The fundamental task is to calculate the improved classification rate of the datasets after feature selection. So, in the first

phase, implemented Mutual Information, which is a filter approach, for feature selection. Through this approach, the selection of features highly correlated to the output class is done. After that, by selecting these features, we implemented our second phase.

- b. Second phase: The sequential forward feature selection (SFS) method, a wrapper model, is implemented for feature selection. This methodology is based on the Greedy approach. Here, the classifier first calculates the performance of each feature individually. After that, in the next step, it selects the feature with the highest performance, pairs it with the remaining features, and evaluates the classifier's performance. For understanding, let's say there are four features f_1 , f_2 , f_3 , and f_4 . In the first step, the individual performance of each feature is evaluated. Let f_2 be the feature with the highest performance. In the next step, we start from f_2 and pair it with the remaining features (pair will be $\{f_2 f_1\}$, $\{f_2 f_3\}$, $\{f_2 f_4\}$) and evaluate their performance. Then we again start with the highest performance pair and repeat the same process by adding the remaining features. Finally, we get the optimal feature set with maximum performance. KNN classifier is used with SFS while evaluating the performance and selecting the optimal features.

3.4.4. Method and Algorithm

This sub-section covers the implementation details of the MISFS algorithm phase-wise. Also, an algorithm is designed to implement the MISFS approach completely.

Phase I:

In the first step, implemented a Mutual Information-based filter approach for ranking all features. The method used here considers the correlation among different features and the correlation among the target and features for ranking. The highly correlated features are ranked higher, and features with less correlation are ranked low. MI detects non-linear relationships between variables, as an example, for the popular correlation coefficient. MI is also defined for multidimensional variables, taking joint relevancy and redundancy of features into account while selecting features. If A and B are independent variables, then there will be no MI between them; it will be zero. This is because A contains no information about B and vice versa. The following concept is used for implementing mutual information.

- i. We used the concept of probability density function to calculate mutual information between A and B, defined as

$$I(A; B) = H(A) - H(A|B) = H(B) - H(B|A) = H(A) + H(B) - H(A; B) \quad (3.1)$$

where,

- $H(.)$ denotes entropy;
 - $H(A|B)$ and $H(B|A)$ are conditional entropies;
 - $H(A; B)$ is the joint entropy of A and B
- ii. Joint entropy $H(A; B)$ is calculated as

$$H(A) = - \int_a p_A(a) \log p_A(a) da \quad (3.2)$$

$$H(B) = - \int_b p_B(b) \log p_B(b) db \quad (3.3)$$

$$H(A; B) = - \int_a \int_b p_{A,B}(a, b) \log p_{A,B}(a, b) da db \quad (3.4)$$

where,

- $\log p_{A,B}(a, b)$ is the joint probability density function;
 - $p_A(a)$ and $p_B(b)$ are the marginal density functions of A and B, respectively.
- iii. The marginal density functions are

$$p_A(a) = \int_b p_{A,B}(a, b) db \quad (3.5)$$

$$p_B(b) = \int_a p_{A,B}(a, b) da \quad (3.6)$$

- iv. By substituting equations (2) – (4) into Equation (1), the mutual information equation will be

$$I(A; B) = \int_a \int_b \frac{p_{A,B}(a, b) \log p_{A,B}(a, b)}{p_A(a)p_B(b)} dadb \quad (3.7)$$

In discrete forms, the integration is substituted by summation over all possible values that appear in data.

Phase 2:

After selecting the features according to their ranking with MI, in the next phase, the Sequential feature selection method based on wrapper methodology for selecting the best features from the remaining ones is applied. There are multiple ways to implement Sequential Feature Selection, among which the SFS method was employed. It uses a greedy approach that reduces the primary p-dimensional feature set to an m-

dimensional feature subset. The property of SFS is that it inevitably selects the most relevant subset of features according to the problem. Using this methodology, the computational efficiency of the model is improved and removed the irrelevant as well as noisy features, thus reducing the model generalization error.

Algorithm 1 covers the MISFS approach for feature selection. The algorithm, in total, is divided into 13 steps. Steps 1 – 7 cover Phase 1, and the remaining steps cover Phase 2. Line 3 – 6 implements the MI for ranking the features and passes those features in line 7 for further implementation and evaluation for phase 2. The initialization for phase 2 is done in line 8, and further implementation (inclusion) is done from lines 9 – 11 and repeated until obtaining the subset of features with maximum classifier performance. Termination criteria are covered in line 12. Step 13 shows the implementation of cross-validation in phase 2.

Algorithm 1: MISFS model	
Input: d , total features available in repository P	
$P = p_1, p_2, p_3, \dots, p_d$	
Result: Best minimum features X_m selected after applying MISFS.	
Phase 1: Mutual Information (MI)	
1.	d , total features available in dataset P
	$P = p_1, p_2, p_3, \dots, p_d$
2.	apply Mutual Information (MI) for ranking the features as in step
3.	for <i>each</i> d do
4.	compute MI (p_i, C)
5.	Rank each feature p_i & arrange them in descending order
6.	end for
7.	Select top k features based on ranking for phase II
	$P' = \{p_1, p_2, p_3, \dots, p_k\}$
Phase 2: Sequential Forward Feature Selection (SFS)	
Input: k features selected after MI (from step 7)	
	$P' = \{p_1, p_2, p_3, \dots, p_k\}$
Output:	
	$X_m = \{x_i i = 1, 2, \dots, m; x_i \in P', \text{ where } m = (0, 1, 2, \dots, k)\}$
Through SFS we selected the best m features where $m < k$, and it is a subset of features in P' .	
8.	Initialization
	$X_0 = \emptyset, m = 0$
At the initial stage of SFS, we consider an empty set $\emptyset$, so the $m = 0$ (where m is the subset of features from k features).	
Inclusion:	
9.	$x = \arg \max S(X_m + x_i), \text{ where } x_i \in P' - X_m$
10.	$X_{m+1} = X_m + x$
11.	$m = m + 1$
	goes again to step 9
12.	Termination:
	$m = p$
13.	Cross-validation and KNN used with SFS
	Cross-validation is used while working in phase 2.

Algorithm 1 covers the MISFS model for feature selection and measuring classification accuracy. In this algorithm, in line 4, C stands for computation, and steps 9 – 11 are inclusion step. In this step, another attribute x is included in the attribute subset X_k . x is the attribute that capitalizes the performance of the criterion function. It means that if adding x in X_m increases the classifier's performance, then it is included; otherwise, it is excluded. This procedure is repeated till all the features are exhausted and then algorithm satisfied the termination criterion. Step 12 is the termination step. In this step, the algorithm adds features from the feature subset P' until the feature subset of size m contains the number of desired features p that we specified a priori. The proposed algorithm finds the m best features, which maximize the classifier performance. Step 13 covers the cross-validation in Phase 2. While working in Phase 2, 10-fold cross-validation with a Sequential Forward Feature selection method along with k -NN as a classifier to avail the optimized feature selection. Cross-validation also helps overcome overfitting and increases the prediction model's performance estimation. The computational time of the algorithm is $\min(f(nm))$, where f is the selected features, n denotes the number of features and m denotes the number of samples.

3.4.5. Workflow of the MISFS

The workflow of MISFS is covered in this section using diagrams. Figure 3.4 explains the flow of the methodology, and Figure 3.5 explains how cross-validation is implemented in MISFS, as covered in section 3.4.4 and Algorithm 1.

Figure 3.4 is partitioned into six steps. First step, consider the original dataset for implementation. Next, splitted the dataset in 70:30 ratio to training and testing data. In the third step, implemented MI on training data for feature ranking and selected the top k features for further analysis using SFS, in step 4. Step 5 implements SFS on the features selected after MI to select the best features with higher accuracy. And finally, the optimal feature subset could be obtained. Then, the model is implemented on testing data to calculate accuracy using KNN and LR.

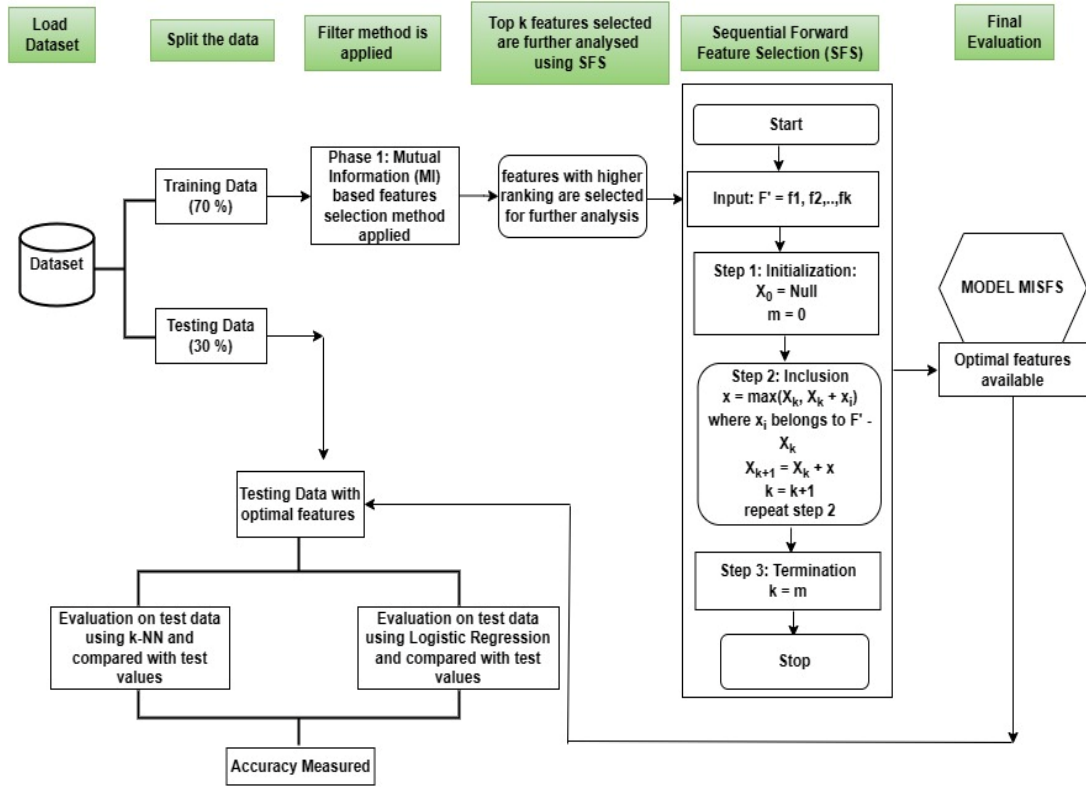


Figure 3.4. The flow of the proposed methodology

Figure 3.5 displays how cross-validation is implemented in SFS. Cross-validation is implemented in Step 2 of SFS on the training dataset for hyperparameter tuning and feature selection. The training set is divided into the 'n/k' validation fold and the remaining training fold in K-fold cross-validation. A total of k experiments is performed, with shifting validation fold in each experiment and calculating the performance at each experiment.

After that, the average performance is calculated, and optimal features are provided. After receiving the optimal features, the training accuracy using KNN and LR is evaluated.

3.5. Experimental Evaluation and Comparative Analysis

3.5.1. Experimental Setup

The methodology is executed on multiple datasets to see the performance of the proposed algorithm. The motive of the approach is to select the minimum number of features with moderate accuracy. To achieve this, the dataset is divided into training

and testing datasets. After that, used MI to find features highly correlated with the output class. After selecting the features with high MI, the model implemented the SFS approach on the remaining features to obtain the features with moderate accuracy. The methodology used the model KNN with 10-fold cross-validation for SFS. An algorithm is covered in section 3.4.4.

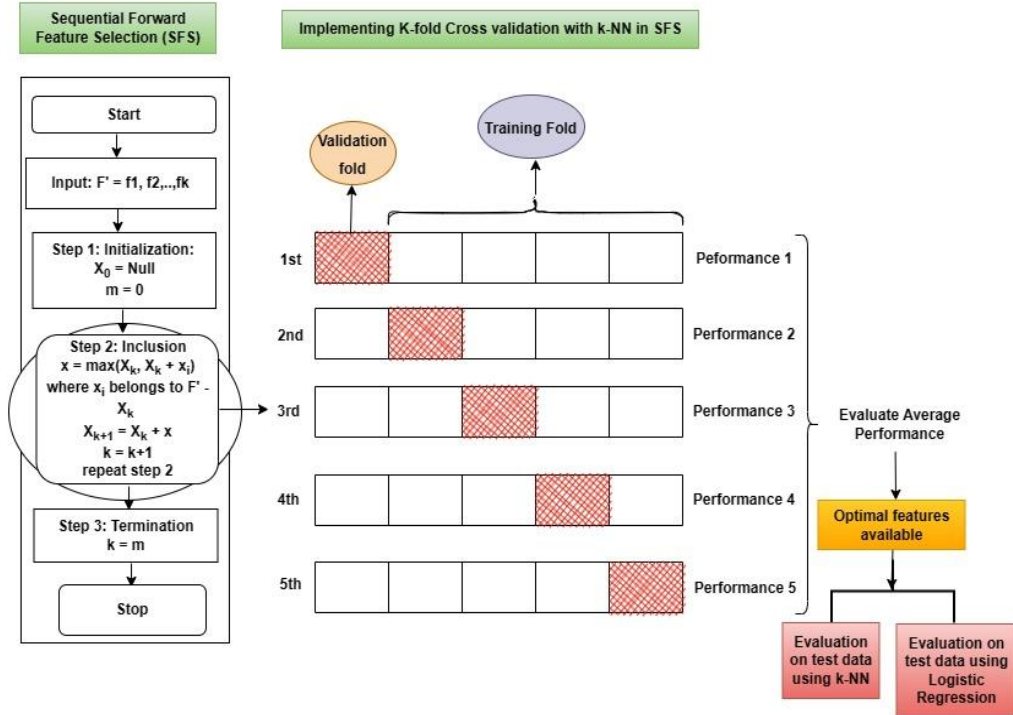


Figure 3.5. K-fold cross-validation in Sequential Forward Feature Selection

3.5.1.1. Datasets Used

The following datasets have been chosen to evaluate the performance of the proposed method in terms of the number of features selected and its efficiency. A few of them are from the UCI repository, and others are being taken from different sources. The datasets that are considered are PIMA Indian diabetes, Spectfheart, Breast Cancer, Parkinson, PCOS, and Cleveland. Table 3.1 is used to show the necessary information regarding the above data samples. Among these datasets, Spectfheart and PCOS have a large number of features, 45. One dataset is considered, which is multiclass with 5 classes.

- i. **PIMA Indian Diabetes:** It has 768 samples from female patients of Pima Indian tradition, with 8 numeric valued features and output as a binary class. This dataset is used to diagnose type 2 diabetes. This dataset contains information on 21 years

or older women and is of non-linear type. The dataset includes two classes: Class 1 is normal with 500 samples, and Class 2 is patients with Pima Indian Diabetes with 268 samples. The eight features are pregnancies, glucose, BP, BMI, skin thickness, age, insulin, and diabetes pedigree function.

- ii. Spectfheart: It contains 267 samples diagnosing Cardiac Single Proton Emission Computed Tomography (SPECT) images. It consists of 44 features, and normal and abnormal are the classes for classifying patients. The details of 44 features could be examined from the UCI repository.

Table 3.1. Overview of Data Samples

Dataset	#Samples	#Attributes	#Classes	Variable Description
PIMA	768	9	2	768 x 9
Spectfheart	267	45	2	267 x 45
Breast Cancer	569	33	2	569 x 33
Parkinson	195	24	2	195 x 24
PCOS	541	45	2	541 x 45
Cleveland	297	13	5	297 x 13

- iii. Breast Cancer: It comprises 569 samples of heart disease patients, having 33 features and output as a binary class. Output is a challenge to classify whether each person's tumor lies malignant or benign. The 33 feature details are given in the UCI repository.
- iv. Parkinson dataset: It comprises 195 samples of heart disease patients, having 24 features and output as a binary class. Output is the "status" feature, representing 0 for healthy and 1 for Parkinson's disease. The details of features are given in the UCI repository.
- v. PCOS dataset: It gathered information from 10 different hospitals. It contains 541 samples of PCOS patients, with 45 features and output as a binary class. Output is whether the male person is infected with PCOS or not. The 45 feature details are given in the UCI repository.
- vi. Cleveland dataset: It comprises 297 samples of heart disease patients, with 13 integer-valued features and output as 5 classes from 0 to 4. The output is to see that the patient has heart disease, and 0 to 4 signifies the absence and distinguished occurrence of disease. The 13 features are age, sex, cp, tpb, sc, fbs, resting

electrocardiographic results, mhr received, exercise-induced angina, op, the peak exercise slope, and major vessels amount, and thal.

3.5.1.2. Performance Metrics

To calculate the performance of these datasets on the MISFS, we have considered the confusion metrics, classification rate, recall, specificity, precision, and f-measure as evaluation metrics [140]. Briefings of these evaluation metrics are shown below:

- **Confusion Matrix:** This matrix is used to measure how many samples are perfectly classified and how many samples are misclassified. Four terms are considered in the confusion matrix.
 - a. **True positive (TPOS):** Samples that are labeled positive are predicted to be positive.
 - b. **True Negative (TNEG):** Samples that are labeled negative are predicted to be negative.
 - c. **False Positive (FPOS):** Samples that are labeled negative are misclassified as positive.
 - d. **False Negative (FNEG):** Samples that are labeled positive are misclassified as negative.
- **Accuracy or Classification Rate:** Calculates the total ratio of classes predicted properly by the classifier.

$$Accuracy = \frac{TPOS + TNEG}{TPOS + FPOS + TNEG + FNEG} \quad (3.1)$$

- **Sensitivity (or Recall):** It calculates what ratio of class labels that are predicted as positive belong to positive.

$$Sensitivity = \frac{TPOS}{(TPOS + FNEG)} \quad (3.2)$$

- **Specificity:** It calculates what ratio of class labels that are predicted as negative belongs to negative labels.

$$Specificity = \frac{TNEG}{(TNEG + FPOS)} \quad (3.3)$$

- **Precision:** It demonstrates the number of predicted classes on all positively classified labels.

$$\text{Precision} = \frac{TPOS}{(TPOS + FPOS)} \quad (3.4)$$

- F-measure: Precision and recall are inversely proportional to each other; increasing one usually decreases the others. This inverse relationship between precision and recall is called F-measure. It is calculated using HM of precision and recall and considering both measures. It is computed as follows:

$$\text{F-measure} = 2 * \frac{\text{Recall} \times \text{Precision}}{(\text{Recall} + \text{Precision})} \quad (3.5)$$

3.5.1.3. Comparison Techniques

After designing the algorithm, it was used to derive the optimized feature subset on the datasets taken for evaluation and also calculated accuracy including other metrics on the algorithm using K-nearest neighbor and Logistic Regression.

The results of MISFS are compared in three ways:

- At the first level, calculates the accuracy of the datasets considered in this work with different classifiers, taking into account all the features, and comparing with the proposed algorithm.
- At the second level, implement the Pearson correlation method to select the optimal features and then apply dissimilar classifiers considered in point (a) for calculating the accuracy and comparing it with the proposed algorithm.
- At the last level, compared the performance of the MISFS with the existing methods.

3.5.2. Results and Discussion

The performance of the MISFS algorithm is evaluated on the above-mentioned datasets. The experimental results are shown here:

3.5.2.1. Features Selected by MISFS

On applying the MISFS on the datasets mentioned above, the optimized features selected on each dataset are shown in Table 3.2. It shows that the feature reduction of larger datasets is comparatively more compared to smaller feature sets. In Pima Indian diabetes set, there were 9 features, and after applying the proposed algorithm, it was

reduced to 4. Parkinson dataset was reduced from 24 to 12, and the Cleveland dataset was reduced to 6 out of 14 after applying the MISFS algorithm.

Table 3.2. Optimized Features Selected

Dataset	Total Features including output class	Features after applying the MISFS algorithm
PIMA	9	4
Spectfheart	45	6
Breast Cancer	33	12
Parkinson	24	12
PCOS	45	15
Cleveland	14	6

This shows that on smaller features, the reduction is about 50%. While working on other datasets which have larger feature sets, the reduction was more than 75%. The Breast Cancer dataset with 33 features was reduced to 12, while the PCOS dataset with 45 features was reduced to 15 with the proposed algorithm, with percentage reductions up to 36.36% and 33.33%, respectively. In these datasets, the reduction in features was near to or greater than 65%. Considering the Spectfheart dataset, the features were reduced from 45 to 6. That is, only 13.33% of features remained, showing more than an 85% reduction.

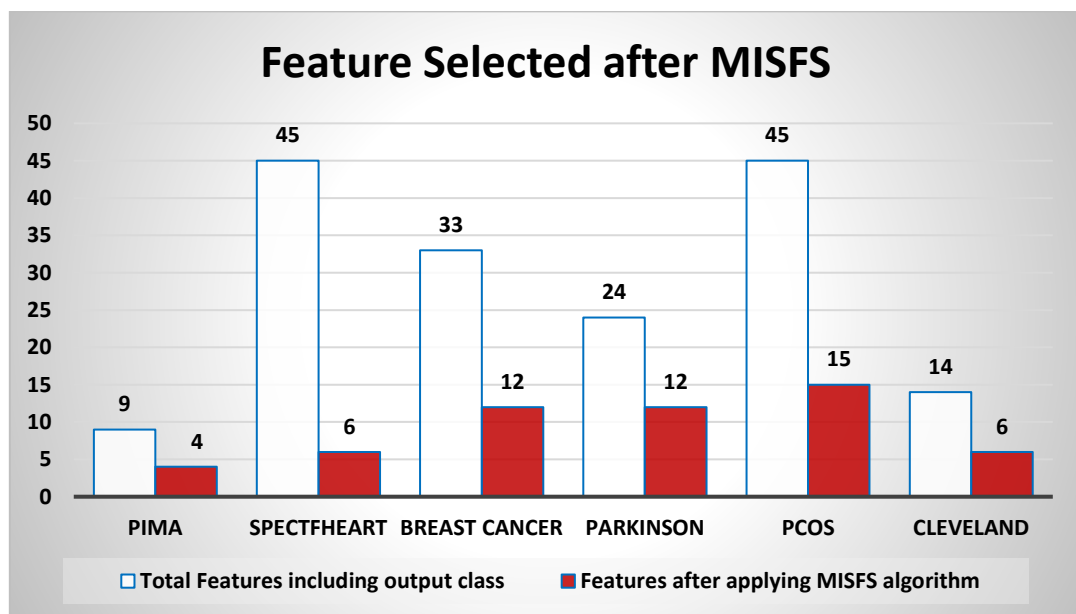


Figure 3.6. Features selected by the proposed algorithm MISFS on each dataset

Figure 3.6 shows a bar chart of features reduced on each dataset after applying the algorithm MISFS. Thus, it covers the general data reduction, limiting the storage requirement to some extent and increasing the algorithm speed, which could be strongly proven after achieving the accuracy. It can be seen from Figure 3.6 that on applying the algorithm, the total features obtained are minimized, and it is seen that in datasets with a large number of features (Spectfheart and PCOS), the performance of the algorithm is comparatively better as compared to smaller feature datasets.

3.5.2.2. Performance on different datasets

This section presents the results of the proposed model MISFS on real-world datasets. Each dataset result on the proposed algorithm is shown in Table 3.3.

Table 3.3. Performance of Algorithm MISFS

Dataset	Proposed Method	Accuracy	Precision	Recall	F1-score	Support
Pima	MISFS+KNN	77.48	0.8098	0.8400	0.8250	231
	MISFS+LR	75.75	0.7853	0.8854	0.8323	
Spectfheart	MISFS+KNN	76.540	0.5	0.6316	0.5581	81
	MISFS+LR	79.012	0.6	0.3158	0.4138	
Breast Cancer	MISFS+KNN	94.73	0.9626	0.9537	0.9581	171
	MISFS+LR	97.07	0.97	0.97	0.97	
Parkinson	MISFS+KNN	94.91	0.95	0.95	0.95	59
	MISFS+LR	89.83	0.89	0.90	.90	
PCOS	MISFS+KNN	84.66	0.85	0.85	0.84	163
	MISFS+LR	87.11	0.87	.87	.87	
Cleveland	MISFS+KNN	52.22	0.54	0.59	0.59	90
	MISFS+LR	58.88	0.54	0.59	0.59	90

The result of the proposed algorithm MISFS on different datasets shows that the highest accuracy achieved by the algorithm is 97.07% on the Breast cancer dataset by using the classifier Logistic Regression. (LR). Table 3.3 shows the accuracy of each dataset with MISFS using LR and KNN. It also shows the precision, recall, and f1-score achieved on each dataset. The accuracy achieved by most of the datasets is above 84% except for Spectfheart, Pima Indian diabetes, and Cleveland datasets. Figure 3.7 shows the bar graph of accuracy achieved on each dataset using MISFS.

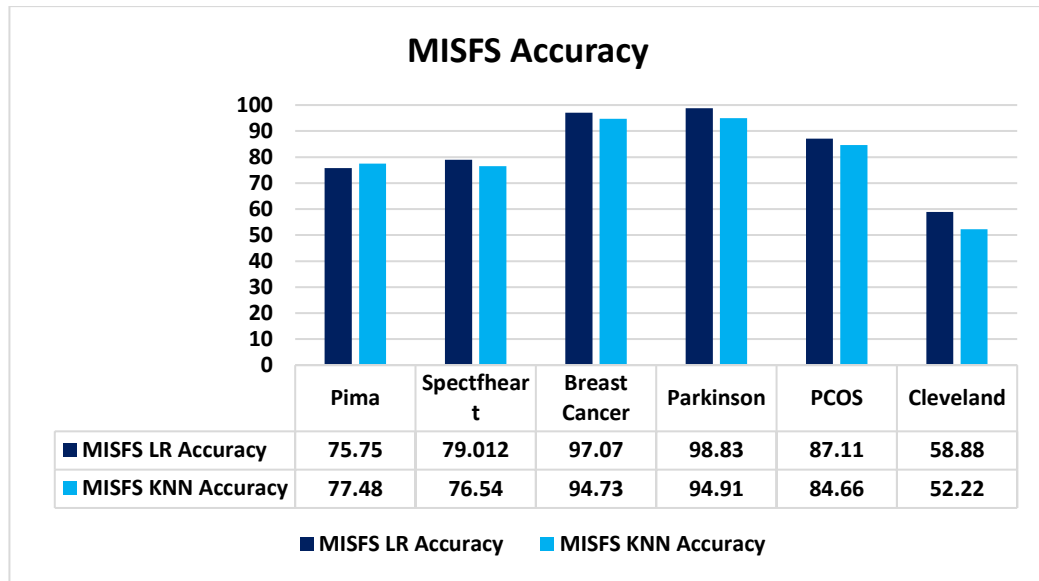


Figure 3.7. Accuracy of each dataset on the algorithm MISFS with classifier LR and KNN

3.5.2.3. Comparative study of MISFS with LR, KNN, and PCC

The comparison is implemented among the highest accuracy achieved by the proposed algorithm MISFS with LR, KNN, PCC-LR, and PCC-KNN shown in Table 3.4.

Table 3.4. Comparison of performance of proposed algorithm MISFS with LR, KNN, and PCC

Dataset	Result	Methodology				
		MISFS	LR	KNN	PCC+KNN	PCC+LR
Pima	FS	4	9	9	9	9
	Accuracy	77.48	72.39	65.62	73.95	72.39
Spectfheart	FS	6	45	45	45	All
	Accuracy	79.012	72.89	67.90	77.61	85.07
Breast Cancer	FS	12	33	33	28	28
	Accuracy	97.07	96.50	96.50	96.50	94.40
Parkinson	FS	12	24	24	13	13
	Accuracy	94.91	83.67	89.79	85.71	89.79
PCOS	FS	15	45	45	38	38
	Accuracy	87.11	86.02	67.64	89.70	85.29
Cleveland	FS	7	14	14	All	All
	Accuracy	58.88	58.66	48.89	56	56

Comparing the performance of MISFS with LR, KNN, and PCC with both LR and KNN performance on the Pima dataset, it is noticed that MISFS got the maximum accuracy of 77.48% with the Logistic Regression classifier, which is larger than the accuracy achieved by LR, KNN, and PCC with both KNN and LR. The accuracy achieved by MISFS is higher on all datasets, as shown in Table 3.4, except on PCC with LR on Spectfheart and PCC with KNN on PCOS. Evaluating the results of the Spectfheart dataset with the MISFS, got the highest classification accuracy of 79.012%. Evaluating the performance of Spectfheart dataset having all features, with classifiers mentioned, received the accuracy of 67.90% with K Nearest Neighbours and 72.89% with LR, which is much lesser than received by the proposed algorithm. The reason could be that the proposed algorithm tried to find the best feature selection, and outliers are removed using mutual information, so the proposed algorithm's performance is improved. Also, the proposed algorithm's precision, recall, and f1-score are average. Figure 3.8 shows the comparison of each dataset with the LR and KNN accuracy with all features and also the comparison of PCC for feature selection with LR and KNN separately.

With the MISFS algorithm, the Breast Cancer dataset received 98.23% training accuracy with KNN, 94.73% testing accuracy with the KNN classifier, and 97.07% testing accuracy with LR. The precision, recall, and f1-score value is also above 0.95 for both the approaches with the proposed methodology. It showed that the algorithm proposed is performing soundly on the dataset. Evaluating the performance of these classifiers with all features received an accuracy of 96.05% with both KNN and LR, which is smaller than the accuracy received by MISFS. Again, applying PCC for selecting features and then applying these classifiers received an accuracy of 96.50% and 94.40% with KNN and LR, respectively, which is again smaller than MISFS. The Parkinson dataset received a training accuracy of 95.58% (not shown in Table 4) with KNN and almost similar testing accuracy of 94.91% with KNN. When comparing the result of MISFS with the dataset with all features, its performance was 89.79% with KNN and 83.67% with LR, which is low compared to the performance we received with MISFS. Using PCC for feature selection with the threshold value of 0.5 to 0.95 we received an accuracy of 85.71% with KNN and 89.79% with LR, which is less than achieved by the proposed MISFS algorithm.



Figure 3.8. Comparison of MISFS accuracy with accuracy achieved with all features and PCC on datasets considered

Similarly, the PCOS dataset received higher accuracy with MISFS when compared to its performance with LR, KNN, and PCC-LR. In contrast, PCC-KNN has higher accuracy compared to MISFS. Similar is the case with the Cleveland dataset.

3.5.2.4. Comparison of the algorithm MISFS with the existing work

In this section, we compared the accuracy of the proposed algorithm MISFS on each dataset with the accuracy of existing work separately from Table 3.5 to Table 3.10.

Table 3.5. Comparing MISFS with existing work on Pima Indian diabetes

Pima Indian Diabetes Dataset			
Authors	Techniques	Accuracy	Feature Selected
[154]	PCA_C4.5 DT, PCA_KNN	74.78	4
[164]	Naïve Bayes, KNN, DT, Extra Trees	72.4137	-
[165]	KNN, SVM	75	3
[166]	SVM, DT	75.06	
MISFS	LR, KNN	77.48	4

Table 3.6. Comparing MISFS with existing work on the Spectfheart

Spectfheart Dataset			
Authors	Techniques	Accuracy	Feature Selected
[167]	Adaboost with Threshold Classification (TC)	70.59	-
	Multi (TC)	75.40	
[168]	ReliefF with new relief- feature weighting objective function	76.86	-
[169]	RFE with SVM	77.1368	14
[170]	Grey Wolf Optimizer with KNN	79.40	23
MISFS	LR, KNN	79.012	6

Table 3.7. Comparing MISFS with existing work on the Breast Cancer dataset

Breast Cancer Dataset			
Authors	Techniques	Accuracy	Feature Selected
[171]	Chi-square, SVM, Random Forest, Naïve Bayes, Logistic Regression	97.0	17
[172]	Gradient Boosting	96.77	-
[173]	LR with area under the curve	96	-
[174]	SVM	97.06	-
[170]	Grey Wolf Optimizer with KNN	94.60	3
MISFS	LR, KNN	97.07	12

Table 3.8. Comparing MISFS with previous work on the Parkinson Dataset

Parkinson Dataset			
Authors	Techniques	Accuracy	Feature Selected
[175]	ICA + DWT + LSVM	84.5	-
[176]	PCM with SVM	89	-
[177]	Extra Tree, MI Gain, GA with Naïve Bayes, K-Nearest Neighbour, RF	95.58	11
[178]	Recursive Feature Elimination, Feature Importance, SVM, ANN, CART	93.84	-
MISFS	LR, KNN	97.07	12

Table 3.9. Comparing MISFS with existing work on the PCOS dataset

PCOS Dataset			
Authors	Techniques	Accuracy	Feature Selected
[179]	Red Deer Algorithm with RF	89.81	20
[180]	LR with 5-fold cross-validation	85.022	13
[181]	KNN, SVM	90.83	10
[180]	LR	83	14
MISFS	LR, KNN	87.11	15

Table 3.10. Comparing MISFS with existing work on the Cleveland Dataset

Cleveland Dataset			
Authors	Techniques	Accuracy	Feature Selected
[182]	FS + R	54.54	-
[183]	FURIA	56.57	-
[184]	FARC	57.92	-
MISFS	LR, KNN	58.88	6

Comparing the accuracy of the existing work with the proposed work on the Pima Indian diabetes dataset, Spectfheart dataset, and Breast Cancer dataset, as shown in Table 3.5 – 3.7, it is noticed that the classification rate of the MISFS is better and almost equal to all the other existing methods. Moreover, the features selected via MISFS on the Spectfheart dataset are much smaller than existing methods with similar or larger accuracy. When talking about the Breast Cancer dataset, the number of features selected by MISFS is smaller than a few of the existing methods and larger than a few of the existing methods with similar or larger accuracy. Comparing the accuracy of the

existing work with the proposed work on the Parkinson dataset, it is noticed that the classification rate of the MISFS is better than the other existing methods. Also, the number of features selected by MISFS is almost similar to existing methods with higher accuracy. In the PCOS dataset, as shown in Table 3.9, it is noticed that MISFS accuracy is better with few existing methods, but on the same side, few existing methods have higher accuracy compared to MISFS. Also, the features selected are almost similar to other existing methods. Comparing the performance of the Cleveland dataset, it is noticed that the proposed algorithm MISFS performance is better than existing methods by approximately 10%.

3.5.2.5. Discussion and Insights

This section discusses the overall work done on this research work. Six datasets were taken for evaluating the MISFS. All repositories are of human being diseases. Accuracy or classification rate, sensitivity or recall, specificity, precision, and F-measure (or F1-score) are used for evaluation. Several features selected were also an important part of this research.

Evaluating the MISFS algorithm on the various performance metrics, its results were influential. Table 3.11 shows the accuracy of the proposed algorithm, MISFS, on different datasets with a comparative performance study with other methods, as explained in Section 3.5.1.3.

Table 3.11. Summary of MISFS accuracy compared with PCC and datasets considering all features

	MISFS		PCC		ALL FEATURES	
Classifier	LR	KNN	LR	KNN	LR	KNN
Pima	75.75	77.48	72.39	73.95	72.39	65.62
Spectfheart	79.012	76.540	85.07	77.61	72.89	67.90
Breast cancer	97.07	94.73	94.40	96.50	96.50	96.50
Parkinson	89.83	94.91	89.79	85.71	83.67	89.79
PCOS	87.11	84.66	85.29	89.70	86.02	67.64
Cleveland	58.88	52.22	56	56	58.66	48.89

It can be noticed from Table 3.11 that considering all six datasets, except Spectfheart and PCOS, the performance of MISFS is better, either through KNN or LR or both

when compared with the performance of PCC and all feature datasets. The highest accuracy received by the algorithm is 97.07%. Considering the Spectfheart dataset, it could be noticed that PCC with LR only received a higher accuracy of 85.07%, and accuracy by other methods is approximately similar to the accuracy achieved by MISFS.

Comparing the performance of all the datasets with the existing work (Table 3.5 – Table 3.10), it can be seen that the performance of the proposed MISFS algorithm is similar to or approximately better than the existing work performance. The result of existing work for each dataset is separately shown in Tables 3.5 – 3.10 for Pima Indian diabetes, Spectfheart, Breast Cancer, Parkinson, PCOS, and Cleveland, respectively. The highest accuracy received by the proposed algorithm is 97.07 using LR, and the lowest accuracy received by MISFS is 75.75 by LR only. Overall the performance of the MISFS is better for most datasets and is approximately similar to the few other datasets. Also, the features are reduced more compared to other existing work, and the accuracy achieved is high. So, MISFS can be used for feature selection and precision.

In terms of feature selection, the number of features optimized by MISFS is more compared to existing work with higher accuracy. Also, more than 85% reduction is made by MISFS on higher feature datasets and more than 50% reduction in smaller feature datasets (Figure 3.6). It concludes that the MISFS is better in both feature selection and accuracy on datasets with higher features.

The challenges mentioned in the literature for classification problems and feature selection include reducing the number of features, computational complexity, computational cost, storage space, increasing accuracy, and the ratio of features selected. The MISFS methods achieved most of these including a reduction in features at a higher rate on high dimensional data and an average rate on low dimensional data—helping in reducing the storage space to some extent and reducing the ratio of features selected. Have received up-to-the-mark accuracy compared to existing methods, reducing the computation time – thus reducing computation complexity and cost.

While the MISFS algorithm demonstrates strong performance in terms of dimensionality reduction, classification accuracy across multiple datasets, scalability remains an important consideration. The current evaluation focuses on six benchmark datasets with varying feature dimensions. While this reflects reasonable efficiency for

medium-sized datasets, applying MISFS to very large-scale or ultra-high-dimensional datasets may require further optimization to handle increased computational demands. The algorithm's complexity primarily depends on the number of features and the iterative selection process. Incorporating parallel computation, adaptive feature grouping, or incremental selection mechanisms could enhance its scalability for big data environments. Future studies could also include a broader range of datasets from different domains (e.g., text, image, genomic) to validate the algorithm's generalizability.

3.6. Summary

To improve the performance of the prediction model both in accuracy and feature selection, this paper proposed a hybrid algorithm MISFS which combines MI and SFS using cross-validation. The main motivation of this research is to improve the accuracy and feature selection for classification problems. The result is compared with existing work and different other methods to judge the performance of MISFS. Also, computational complexity is improved. Furthermore, the proposed approach is used in disease diagnosis datasets, but it can be further used in different application areas such as engineering applications, intrusion detection, text recognition, and many more. It can be concluded from the results that the hybrid approach proposed in this research work can be used for feature selection and also for measuring classification accuracy as the number of features selected with this method decreases with larger attributes in the dataset. Also, the performance is greater than the existing work done on these datasets. The highest accuracy achieved by the proposed MISFS is 97.07%. The proposed method meets most of the challenges mentioned in the literature.

The outcome of this chapter serves as the foundation for the subsequent stages of the research. The reduced feature sets obtained through MISFS were used as input to design and optimize fuzzy rule-based systems (FRBS). In Chapter 4, different learning and optimization techniques for FRBS are explored to improve adaptability and performance. Building upon this, Chapter 5 further investigates the use of Genetic Algorithms (GA) and Particle Swarm Optimization (PSO) for fine-tuning membership functions, thereby evaluating the model on different parameters and enhancing the overall effectiveness of the fuzzy inference system.

Although the MISFS algorithm demonstrates linear computational complexity and performs efficiently on medium-scale datasets, its scalability may be affected in very high-dimensional or large-sample datasets. To address this, future adaptations may consider parallel implementations, feature grouping strategies or approximate mutual information estimations to enhance efficiency for large-scale problems.

CHAPTER 4

LEARNING AND OPTIMIZATION TECHNIQUES FOR FRBS

4.1. Introduction

The motivation behind the study and development of FRBS is to address real-world problems related to social, economic, and environmental decisions. The process of dealing with these decisions is called sustainable development. Parameters defined by sustainable decision-making are complex and ill-defined having a high degree of uncertainty [185]. Fuzzy logic [1] is considered to be a standard tool for appraising sustainability.

This chapter's primary goal is to conduct an exhaustive investigation of the learning and tuning methodologies in FRBS. Fuzzy logic is a tool of artificial intelligence dealing with uncertainty. The term “fuzzy” comes from this vagueness. It is a mathematical model of representing vague and imprecise information. This research chapter aims to:

- i. Examines different learning approaches and their authority in constructing definite fuzzy rule-based systems.
- ii. Analyzing various tuning techniques for FRBS, for optimizing rule base and membership function, in order to increase the systems performance and as well as interpretability, in order to minimize the trade-off between interpretability and accuracy.
- iii. Conduct a comparative analysis of different learning and tuning techniques to identify their strengths, weaknesses and suitability for various applications.
- iv. Provide valuable insights into the latest practices in the field and their practical implications for real-world applications.

4.2. Related Work

Fuzzy systems, developed by Zadeh in 1965 [1], have found their most powerful application in FRBS. These systems extend traditional rule-based systems by dealing with fuzzy rules rather than classical logic rules, making them highly effective in multiple areas where uncertainty and vagueness are prevalent. These systems utilize fuzzy logic to represent different forms of knowledge and solve diverse problems across various domains [185]. An FRBS comprises three principal components: fuzzification, rule-base, and defuzzification. The fuzzification process transforms crisp input values into fuzzy sets using appropriate membership functions. The RB contains fuzzy if-then rules that define the connections between inputs and outputs. Finally, the defuzzification process converts fuzzy output sets back into precise values, yielding the system's ultimate output. The inference system and knowledge base (KB) are key components of an FRBS, with the fuzzy inference process deriving outputs based on provided inputs and the KB comprising both the DB and fuzzy rules pertinent to the problem being addressed.

Two prominent types of FRBS are Mamdani [186] and Takagi-Sugeno-Kang (TSK) [5], which differ in how they express the consequent of the fuzzy rules within the KB. Mamdani system uses linguistic variables for the consequent, while TSK uses polynomial functions of the inputs. Figure 1.1 already shows the fundamental structure of Mamdani FRBS. The linguistic rule structure of Mamdani FRBSs is covered in Chapter 1 from equation 1.1 – 1.4.

A more adaptable KB structure has been introduced recently, enabling enhanced accuracy in linguistic models while maintaining their interpretability [187]. This updated approach allows linguistic rules to be associated with two distinct consequents. Such a modification enables FRBS to achieve improved local accuracy in complex regions of the data space, as the rule output can now fall in an intermediate range between the two primary labels. This adjustment also carries a clear linguistic interpretation:

$$\text{IF } X_1 \text{ IS } A_1 \text{ and ... and } X_n \text{ is } A_n \text{ THEN } Y \text{ is between } B_1 \text{ and } B_2. \quad (4.1)$$

Recently trends seen the emergence of a new variation of these systems that prioritizes accuracy over interpretability. Referred to as approximate Mamdani-type FRBSs[179, 182] these differ from traditional systems, which are often labelled as linguistic or descriptive Mamdani-type FRBSs due to their emphasis on clarity and interpretability. The primary focus of approximate systems lies in fuzzy modelling [181, 185], where achieving high precision in predictions takes precedence over the ability to describe the model linguistically. Diagrammatically approximate and descriptive FRBS structure is similar as shown in Figure 1.1. The difference lies in the consideration of rules in KB. These rules in approximate FRBS do not use linguistic variables but directly use fuzzy variables. Thus, IF-THEN rules in approximate FRBS is

$$IF X_1 \text{ is } A_1 \text{ and ... and } X_n \text{ is } A_n \text{ THEN } Y \text{ is } B. \quad (4.2)$$

Fuzzy systems have emerged as innovative tools to model complex and ill-defined systems. Applying fuzzy logic to rule-based systems results in FRBSs, which utilize IF-THEN rules with fuzzy logic statements for both antecedent and consequent. The key advantages of FRBSs over classical rule-based systems include:

- The ability to handle uncertainty using fuzzy sets.
- Enhanced robustness and flexibility of inference techniques due to fuzzy logic approximate reasoning methods.

The three main inference methods in FRBS are Mamdani, TSK, and Tsukamoto. Each method has its unique characteristics, such as Mamdani's use of intersection and union operators for implication and aggregation, resulting in fuzzy outputs requiring defuzzification; Sugeno's use of crisp functions as consequents, producing crisp outputs combined as a weighted average; and Tsukamoto's use of monotonic membership functions in consequents, producing crisp outputs directly.

Despite significant progress in FRBS learning and tuning, several challenges remain. Handling high-dimensional data while preserving interpretability is a major issue. Ensuring robustness against noisy and uncertain real-time data is also critical. Hybrid approaches need further investigation to create more accurate and adaptive FRBS models. Additionally, developing methods for online and incremental learning is essential for real-time adaptation as new data arrives. Addressing these gaps is crucial

for enhancing the effectiveness and applicability of FRBS in complex and dynamic environments.

4.3. Learning of FRBS

Learning Fuzzy Rule Based System involves developing models to automatically construct or optimize fuzzy rule sets based on data. In learning mechanism, a process is defined which helps in the learning of rule-based or of knowledge based. Figure 4.1 shows the different learning and tuning methods of FRBS which are explained in section 4.3, 4.4 and 4.5.

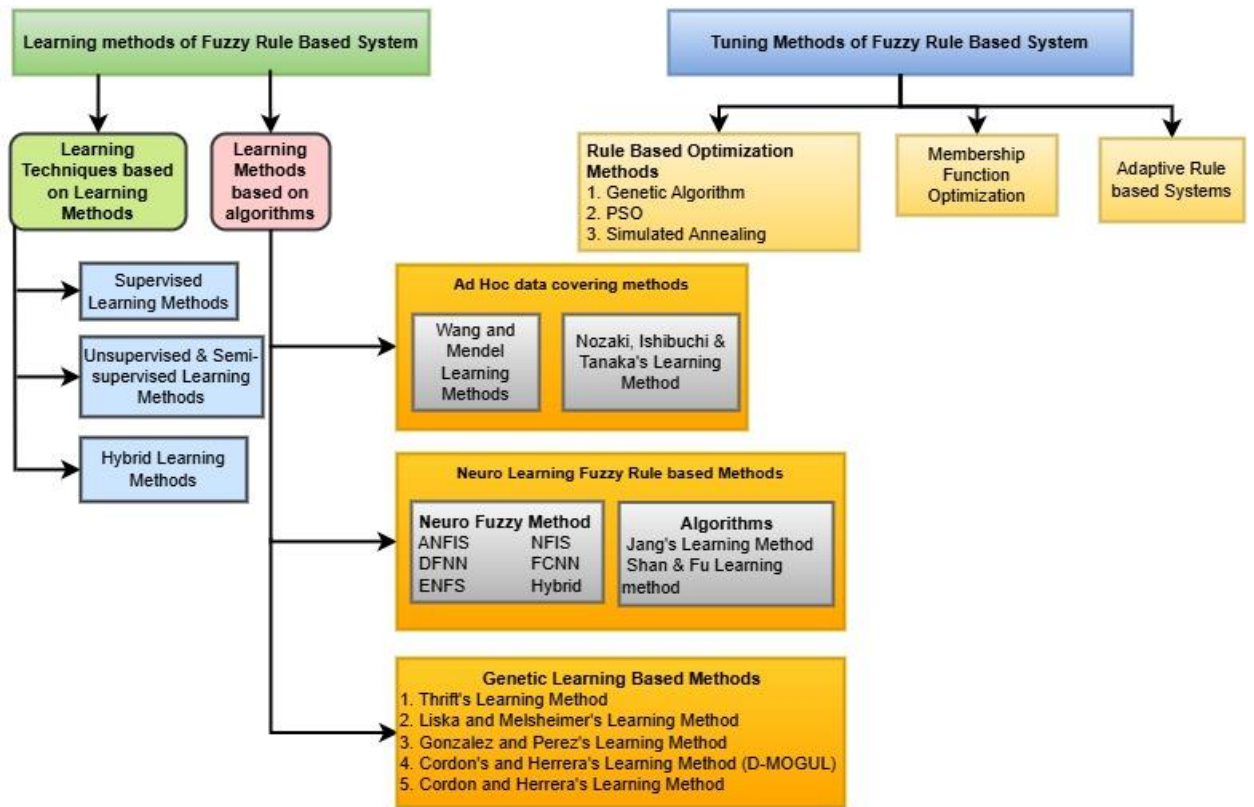


Figure 4.1. Learning and Tuning of Fuzzy rule-based methods.

This section discusses briefly learning techniques based on learning methods and performance evaluation metrics known to be used in FRBS:

- i. **Supervised Learning:** This approach involves utilizing labelled training data [43] to both learn the fuzzy rules and optimizing the MFs. Gradient-based optimization, evolutionary algorithms, and particle swarm optimization are some methods

employed to iteratively update the rule parameters and to improve the model's accuracy.

- ii. **Unsupervised and Semi-Supervised Learning:** Unsupervised techniques, such as fuzzy clustering, identifies fuzzy rules and memberships from unlabelled data. Semi-supervised learning [43] integrates labelled and unlabelled data to enhance the model's learning process and robustness.
- iii. **Reinforcement Learning:** Integration of reinforcement learning [186, 187] with Fuzzy Rule Based System (FRBS) allows the system to learn optimal policies by receiving feedback in the form of rewards for favourable actions and penalties for unfavourable ones. This combination enables adaptive and intelligent decision-making in dynamic environments.
- iv. **Hybrid Learning Schemes:** Hybrid learning schemes [140] combine multiple learning techniques/approaches to improve the performance of FRBS. As an example, a hybrid approach might use supervised learning to initialize the fuzzy rule set and then tune those rules using an evolutionary algorithm.

Performance evaluation metrics: Evaluating the performance of an FRBS is essential to gauge its ability to unseen data. Common metrics for evaluating [135, 183] FRBS performance include Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Accuracy, F-measure, Cross-Validation. The choice of technique and metric depends on the problem characteristics and data nature.

4.4. Learning Algorithms Fuzzy Rule-Based Systems

Designing intelligent systems using FRBS involves two main tasks: defining the fuzzy inference process and deriving an adequate KB. The fuzzy inference process involves selecting fuzzy operations and applying fuzzy rules for decision-making, whereas creating the KB involves defining linguistic variables, MFs, and fuzzy rules as per the problem considered. This is complex and often tackled with automatic techniques from soft computing [192], including Fuzzy Logic, NNs [193], and GAs [193] shown in Figure 4.2.

It is essential to distinguish between learning and tuning problems. In tuning problems, a predefined rule base and a preliminary database are used and the objective is to find a better set of parameters defining the database. In learning problems, a process including the learning of the rule base or the whole knowledge base is performed. Several learning algorithms for fuzzy rule-based systems are grouped into ad-hoc covering approaches, neural networks, and genetic algorithms.

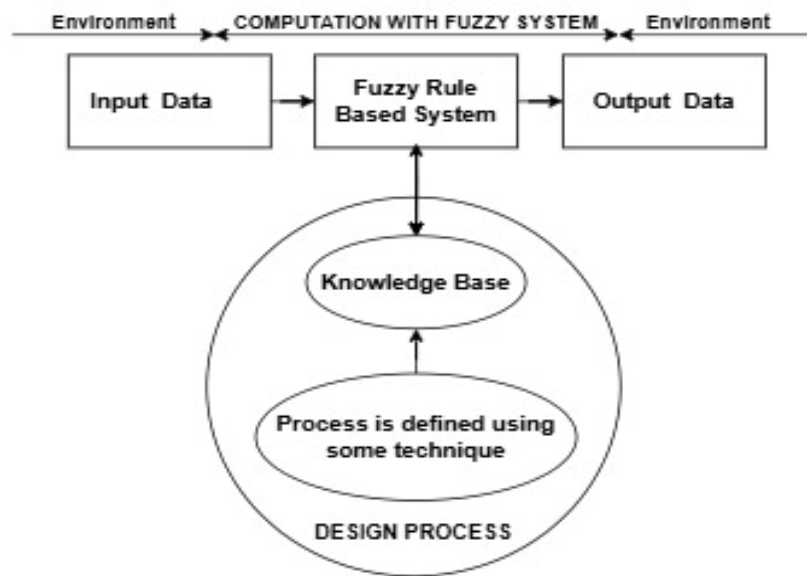


Figure 4.2. Design of FRBSs [191]

4.4.1. Ad-Hoc Covering Methods: Ad-hoc data covering methods learn fuzzy rules based on data characteristics through iterative procedures. Notable methods include Wang and Mendel's Learning Method [194], which generates fuzzy rules from numerical data by dividing input and output variable spaces into fuzzy sets, generating linguistic fuzzy rules, assigning degrees to each rule, creating a combined fuzzy rule base, and mapping to create the final rule base. This method is simple and performs well, prioritizing high-quality rules. Another method, Nozaki, Ishibuchi, and Tanaka's Learning Method [195], aims for good interpretability and high accuracy by using the TSK model to generate a fuzzy model and then transforming it into a linguistic fuzzy model. This involves generating simple TSK rules and deriving Mamdani type rules from them, maintaining performance while enhancing interpretability. Figure 4.3 shows an example of a uniform fuzzy partition while Figure 4.4 shows the derivation of Mamdani rules from TSK rules.

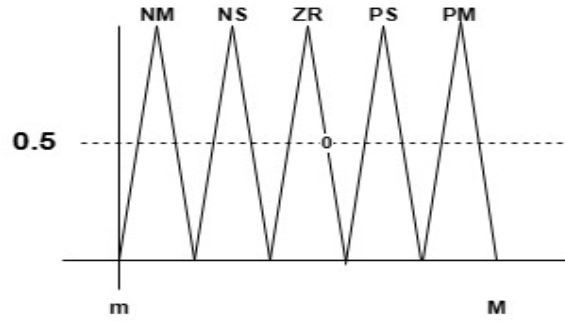


Figure 4.3. Graphical representation of a uniform fuzzy partition

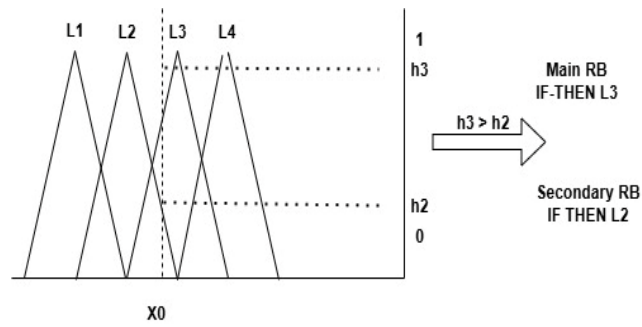


Figure 4.4. Deriving Mamdani rules from TSK rules [10]

4.4.2. *Neural Network-Based Learning Methods*: Neural network-based learning methods [193] for FRBS blend the strengths of neural networks with the interpretability of fuzzy models, enhancing their capability to manage complex, high-dimensional data. Key approaches include ANFIS, which merges Takagi-Sugeno fuzzy inference with backpropagation learning, and NFIS, integrating neural networks with Mamdani-type inference. Advanced variants like Deep Fuzzy Neural Networks (DFNN) and Fuzzy Convolutional Neural Networks (FCNN) employ deep learning and convolutional techniques to handle data complexity and uncertainty. Evolving Neural Fuzzy Systems (ENFS) and Hybrid Neural-Fuzzy Classifiers combine evolutionary algorithms and neural networks to evolve rule bases and enhance interpretability. Notable examples include Shan and Fu's Learning Method [196], which uses a layered neuro-fuzzy system with rule pruning to refine the rule base, and Jang's ANFIS [197], shown in Figure 4.5, a hybrid model that iteratively refines fuzzy rules using a combination of least squares and gradient descent methods. These methods are

widely used for function approximation, system modelling, and control, offering a robust framework for complex data patterns.

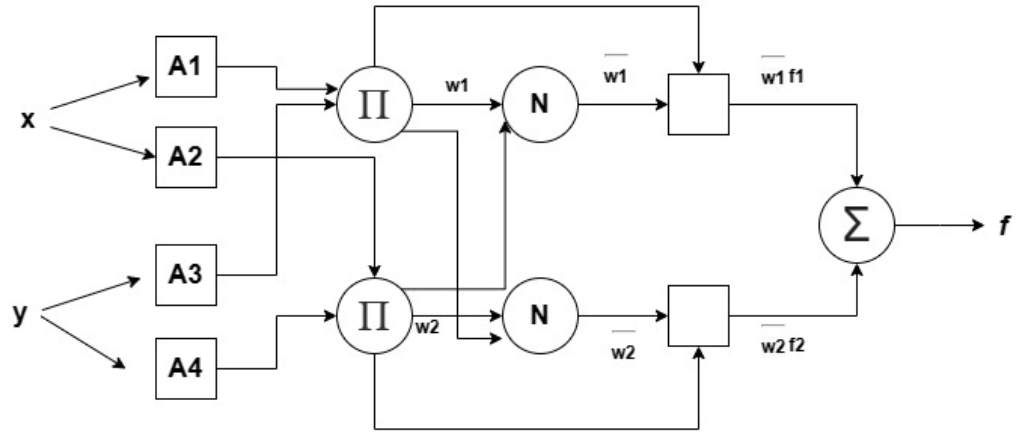


Figure 4.5. Structure of ANFIS

4.4.3. *Genetic Learning Fuzzy Rule-Based Systems (GLFRBS)*: GLFRBS[198] integrate GA with FRBS to optimize rule bases for specific problems. GAs, inspired by natural selection, search for optimal solutions in large spaces. The GLFRBS process begins with initial population of fuzzy rule bases, each representing a set of fuzzy rules with associated MFs and weights. Fitness evaluation assesses each rule base's performance based on an objective function, such as classification accuracy. The selection process favours high-performing rule bases for reproduction, followed by crossover as well as mutation for creating new rules. The offspring, combined with some top-performing individuals from the previous generation, replace the old population. This cycle continues until a termination condition, such as a fitness threshold or maximum generations, is met. The best-performing rule base is then selected.

Genetic algorithms are also applied to fuzzy rule-based systems through different approaches, including Michigun, Pittsburgh, and Iterative Rule Learning (IRL). Notable methods include:

- *Thrift's Learning Method*: This method is based on encoding all the cells of the complete decision table in the chromosomes. This method utilizes genetic algorithms with integer coding [199], employing elitist selection and genetic

operators like mutation and crossover. Fitness is measured based on the FRBS's performance in reaching equilibrium points in controlled systems.

- *Liska and Melsheimer's Learning Method*: They proposed [200] a method that focuses on all aspects of knowledge-based (KB) learning, including the number and structure of fuzzy rules and MF parameters, with subsequent tuning for optimal adjustments. This method simultaneously does learning-optimizing of all three parts. This method represents the KB as a chromosome composed of three substrings, encoding membership function associated with all system variables, the structure of each rule in RB, and encoding membership functions in the rule consequents respectively. The performance of each chromosome in a population is evaluated by an exponential ranking technique based on the error function. After evaluating the fitness function and finishing the genetic learning phase, tuning of the membership function is done for optimization using a multilayer feedforward network to avoid the stagnation of GA.
- *Gonzalez and Perez's Learning Method (SLAVE)*: They proposed a Structural Learning Algorithm in Vague Environments (SLAVE), which is an inductive learning algorithm designed for vague environments, and has been developed and established for effective fuzzy rule-based learning [195, 196]. The rule structure for SLAVE is similar to Equation 1.1 covered in Chapter 1. Another feature of this learning algorithm is that it uses an IRL approach for generating the rule set. Figure 4.6 shows the different elements of the SLAVE learning algorithm and the relations among them.
- *Cordon and Herrera's Learning Method (D-MOGUL)*: The approach designed by these authors is based on the MOGUL paradigm presented in the paper[203]. This approach automatically generates a complete knowledge base [188] in three steps: iterative rule-base generation, simplification of the genetic approach, and genetic tuning.
- *Cordon & Herrera's WM-ALM Method*: This methodology (Wang Mendel-Accurate Linguistic Model) aims to design high-accuracy linguistic models [187] with comprehensive descriptions, enhancing the precision and clarity of FRBS. These methods highlight the flexibility of genetic algorithms in optimizing fuzzy rule-based systems, enabling automatic discovery of rules and tuning of parameters.

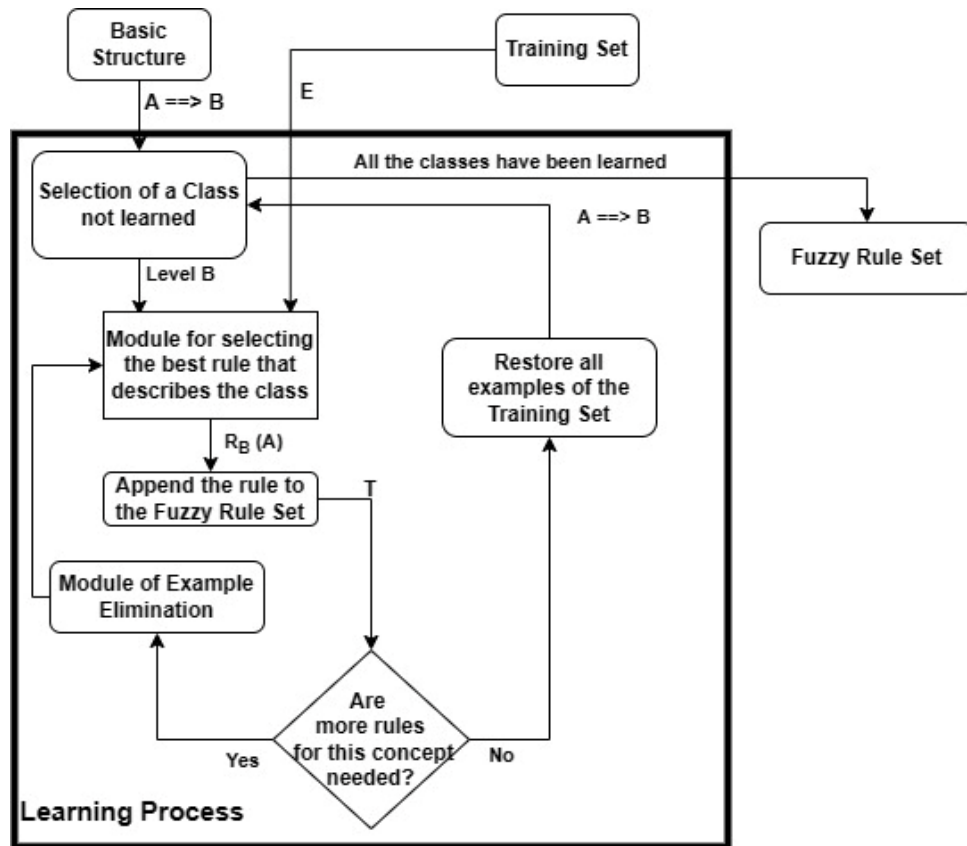


Figure 4.6. Learning through SLAVE [10]

4.5. Tuning Fuzzy rule-based system

The performance of the Fuzzy rule-based system depends on the RB and the MFs associated with the DB. It is important to tune these parameters as per the need of the model to be processed. Tuning FRBS is essential for refining the parameters of the rule base and MFs to enhance performance and achieve a balance between accuracy and interpretability. The tuning process involves optimizing both the rule base, which defines the system's behavior, and the MF [195] that represent the fuzzy partitions. This is done by decreasing the error function based on model or training set as evaluated.

4.5.1. *Rule Base Optimization Methods* [10, 198] involve finding the best combination of fuzzy rules and their parameters. Popular methods include:

- *Genetic Algorithms*: This method mimic natural selection [198] to refine the ruleset through selection, crossover, and mutation, converging toward an optimal solution.

- *Particle Swarm Optimization*: This technique [198] is inspired by social behaviours, particles representing potential solutions move within the search space, adjusting positions based on individual and collective experiences.
 - *Simulated Annealing*: Inspired by metallurgical annealing [198], it explores the search space freely at high temperatures and gradually cools to converge toward an optimal solution.
- 4.5.2. *Membership Function Optimization* [10, 181] focuses on adjusting the parameters to accurately represent input data uncertainty and linguistic interpretation. Techniques include gradient-based methods, evolutionary algorithms, and neural network-based approaches such as Jang’s ANFIS and Cordon and Herrera’s tuning method.
- 4.5.3. *Adaptive Rule-Based Systems* [181, 198] dynamically adjust to changes in data or the environment by continuously monitoring performance and updating the fuzzy rule set. Techniques like online learning, reinforcement learning, and dynamic genetic algorithms facilitate adaptive behaviour.
- 4.5.4. *Interpretability-Accuracy Trade-Off* is a critical consideration in tuning FRBS. While interpretable rule sets enhance understanding, they may reduce accuracy, whereas complex rule sets improve accuracy but become less interpretable [10, 198]. Balancing this trade-off involves adjusting rule base complexity, the number of rules, and using optimization techniques that manage model complexity constraints.

The effectiveness of these tuning methods depends on the problem domain, available data, and FRBS complexity. Experimentation and comparison of different strategies are often necessary to identify the most suitable approach for a given application.

4.6. Experimental Evaluation and Comparative Analysis of FRBS Learning and Tuning methods

This section provides a critical evaluation of various learning and tuning methods for fuzzy rule-based systems based on experimental results reported in existing literature.

The analysis highlights the comparative performance, interpretability and computational characteristics.

4.6.1. Advantages and disadvantages of FRBS learning and tuning methods: The study covers three methods for learning and tuning FRBS: Ad Hoc Data Covering Methods, Neural Learning FRBS, and GLFRBS. Comparative analysis of each is done in Table 4.1.

Table 4.1. Benefits and Drawbacks of different learning and tuning methods

Method	Advantages	Disadvantages
Ad Hoc Data Covering Methods	High interpretability, simplicity, expert knowledge integration	Limited scalability, flexibility, and automation
Neural Learning Fuzzy Rule Based Methods	High adaptability, automation, and accuracy	Complexity, potential interpretability issues, high computational resources
Genetic Learning Fuzzy Rule Based Methods	Global optimization, flexibility, balance of exploration and exploitation	Computationally intensive, parameter sensitivity, complexity

4.6.2. Comparative performance of learning and tuning methods across different metrics: The comparison of learning methods for FRBS shown in Table 4.2 examines Ad Hoc, Neural Network, and Genetic Learning methods across various criteria. The choice of learning method for FRBS should align with the application's needs. Ad Hoc Methods are best for small-scale, expert-driven systems due to their high interpretability and efficiency but have limited adaptability and scalability. Neural Network Methods, suited for dynamic and real-time systems, offer high accuracy and adaptability but are less interpretable and moderately efficient. Genetic Learning Methods excel in large-scale, complex optimization tasks, with high flexibility and scalability, but are computationally intensive and moderately interpretable.

Table 4.2 shows that Ad Hoc methods are best suited for applications where interpretability and simplicity are crucial, such as in expert systems or small-scale projects with significant human expertise involvement while Neural network methods excel in environments where adaptability to new data and automation are critical, such as dynamic systems and real-time applications. These methods are preferable when

high accuracy is required and interpretability is less of a concern and Genetic learning methods are ideal for complex optimization problems and applications requiring a balance between exploration and exploitation of the search space. They are well-suited for large-scale and diverse datasets where finding the global optimal solution is essential.

Table 4.2. Suitability and Performance metrics comparison of three approaches

Criterion	Ad Hoc Methods	Neural Network Methods	Genetic Learning Methods
Application Suitability	Expert-driven, small-scale systems	Dynamic, real-time systems	Complex optimization problems, large-scale systems
Interpretability and Automation	High interpretability and simplicity	Lower interpretability, higher complexity	Moderate interpretability, requires additional techniques
Adaptability and Automation	Limited adaptability, high reliance on human expertise	High adaptability and automation, suitable for dynamic data	Moderate adaptability, good for diverse datasets
Global Optimization and Flexibility	Low focus on global optimization	Moderate flexibility, good local optimization	High focus on global optimization, high flexibility
Performance Metrics			
Accuracy	Moderate	High	Moderate to high
Computational Efficiency	High (Most efficient)	Moderate	Low (most intensive)
Scalability	Low Scalability	Moderate Scalability	High Scalability
Interpretability	Best	Requires Additional Techniques	Moderate, requires additional techniques

4.6.3. Comparative Study of FRBS Learning Methods and Tuning Methods: Table 4.3 presents a qualitative comparison of major FRBS learning and tuning methods, based on key performance metrics reported in literature. The evaluation reflects observed trends in accuracy, interpretability, computational efficiency, rule count rather than specific dataset-dependent values.

This table highlights the competency of various learning methods in FRBS. These techniques provide high accuracy, with specific methods like ANFIS, SLAVE, and D-

MOGUL excelling in performance. Methods like the Wang-Mendel Method and Shan Fu offer high interpretability due to their simpler rule sets. In terms of computational efficiency methods such as Nozaki et al. and Liska and Melsheimer are highly computational, suitable for real problems. Considering rule count, the number of rules generated varies widely among the methods, influencing both interpretability and complexity. Comparing both Jang's ANFIS for tuning and Cordon and Herrera's tuning method achieve high accuracy with moderate to high interpretability. They maintain moderate computational efficiency and apply to a variety of datasets, including control systems and classification tasks.

Table 4.3. Learning Methods and Tuning Methods Comparison

Method	Algorithm	Accuracy	Interpretability	Computational Efficiency	Rule Count
[194]	Wang-Mendel Method	Moderate to high	High	Moderate	Moderate
[195]	Nozaki	High	Moderate	High	Variable
[196]	Shan and Fu	Moderate	High	Moderate	Moderate
[197]	ANFIS	High	Moderate	Moderate	Variable
[199]	Thrift	High	Moderate	High	Variable
[200]	Liska and Melsheimer	High	Moderate	High	Variable
[201]	SLAVE	High	Moderate	Moderate to high	Variable
[203]	D-MOGUL	High	Moderate to high	Moderate	Variable
[187]	WM-ALM	High	Moderate	Moderate to high	Variable
[197]	ANFIS for tuning	High	Moderate	Moderate	Variable
[188]	Tuning	High	Moderate to high	Moderate	Variable

4.7. Summary

Fuzzy Rule-Based Systems (FRBS) offer robust solutions for managing complexity and uncertainty in various applications. This research work explored different types of FRBS, including learning and tuning methods for FRBS, emphasizing the importance of balancing accuracy and interpretability. Through comprehensive analysis and comparison, valuable insights were provided into the strengths and weaknesses of

various approaches. In the present work the authors reviewed ad-hoc learning methods and techniques like NNs and GAs which can be applied for learning of FRBS. This paper explores various methods proposed by different researchers some in brief and somewhat detailed with their implementation details. Further the advantages and disadvantages of learning methods as well as tuning methods are covered. Further these approaches are compared on the basis of performance metrics as well as different proposed algorithms are compared on the basis of rule-base, interpretability and accuracy.

It is concluded that Neural learning methods often provide the highest accuracy due to their advanced capabilities while Ad hoc methods are the most computationally efficient, followed by neural methods, with methods being most intensive. Further, Genetic methods scale better with large and complex datasets compared to ad hoc methods. Ad hoc methods offer the best interpretability, with neural and genetic methods may require additional techniques to make the results interpretable.

CHAPTER 5

OPTIMIZING FRBS USING GA AND PSO

5.1. Introduction

Fuzzy Rule-Based Systems (FRBS) are widely recognized for their ability to handle uncertainty and model complex systems using human-understandable rules. Interpretability and accuracy are two important parameters to measure the performance of any FRBS[205]. Interpretability in FRBS is about the understandability and simplicity of the fuzzy rules and the structure of the fuzzy model. It is a fundamental aspect because one of the primary goals of fuzzy systems is to emulate human reasoning in a way that is easily interpretable [206]. In FRBS, accuracy is the system's ability to correctly approximate desired outputs or make correct decisions based on the fuzzy rules. A critical challenge in FRBS design is balancing interpretability and accuracy. While interpretability is often associated with simpler rule sets and membership functions, this can lead to a loss in accuracy. Conversely, enhancing accuracy typically results in increased system complexity, reducing interpretability. While there is generally a trade-off between accuracy and interpretability, certain techniques and strategies can help in optimizing both to a certain extent. This research work explores the simultaneous improvement of both accuracy and interpretability in FRBS by optimizing membership functions [206] through advanced tuning techniques. Specifically, we investigate the use of GA and PSO to fine-tune [207] the membership functions, aiming to optimize balance between accuracy and interpretability. Experimental results demonstrate that both GA and PSO can effectively enhance the performance of FRBS by refining membership functions, leading to significant improvements in accuracy while maintaining or even enhancing system interpretability. This approach offers a promising direction for more efficient and user-friendly FRBS in various applications.

5.2. Trade-off between Interpretability and Accuracy

Interpretability is an important aspect of FRBS, referring to how easily humans can comprehend the rules and structure of the system. It is a defining feature of fuzzy systems, as one of their primary objectives is to replicate human reasoning in a way that is accessible and comprehensible. High interpretability ensures that the fuzzy rules are simple, transparent, and closely aligned with human intuition, which is particularly important in applications requiring user trust and understanding. For instance, in medical diagnosis systems [208], clinicians must understand the reasoning behind the system's decisions to make informed judgments. **Parameters for measuring interpretability** in FRBS include, for example, the number of rules, the complexity of the rule antecedents (e.g., the number of conditions per rule), the comprehensibility of the linguistic terms used, and MFs granularity. Interpretability of FRBS is directly related to the complexity of the fuzzy rules and the shapes of the MFs used. Simpler rule sets and more straightforward membership functions contribute to higher interpretability, making it easier for users to trace the decision-making path and understand how specific inputs lead to certain outputs. However, achieving high interpretability often requires a trade-off, as overly simplistic rules may fail to capture the nuances of the underlying data, leading to reduced accuracy.

Accuracy in FRBS refers to the system's ability to make correct decisions or approximate desired outputs based on the fuzzy rules and input data. It is an indicator of the system's effectiveness in accurately predicting or classifying outcomes. Metrics for measuring accuracy in FRBS include classification rate, MSE for regression tasks, including other statistical measures such as precision, recall, and F1-score, depending on the application. High accuracy is vital in applications where precise decision-making is critical, such as in automated control systems, financial forecasting, or autonomous vehicles. Accuracy in FRBS is influenced by the granularity and precision of the fuzzy rules and the configuration of MFs. More granular and finely-tuned rules typically enhance accuracy by capturing subtle variations in the data. However, enhancing accuracy often involves increasing the complexity of the rule base and MF, which can make the system less interpretable. In many practical scenarios, achieving the highest possible accuracy is paramount, but it must be balanced against the need for a clear, understandable system that users can trust and verify.

A significant challenge in the design of FRBS is balancing the trade-off between interpretability and accuracy. While interpretability emphasizes simplicity and ease of understanding, accuracy focuses on the system's performance and precision. In practice, improving one often comes at the expense of the other. For example, to enhance interpretability, one might simplify the rule set or use fewer, broader membership functions, which can reduce the system's ability to model complex patterns accurately. Conversely, increasing the number of rules and refining the membership functions can improve accuracy but result in a more complex and less interpretable system. This trade-off is a fundamental issue in the development of fuzzy systems, particularly in applications where both high accuracy and clear, understandable decision-making are required. Researchers and practitioners have been exploring methods to optimize this balance, seeking ways to maintain or improve accuracy while keeping the system interpretable.

5.3. Background

5.3.1. Interpretability in Fuzzy Systems and Machine Learning

Interpretability [204] is an important concept in both ML and FRBS, as it helps users understand how decisions are made by these systems. However, interpretability in these two fields has different meanings and implications. Interpretability in ML refers to the ability to explain or present the reasonings behind the predictions made by a model. While, in FRBS, interpretability is about the understandability and simplicity of the rules and the structure of the fuzzy model. It is a fundamental aspect because one of the prime goal of fuzzy systems is make the system interpretable. Table 5.1 differentiates the interpretability in FRBS to that of ML on various parameters:

5.3.2. Types of Interpretabilities in FRBS

There are two approaches to measure the interpretability of FRBS:

1. Complexity Based Interpretability[209] – It helps in decreasing the complexity of the model by measuring the number of rules, variables, labels per rule, etc.
2. Semantic Based Interpretability[209] – This method maintains the semantics of the MFs by applying limitations on Membership Functions or measures like distinguishability, coverage and others.

Table 5.1. Key differences between interpretability in FRBS and ML

Aspect	Interpretability in FRBS	Interpretability in ML
Definition	Interpretability is about how easily humans can understand the structure of the system.	Interpretability refers to the ability to explain the decision-making process of the model.
Importance	Essential for replicating human reasoning and ensuring the system is understandable and transparent to users.	Important for trust, transparency, and accountability, especially in critical applications (e.g., healthcare, finance).
Model Types	Generally high because FRBS uses human-readable rules and linguistic terms.	Often high in simple models (e.g., Linear Regression, Decision Trees) and low in complex models (e.g., Deep Neural Networks, Ensemble Models).
Factors involved	Number of rules, complexity of rule antecedents, comprehensibility of linguistic terms, and granularity of membership functions.	Complexity of the model, number of parameters, and feature interactions.
Interpretability Metrics	Number of rules, number of conditions per rule, simplicity of membership functions, and the transparency of the rule base.	Feature importance, SHAP, LIME, and rule extraction methods.
Trade-offs	Simpler rule sets enhance interpretability but can reduce accuracy; more detailed rules improve accuracy but reduce interpretability.	High interpretability often means lower model complexity and potentially reduced accuracy; more complex models (e.g., Neural Networks) can achieve higher accuracy but are less interpretable.
Techniques to Enhance Interpretability	Reduction of rules, simplification of membership functions, and use of linguistic terms that are easy to understand.	Use of interpretable models, dimensionality reduction, post-hoc explanations (e.g., visualization techniques, surrogate models).

These two types of interpretabilities have to be considered in KB components as well as linguistic fuzzy partition and RB, making the taxonomy double axis. These complexities can further be differentiated based on rule-based and fuzzy partition based. The taxonomy is shown in Figure 5.1.

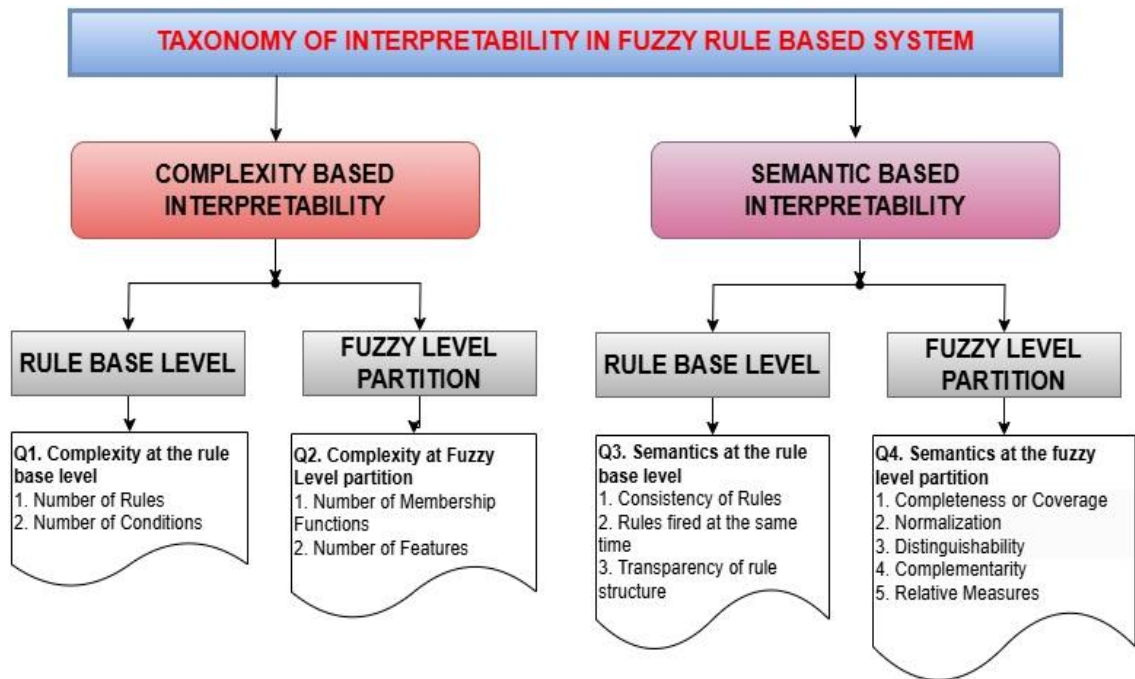


Figure 5.1. A taxonomy to analyze the interpretability of FRBS [209]

The brief description of these interpretability measures [210] with requirements to reduce the complexity is covered in the following Table 5.2:

Table 5.2. Interpretability parameters definition

Complexity at the rule-based level	
Number of Rules (NR)	Definition: Total number of fuzzy rules is the system. Interpretability Impact: Reduction in number of rules leads to simpler rule bases, thus enhancing interpretability. Understanding system's behavior and decision-making process becomes easier for users. System having too many rules is complex and difficult to interpret.
Number of Conditions (NC)	Definition: It refers to total number of antecedent conditions in a rule. Interpretability Impact: Small number of conditions typically ranging from 5 to 9 (7 ± 2) a person can easily understand and interpret. More conditions per rule makes system complex and difficult to understand.
Complexity at fuzzy level partitions	
Number of Membership Functions (MF)	Definitions: Each feature or variable has a set of Membership Functions to represent its Linguistic values. Total number of MFs needed to represent all features is called Number of MFs.

	Interpretability Impact: Small number of MF per feature reduces the complexity of the system and is easy to interpret. Normally, MFs ranging from 7 ± 2 per feature is conceptually a person can understand easily. Increasing number of MFs increases the granularity (number of linguistic terms or MFs) but at the same time increases complexity thus, reducing interpretability.
Number of features or variables (NF)	Definition: Number of features denotes the dimensionality of datasets. More features represent high dimensional dataset and less features represent low dimensional datasets. Interpretability Impact: Reducing the number of features can increase the readability of the KB. Feature selection is done for dimensionality reduction. Lower number of features makes the interpretation of system easier and reduce complexity and computational demands.
Semantics at the Rule Base level	
Consistency of the RB (C)	Definition: It denotes the degree of the rule to be consistent with each other without any contradictions in the system. Contradiction with respect to fuzzy system means that two or more rules provides opposing or conflicting outputs (consequent part) for the same or similar input conditions (similar premise or antecedents). Interpretability Impact: Consistent rules increase interpretability while inconsistent rules reduce interpretability. Depending upon the type of problem (classification, regression, control etc.) studied, the analysis of consistency will be different.
Number of rules fired at the same time (RF)	Definition: It means how many rules are activated at the same time for a given input in the system. Interpretability Impact: Minimizing the number of rules fired that are activated for a given input increases the interpretability.
Semantics at the fuzzy level partition	
Completeness or Coverage (CO)	Definition: It defines the extent to which the rules cover the entire input space to include membership function (MF) coverage across the universe of discourse for each variable. Interpretability Impact: When MFs cover the entire universe of discourse, each input point can be evaluated within at least one fuzzy set. This improves interpretability by ensuring that each input scenario is meaningful within the fuzzy system's structure, without "gaps" where inputs have zero membership in all fuzzy sets.
Normalization (NOR)	Definition: MFs are considered normal if there is at least one data point in the universe of discourse that has a membership value of one, corresponding to the maximum membership degree. Interpretability Impact: A normalized MF clearly represents its linguistic term by reaching full membership (1) at a certain point, making it easier to interpret
Distinguishability (DIS)	Definition: An MF has high distinguishability if it represents a unique linguistic concept and doesn't overlap too much with adjacent MFs. High distinguishability

	prevents confusion between MFs, ensuring that each MF reflects a clear and distinct part of the variable's range. Interpretability Impact: It helps in enhanced clarity, reduced ambiguity and improved consistency.
Complementarity (COM)	For every element in the universe of discourse, the sum of all its membership values should be close to one. This ensures a balanced distribution of meanings across the elements.

5.3.3. Interpretability Index

Various approaches have been available to analyze the interpretability of fuzzy models. Following indexes are proposed in to measure the interpretability of the fuzzy rule set:

1. Compactness: It means that rules must be small in number for better interpretability. Compactness = Number of levels
2. Similarity: It means that similar rules must be minimized to enhance the rule distinguishability.
3. Redundant: rules must be avoided.
4. Consistency, avoiding incoherent rules.
5. Completeness or Coverage: Table 5.2 covers the definition of coverage.

5.3.4. Relationship between Accuracy and Interpretability in FRBS

There exists a relationship between accuracy and interpretability in FRBS, often described as trade-off, meaning that improvements in one sometimes leads to a decrease in the other.

5.3.5. Learning and Tuning in FRBS

It is possible to increase both interpretability and accuracy simultaneously in FRBS through learning and tuning methods. In spite of a trade-off between interpretability and accuracy, certain techniques and strategies can help in optimizing both to a certain extent. These are done through learning and tuning methods.

Learning [211] in FRBS typically involves the automatic generation or optimization of fuzzy rules and MFs from data, utilizing techniques such as fuzzy clustering, genetic algorithms, or integrating NNs with fuzzy logic, as seen in neuro-fuzzy systems like ANFIS[197] in Section 4.4. These methods help in constructing a rule base that is both meaningful and relevant to the underlying data distribution. On the other hand, tuning

[211] focuses on refining the fuzzy system's parameters to enhance its performance. The process involves adjusting membership functions or modifying rules using optimization techniques like gradient descent, evolutionary algorithms, or particle swarm optimization. By combining learning and tuning, FRBS can minimize the trade-off between interpretability and accuracy, ensuring that the systems remain both comprehensible to users and effective in decision-making.

In FRBS, both interpretability and accuracy can be enhanced simultaneously through strategic learning and tuning methods focused on optimizing membership functions. Learning methods, such as fuzzy clustering and ANFIS, generate well-structured and meaningful membership functions from data. Fuzzy clustering algorithms, like Fuzzy C-Means (FCM), create membership functions based on natural groupings in the data, ensuring interpretability and accuracy by reflecting actual data distributions. Similarly, ANFIS integrates neural networks with fuzzy logic to adjust parameters, using a hybrid approach of least-squares estimation and gradient descent, resulting in simple and data-aligned membership functions that preserve interpretability while enhancing accuracy. Tuning methods, such as gradient descent and GAs, further refine these membership functions post-development. Gradient descent fine-tunes parameters by minimizing errors, preserving the interpretability of membership functions while aligning them closely with the data for improved accuracy. GA optimize membership function parameters by evolving populations of solutions through genetic operators, balancing accuracy and interpretability by rewarding simpler, more effective solutions. Table 5.3 covers the detail of learning and tuning techniques in FRBS. Together, these methods enable the development of FRBS that are both user-friendly and highly effective in decision-making processes by carefully optimizing membership functions to balance the trade-offs between interpretability and accuracy. This complete table now clearly separates learning and tuning techniques for both rule-based and membership function optimization. Each entry describes how the technique works and its specific impacts on interpretability and accuracy, providing a clear and concise overview suitable for research purposes.

Table 5.3. Learning and Tuning Techniques in FRBS

Category	Type	Technique	How It Works	Impact on Interpretability	Impact on Accuracy
Learning Techniques	Rule-Based	Genetic Algorithms (GAs) for Rule Optimization	Evolves a population of rule sets through generations, optimizing a fitness function that balances accuracy and interpretability.	Produces concise, understandable rules by pruning unnecessary or redundant ones.	Enhances accuracy by optimizing the rule set to align better with data distribution.
		Fuzzy Clustering for Rule Generation	Uses algorithms like Fuzzy C-Means (FCM) to group similar data points and generate rules based on natural data clusters.	Creates rules that are intuitive and correspond to natural groupings in data.	Maintains or improves accuracy by generating rules that align closely with data structure.
	Membership Function	Fuzzy Clustering-Based Learning	Generates membership functions from clusters; centroids define means and spreads determine widths of membership functions.	Results in intuitive membership functions that reflect natural data groupings.	Data-driven membership functions enhance coverage of input space and improve accuracy.
		ANFIS	Combines fuzzy logic with neural networks; uses a hybrid learning approach (least-squares and gradient descent) to optimize membership function parameters.	Maintains interpretability with simple membership functions related directly to data distribution.	Iteratively refines membership functions to minimize prediction error, achieving high accuracy.
Tuning Techniques	Rule-Based	Rule Pruning and Simplification	Applies techniques to remove redundant or less significant rules, merge similar rules, or adjust membership functions to reduce complexity.	Simplifies the rule base, enhancing interpretability by reducing complexity without losing essential information.	Can improve accuracy by reducing noise or overfitting, leading to a more robust system.
		Multi-Objective Optimization	Uses evolutionary algorithms (e.g., NSGA-II) to optimize rules, balancing objectives like accuracy and interpretability simultaneously.	Ensures a balanced rule base, avoiding over-complexity, and finds optimal trade-offs between accuracy and interpretability.	Finds solutions that effectively optimize both accuracy and interpretability by balancing trade-offs.
	Membership Function	Gradient Descent-Based Tuning	Fine-tunes membership function parameters (mean, variance) to minimize loss	Preserves the simplicity and basic shape of membership functions, maintaining	Aligns membership functions more closely with the data

			functions (e.g., mean squared error).	interpretability while fine-tuning.	distribution, enhancing accuracy.
		Genetic Algorithm-Based Tuning	Optimizes membership function parameters through genetic algorithms; evolves populations using selection, crossover, and mutation to find optimal parameters.	Searches for simpler, more interpretable solutions by constraining membership function complexity.	Enhances accuracy by finding optimized membership function parameters that align better with the system's data.

5.3.6. Genetic Algorithm

GA [198] is an optimization method based on natural selection, operating on a population of possible solutions. It evolves these solutions through multiple generations to identify the optimal solution for a given problem. Figure 5.2. explains its working.

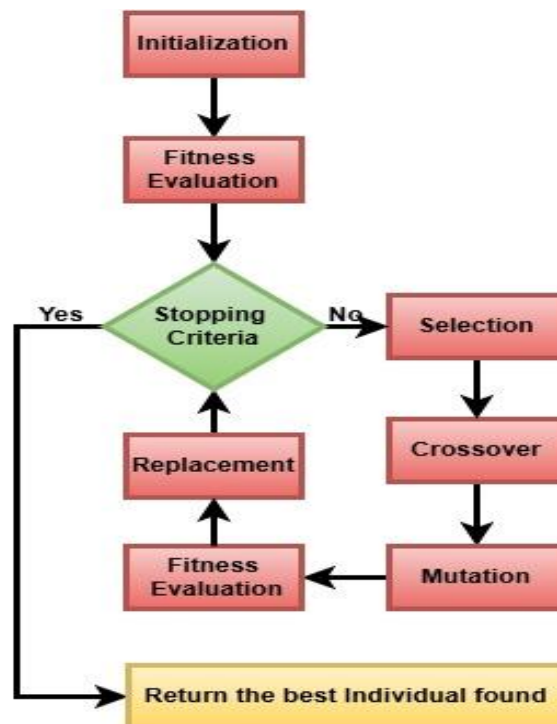


Figure 5.2. Flowchart of the Genetic Algorithm

The process involves the following steps:

1. Initialization: The initial population is generated with random values for the weights.
2. Fitness Calculation: The fitness of each individual is determined by how well it solves the problem at hand.

3. Selection: The tournament selection method is employed to choose pairs of individuals for reproduction. Individuals with greater fitness have a high probability of selection.
4. Crossover: Selected individuals undergo crossover, producing offspring with combined traits from both parents.
5. Mutation: Offspring may undergo mutation, where random changes are made to their weights to introduce diversity.
6. Fitness Evaluation: The fitness of the new offspring is again calculated.
7. Replacement: The old population is replaced with the new offspring, and the process repeats until convergence.

While there are many variations of the basic GA, the fundamental mechanism relies on three key operations:

- i. Evaluating fitness of the individuals;
- ii. Forming a gene pool (intermediate population);
- iii. Recombination and mutation.

A fitness function must be designed for each specific problem being addressed. For a given chromosome, which represents a potential solution, the fitness function returns a single numerical value that is intended to reflect the utility or adaptation of the solution represented by that chromosome. This fitness value serves as a measure of how well the solution solves the problem, guiding the genetic algorithm to prioritize better solutions during the evolutionary process.

There are various methods of selection in genetic algorithms. The crossover operator merges the traits of two parent solutions to produce two offspring, each inheriting features from both parents. Typically, this operation is performed at a random position within the chromosome, with the likelihood of it occurring determined by a predefined crossover probability. On the other hand, the mutation operator introduces random changes to one or more components of a selected solution, thereby increasing the diversity within the population. Each element in the solution vector undergoes a random modification based on a probability defined by the mutation rate, also known as the mutation probability.

5.3.7. Particle Swarm Optimization

PSO[212], developed by James Kennedy and Russell Eberhart in 1985, is a swarm-based optimization technique inspired by the social behavior of birds and fish. It operates on a population of particles, each representing a potential solution. The particles move through the solution space, guided by their best position and the global best position found by the swarm. The algorithm initializes particles with random positions and velocities. The fitness of each particle is evaluated, and the best positions (personal and global) called p_best and g_best are updated. A particle learns through two types of learning: social learning, where it gains knowledge from other particles, and cognitive learning, where it learns from its own past experiences. Through social learning, the particle remembers the best solution encountered by any particle in the swarm, known as global best (g_best). Through cognitive learning, the particle remembers the best solution it has encountered so far, referred to as personal best (p_best).

Initially, the position and velocities of each particle are set randomly. Also, the initial p_best and g_best are set for all particles. Particles then moves from its current position in the search space to find the solution, thus updating its velocity and position to find the personal best and global best. Let the i th particle position at time t be denoted by $x_i(t)$ and its velocity by $v_i(t)$.

The position of the i^{th} particle at time $t + 1$ is updated using the formula

$$p_i(t + 1) = p_i(t) + v_i(t + 1) \quad (5.1)$$

and the velocity update is given by the equation

$$v_i(t + 1) = w \cdot v_i(t) + c_1 \cdot r_1 \cdot (p_{best_i} - p_i(t)) + c_2 \cdot r_2 \cdot (g_best - p_i(t)) \quad (5.2)$$

The personal best (p_best) is updated as follows:

$$p_best_i = p_i(t) \text{ if } f(p_i(t)) < f(p_best_i(t)); \text{ otherwise } p_best_i(t) \quad (5.3)$$

where $p_i(t)$ is the current position of particle i at time t , $f(p_i(t))$ is the objective function value at the current position, $p_best_i(t)$ is the personal best position of particle i at time t , $f(p_best_i)$ is the objective function value at the personal best position of particle i

5.4. Related Work

To address this inherent trade-off, various learning and tuning methods have been developed for FRBS. Learning methods focus on constructing the initial fuzzy rule set and defining membership functions directly from data. Techniques such as FCM and ANFIS help derive initial rules and membership functions that reflect the underlying patterns in the data, providing a robust starting model that balances interpretability and accuracy. Tuning methods, however, are used to refine the MFs and adjust the rule base after the initial model is established. These methods involve fine-tuning the shapes, positions, and widths of the membership functions to improve the system's accuracy without significantly compromising interpretability. Advanced optimization techniques, such as GA and PSO, have proven particularly effective in this regard, as they iteratively search for optimal parameter settings that maximize accuracy while minimizing rule base complexity. The careful application of both learning and tuning methods can significantly enhance FRBS performance, ensuring that the system remains both accurate and understandable. In this section, the performance of the existing work is covered in Tables 5.4 and 5.5.

Table 5.4. Existing work on different metrics of FRBS

Reference	Area	Performance metrics	Interpretability Metrics	Findings
[213]	WBCD	Accuracy is measured	Total rules are used to measure interpretability	A fuzzy classifier that generates a small set of easy-to-understand rules for disease diagnosis.
[214]	Fraction of inspired oxygen (FiO ₂) settings in artificial verification	Root mean square error is measured	Total rules are used to measure interpretability	Best RMSE and rule count among Mamdani models
[215]	Cardiovascular disease risk analysis	Performance is measured in terms of error	Total rules are used to measure interpretability	IRIDIA outperformed ANFIS and TSK in RMSE for cardiac analysis.
[216]	WBCD	Recall, classification rate is measured	Total rules are used to measure interpretability	HFNN and FGPNP outperformed FNN in time complexity, performance, and interpretability.

[217]	WBCD	Classification Rate, Error, and Recall are calculated.	Total rules are used to measure interpretability.	ANFIS model reached 98.25% accuracy on the WBCD.
-------	------	--	---	--

Earlier research primarily concentrated on two parameters: NR and MF to assess model interpretability [212, 213, 214]. The significance of MFs and DIS was also explored by [221], with NR to evaluate interpretability. Both NR and MFs contribute to the complexity and difficulty in understanding a rule. The longer the rule and the more MFs it involves, the harder it becomes for humans to process and comprehend [222]. However, the proposed research considers multiple criteria, including these factors as outlined in Table 5.2, for measuring interpretability. In contrast, previous studies mainly focus on the NR and MFs as the primary metrics [217, 218]. Once again, refer to the summary of existing research on interpretability in FRBSs, presented chronologically in Table 5.5.

The previous research has measured interpretability based on different parameters as discussed in Section 5.3.2, but each paper has focused on only a few specific interpretability parameters and has not considered the Semantics of the fuzzy rule partition parameters, which includes completeness, normalization, distinguishability, and complementarity.

As per the existing work in Table 5.5, most of the studies focus on measures such as rule-based complexity and fuzzy partition complexity, which are traditionally used to assess interpretability. The number of rules in rule-based complexity is one of the most frequently used measures in the literature, making it a reliable indicator of complexity at the rule-base level. However, the total number of conditions might offer a more comprehensive measure, as it accounts for both rule length and the number of rules in a single metric. In fuzzy partition complexity, many studies impose limits on the maximum number of MFs allowed, although, in high-dimensional problems, reducing the number of features could be more beneficial.

Within the rule-based semantics quadrant, only a few measures exist, but some studies have introduced promising ones, such as rule consistency and more recent measures like the number of rules fired simultaneously. In the fuzzy partition semantics quadrant, many studies focus on imposing constraints only, but newer absolute or relative

semantic interpretability measures have emerged, which are better suited for optimization processes. For both rule-based and fuzzy partition semantics quadrants, widely accepted measures have yet to be established, though they may emerge with further use, similar to how the number of rules became a standard for rule-based complexity. This research focuses on all the complexity parameters related to interpretability.

Table 5.5. Summary of the current work on the interpretability of FRBSs

Refer- ence	Type	Complexity at the									
		Rule-based level		Fuzzy level partitions		Rule Base level		Fuzzy level partition			
		NR	NC	MF	NF	C	RF	CO	NOR	DIS	COM
[225], [226]	Classification	Y	-	-	-	-	-	-	-	-	-
[227]	Regression			Y				-	-	-	-
[228], [229]	Control			Y				-	-	-	-
[230]	Control					Y		-	-	-	-
[231]	Control						Y	-	-	-	-
[232]	Regression			Y				-	-	-	-
[221]	Control	Y		Y		Y		-	-	-	-
[233]	Regression	Y		Y				-	-	-	-
[234]	Classification	Y	Y					-	-	-	-
[235]	Regression							-	-	-	-
[236]	Classification		Y	Y				-	-	-	-
[237]	Classification		Y	Y				-	-	-	-
[238]	Classification	Y						-	-	-	-
[239]	Regression	Y	Y					-	-	-	-
[240]	Classification		Y	Y				-	-	-	-
[241]	Regression					Y		-	-	-	-
[242]	Classification	Y	Y					-	-	-	-
[243]	Classification				Y			-	-	-	-
[244]	Classification				Y			-	-	-	-
[245]	Classification	Y	Y	Y				-	-	-	-
[246]	Classification	Y	Y					-	-	-	-
[247]	Classification	Y						-	-	-	-
[248]	Classification	Y	Y					-	-	-	-
[249]	Classification	Y			Y			-	-	-	-
[250]	Regression	Y		Y				-	-	-	-

Refer- ence	Type	Complexity at the									
		Rule-based level		Fuzzy level partitions		Rule Base level		Fuzzy level partition			
		NR	NC	MF	NF	C	RF	CO	NOR	DIS	COM
[251]	Regression and Control	Y						-	-	-	-
[252]	Regression	Y						-	-	-	-
[253]	Regression		Y					-	-	-	-
[254]	Control							-	-	-	-
[255]	Classification	Y	Y					-	-	-	-
[256]	Classification	Y				Y		-	-	-	-
[257]	Classification							-	-	-	-
[258]	Classification	Y	Y			Y		-	-	-	-
[259]	Classification	Y	Y	Y		Y		-	-	-	-
[260]	Classification	Y	Y	Y				-	-	-	-
[261]	Regression	Y						-	-	-	-
[262]	Classification	Y	Y			Y		-	-	-	-
[263]	Regression							-	-	-	-
[264]	Regression	Y						-	-	-	-
[265]	Classification and Regression		Y					-	-	-	-
[266]	Classification		Y				Y	-	-	-	-
[267]	Classification	Y	Y		Y	Y		-	-	-	-
[268]	Regression	Y						-	-	-	-
[269]	Regression	Y				Y		-	-	-	-
[270]	Classification							-	-	-	-

5.5. Methodology

5.5.1. Overview of the LT_FRBS Approach

In this work, the author has proposed a methodology to measure the accuracy, and interpretability and minimize the trade-off between the accuracy and interpretability of FRBS. To achieve this, the learning and tuning algorithm of FRBS is designed. The researchers have worked on this approach because doing a literature review, they found that while working on FRBS systems, most of the authors have improved either accuracy or interpretability. None of them have done a balancing of these two factors. Here, the authors have utilized the optimizing property of GA and PSO to learn and

tune the MF to enhance the systems accuracy as well as designed the rules and used it in such a way as to improve the interpretability of the system.

5.5.2. *Abstract view of LT_FRBS*

A methodology called LT_FRBS is proposed for tuning the MFs of the FRBS system thus optimally balancing interpretability and accuracy. The research used the real-number representation during the tuning process, utilizing Real-Coded Genetic Algorithms (RCGA) and Particle Swarm Optimization (PSO). RCGA is particularly well-suited for optimization problems in continuous domains, as it allows continuous parameters to gradually adapt to the fitness landscape across the entire search space compared to binary encoding in traditional GAs. Such systems are called Genetic Fuzzy Systems (GFS). Additionally, PSO, offers an alternative evolutionary strategy that effectively explores and exploits the search space for optimal solutions. Figure 5.3 shows the flow of LT_FRBS.

5.5.3. *Method and Algorithm of LT_FRBS*

This sub-section covers the implementation details of the LT_FRBS algorithm. Two algorithms have been proposed to perform tuning of membership functions. The first algorithm LT_FRBSGA tunes the membership functions using the Genetic Algorithm and the second algorithm LT_FRBSPSO tunes the membership functions using PSO.

Initially, a dataset X is considered, and scaling is done to normalize (called normalized dataset. X_{norm}) it in a range of $[0, 1]$ using the mathematical formula

$$X_{norm} = \frac{X[i, j] - \min(X_j)}{\max(X_j) - \min(X_j)} \quad (5.4)$$

where, $X[i, j]$ denotes the i^{th} row and j^{th} column value of the dataset X , $\min(X_j)$ denotes the minimum value of the j^{th} column, and $\max(X_j)$ denotes the largest value of the j^{th} column of dataset X .

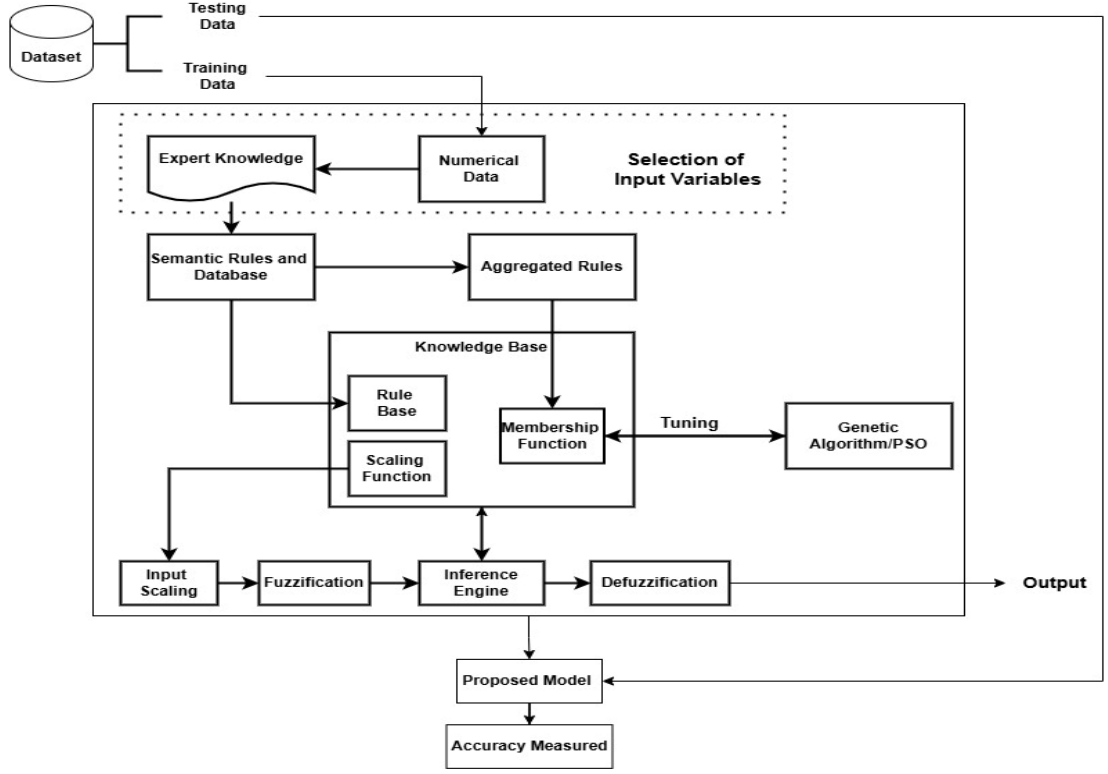


Figure 5.3: Flow of the proposed algorithm LT_FRBS

Figure 5.4 represents the equation and the membership values of the simple triangular membership function for input x at values a , b and c .

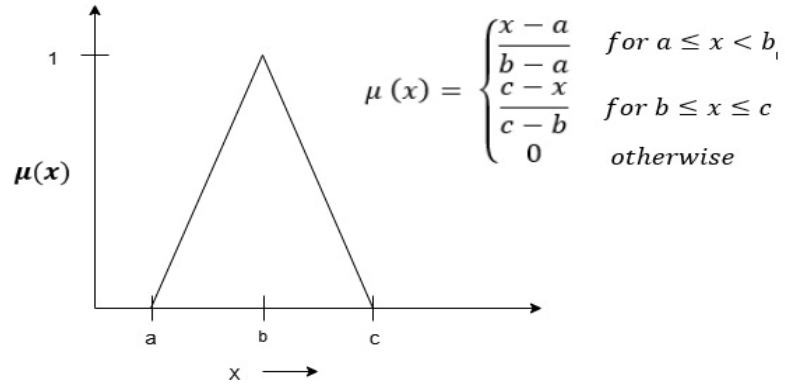


Figure 5.4. Triangular membership function

Triangular membership function of normalized dataset X_{norm} is defined for each feature. Thus, each feature *weight* w , is defined by three linguistic terms small (s), medium (m), and large (l). A triangular membership function is defined against each linguistic variable in such a way that at least one value of each MF should be unity. For

small MF, the value of $a=0.0$, $b=0.0$ and $c=w$, for medium MF, the value of $a=0.0$, $b=w$ and $c = 1.0$ and for large MF, the value of $a = 0.0$, $b = 1.0$ and $c = 1.0$ respectively. Thus, Figure 5.5 represents the membership values for each feature *weight w*:

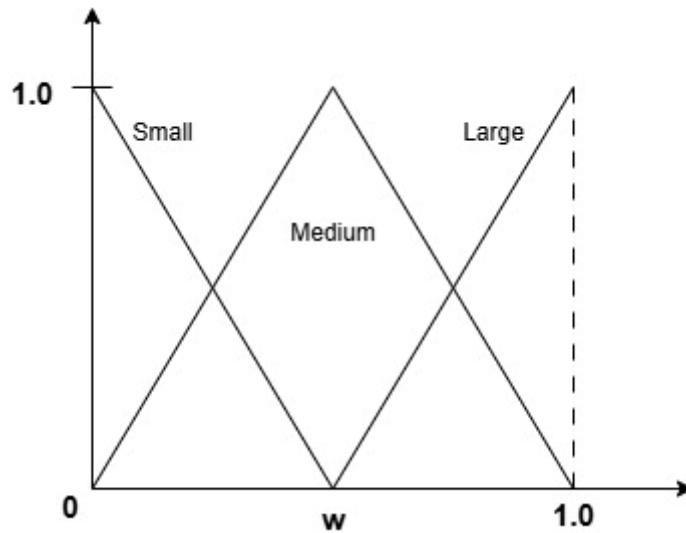


Figure 5.5. Triangular membership function for each feature *weight w*

Thus, maintaining a list of membership values for each normalized dataset feature *weight w* as small, medium, and large membership functions. Figure 5.6 shows that considering a feature in a dataset with a weight w , the membership values for small are $S_1 \dots S_n$ depends on the n values (rows) of that feature. Similarly, medium and large values are stored and other feature values are stored.

<i>weight w</i>		
Small (s)	Medium (m)	Large (l)
$S_1 \dots S_n$	$M_1 \dots M_n$	$L_1 \dots L_n$

Figure 5.6 Membership value for any feature with *weight w*

As already mentioned in Chapter 1, Mamdani FRBS consists of four components: KB, a Fuzzification Interface, an Inference System, and a Defuzzification Interface. The KB is composed of two main components: the Rule Base (RB), which consists of a set of linguistic If-Then rules, and the Database (DB), which defines the fuzzy sets related to the linguistic terms used in the rules. The DB also includes the membership functions (MFs) that define the semantics of these linguistic labels. Each linguistic variable

involved in the problem is associated with a fuzzy partition of its domain, representing the fuzzy sets linked to its respective linguistic terms. The rules are generated automatically using expert knowledge. The Guaje tool is used to generate the rules. The rules with similar consequents are combined using DNF. Thus the rule structure involves the linguistic variables using fuzzy maximum and minimum operator. Thus, linguistic rules possess the following form:

$$\begin{aligned} \text{If } X_1 \text{ is } \{A_{11} \text{ or } A_{13}\} \text{ and } X_2 \text{ is } \{A_{23} \text{ or } A_{25}\} \text{ and } X_3 \text{ is } \{A_{31} \text{ and } A_{32}\} \text{ then} \\ Y \text{ is } B_2 \end{aligned} \quad (5.5)$$

as already discussed in Chapter 1.

The tuning procedure aims to optimize the performance of the FRBS. During this process, a predefined RB and an initial DB are used with the objective to identify a better set of parameters that define DB, improving the system's overall performance. The author used two methodologies for tuning: the first involves using GA, and the second utilizes PSO. Both methods aim to optimize the parameters of the Database (DB) to enhance the performance of the Fuzzy Rule-Based System (FRBS).

The MF tuning is done through GA and PSO to obtain the optimized parameter sets to enhance the performance of FRBS. A priori knowledge of linguistic variables, fuzzy MF parameters, fuzzy rules, number of rules, etc. is available for the genetic design process. In this study, *weights w* contains the parameters of the fuzzy MFs, which are crucial in determining the performance of the fuzzy logic system. These weights influence how the membership functions are defined for each feature of the dataset. They determine where the s, m, and l categories start and end within the normalized range [0, 1]. Triangular MF functions with real values 0, w, and 1.0.

The process begins by generating the MF parameters, which are linked together to form a long string (chromosomes) that represents the complete parameter set of the MF. A fitness function is then defined to assess the performance of each MF set. Subsequently, the algorithm applies crossover, mutation, and reproduction operators to evolve and find the optimal MF.

The fitness function is defined to evaluate how well a given set of weights performs. To evaluate the fitness function, first accuracy of each individual is evaluated. The fitness function converts the accuracy into a fitness score where a higher fitness score

represents better performance. The final output is the best weights found by the GA and their corresponding fitness score.

5.5.3.1. Tuning of FRBS through GA

The Pittsburgh learning approach is employed to fine-tune the membership functions (MFs) using GA. In this method, each chromosome represents a complete set of classifiers. After calculating the fitness of each individual, the GA evaluates their performance by using the fitness function. This process involves applying the fitness function to assess the individuals' weights.

The GA uses the fitness values to select individual selection, perform crossover to create new individuals, and apply mutation to introduce variability. Throughout the evolution process, the GA tracks the best individual and their fitness score. This is done by comparing the fitness of each newly generated individual against the best individuals.

For selection., tournament selection is used, which randomly selects two parents from the population, and fitness is already calculated, then it is used. Otherwise, fitness is calculated using the fitness function. And the parent having better fitness is selected for further operations. After selection, the crossover is done to produce offspring from the selected parents. Crossover is a key genetic operator that enhances the quality of solutions. As RCGA is used, so for crossover, the authors used BLX – alpha crossover.

BLX-Alpha crossover:

It is designed to create offspring that are between the parents, but with some blending that extends the search space beyond the parents' values, using a blending factor (alpha). The parameter alpha determines how far the offspring can "move" outside the range of the parent's values. Typically, alpha is a small positive value (e.g., 0.1, 0.5, or 0.9). Larger values of alpha allow offspring to be farther from the parents, while smaller values make the offspring more similar to the parents. It considers the offspring gene x'_i to be located outside the interval $[x_i^1, x_i^2]$. x'_i is a randomly, uniformly chosen number from the interval

$$x_{min} - I * alpha, x_{max} + I * alpha \quad (5.6)$$

where, $x_{max} = \max(x_i^1, x_i^2)$, $x_{min} = \min(x_i^1, x_i^2)$, and $I = x_{max} - x_{min}$.

Applied mutation to some of the offspring with a certain probability and then evaluated the fitness of the offspring using the fitness function. Finally, output with the best weights is found by the GA and their corresponding fitness score is calculated. The following algorithm 2 covers tuning of MF through GA.

Algorithm 2: LT_FRBS (Tuning through GA)

1. Initialize GA parameters
For each epoch, perform the following steps
 - a. Select parents using tournament selection.
 - b. Perform crossover and mutation to produce offspring.
 - c. Assess the fitness of the population.
 - d. Update best individual and its fitness.
 2. Run GA multiple times to collect performance data.
-

The following parameters are used while working with GA are shown in Table 5.6:

Table 5.6. Parameters for GA

Parameters	Values
Population Size	10
Crossover Probability	0.9
Mutation Rate	0.1
Selection	Tournament
Crossover Operator	BLX-alpha
Mutation Method	Random

The implemented GA adopts tournament selection, BLX-alpha crossover, and random mutation to balance exploration and exploitation during optimization. These operators accelerate convergence by guiding the population toward promising regions of the search space, while mutation maintains diversity to prevent premature stagnation in local optima. The use of a moderate population size and limited epochs ensures computational efficiency without compromising solution quality.

5.5.3.2. Tuning of FRBS through PSO

In the proposed model, the triangular membership function is used. PSO algorithm is specified to adjust the shapes of membership functions by tuning the parameters of membership functions. After calculating the fitness, PSO is used for tuning the parameters of MF to get the g_best and p_best values. The following algorithm 3 covers tuning of FRBS through PSO.

Algorithm 3: LT_FRBS (Tuning through PSO)

1. Initialize PSO parameters, by setting all constant to 0.5, search boundaries for the particles in the interval $[0, 1]$, inertia constant $w =$, constants c_1 and c_2 to 1, initial best position to None and best error position to -1.
For each iteration, perform the following steps
 - a. Assess the fitness of each particle
 - b. Update the global best position and error
 - c. Update the velocity and position of each particle within bounds
 - d. Recors the best parameters and fitness
 2. Run PSO multiple times to collect performance data.
-

The flowchart of optimizing MF through PSO is shown in Figure 5.7. The PSO-based tuning employs an inertia weight ($w = 0.5$) and acceleration coefficients ($c_1 = 1$, $c_2 = 1$) to achieve an effective balance between exploration and exploitation. Random coefficients (r_1 , r_2) introduce stochastic behavior that helps the swarm escape local optima. These parameter settings were chosen to provide fast convergence while preserving global search capability across complex solution spaces.

5.6. Experimental Evaluation and Comparative Analysis

5.6.1. Experimental Setup

The proposed methodology is evaluated on datasets. The datasets considered are already explained in Chapter 3 Section 3.5.1.1. The motive of this approach is to calculate the accuracy, and interpretability of the 10 parameters considered. Also, this proposed algorithm tries to optimize the trade-off between interpretability and accuracy. Two tuning approaches GA and PSO are used to obtain the results (which are named as LT_FRBS Algorithm 2 and 3, but in short and in the future called LT_FRBS)

5.6.1.1.Datasets

The datasets considered are Pima Indian Diabetes, Spectfheart, Breast Cancer, Parkinson, PCOS dataset, and Cleaveland dataset whose details are covered in Section 3.5.1.1.

5.6.1.2.Performance Metrics

To calculate the performance of these datasets on the LT_FRBS, we considered the Accuracy and Interpretability. Briefings of these evaluation metrics are shown below:

- Accuracy: It defines how the model predictions match with the actual outputs:

$$Accuracy = \frac{\text{Number of Correct Predictions}}{\text{Total number of predictions}} \quad (5.7)$$

- Interpretability Measures:

- | | |
|------------------------------------|----------------------------------|
| a. Number of Rules | f. Numbers of rules fired at the |
| b. Number of Conditions | same time. |
| c. Number of membership | g. Completeness or coverage |
| Functions | h. Normalization |
| d. Number of features or variables | i. Distinguishability |
| e. Consistency of the Rule Base | j. Complementarity |

5.6.1.3.Comparison Techniques

After designing the algorithm, the proposed algorithm was used to derive the accuracy as well as interpretability of the datasets considered for evaluation. The results of the LT_FRBS Algorithm 2 and 3 are evaluated in four ways:

- First, the accuracy of each dataset is calculated using LT_FRBS.
- Second, the interpretability of the datasets is evaluated on different interpretability measures.
- At last, a comparative study of LT_FRBS with the existing methods.
- Additionally, a light a thrown on how trade-off is minimized.

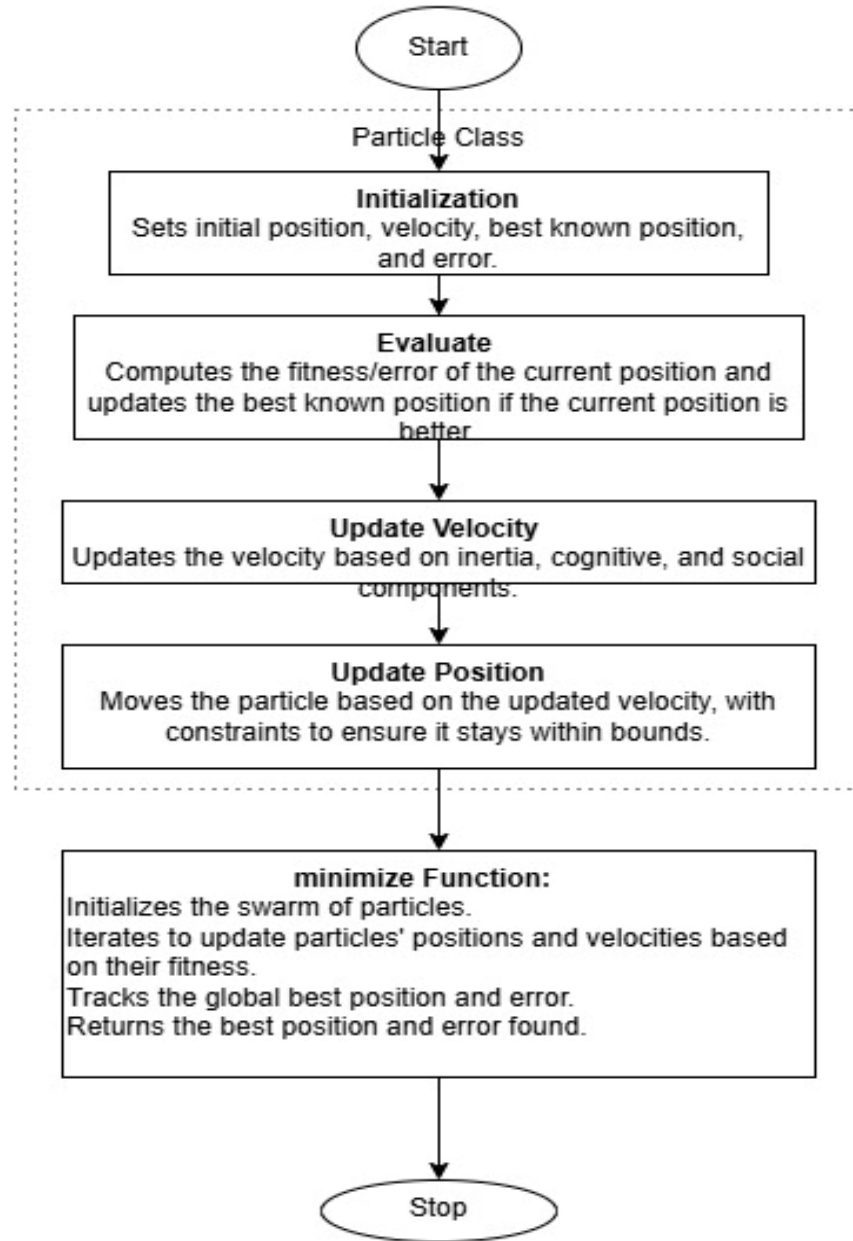


Figure 5.7. Flowchart for tuning of MF using PSO

5.6.2. Results and Discussions

The performance of the LT_FRBS algorithms is evaluated on the above-mentioned datasets. The results are reported in the following sections:

5.6.2.1. Accuracy Measurement on Different Datasets

The accuracy of the LT_FRBS algorithms on real-world datasets is shown in Table 5.7.

Table 5.7. Accuracy of the LT_FRBS Algorithm

Dataset	LT_FRBS using GA (Accuracy)				LT_FRBS using PSO (Accuracy)			
	Maximum	Minimum	Mean	SD	Maximum	Minimum	Mean	SD
Pima	76.692	76.692	76.692	0..000	76.95	76.95	76.95	0.00
Spectfheart	80.14	79.026	79.588	0.0056	79.40	78.65	79.02	0..0037
Breast Cancer	91.91	91.73	91.82	.0008	92.44	91.56	92.00	.0043
Parkinson	66.66	49.74	58.20	0.0846	72.30	69.23	70.76	.0153
PCOS	76.692	76.692	76.692	0..000	76.89	68.02	72.45	0.44
Cleveland	53.87	53.87	53.87	0	53.87	53.87	53.87	0

Table 5.7 shows the accuracy of the algorithm using GA as well as PSO. The result shows the maximum accuracy achieved as well as minimum accuracy and mean on all six datasets.

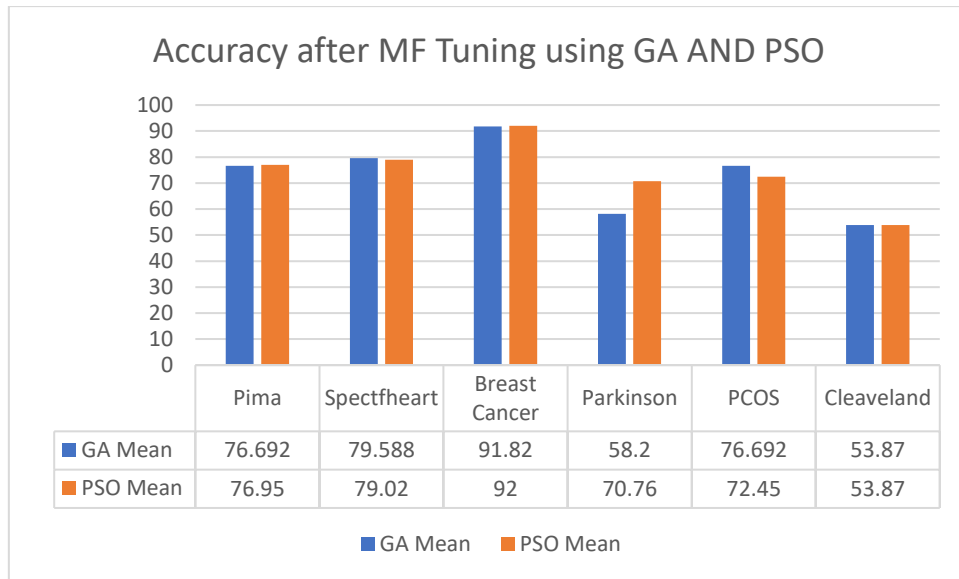


Figure 5.8. Mean Accuracy of each dataset on the proposed algorithm after tuning with GA and PSO

The results show that the maximum accuracy achieved is 92.44% on the breast cancer dataset after tuning of MF using PSO, but almost near to it (91.91%) is achieved on the same dataset using tuning of MF through GA. Four out of six datasets achieved accuracy above 75% after tuning with both GA and PSO except Parkinson and Cleveland. Seeing the accuracy of Parkinson's through PSO is 72.30% approximately closer to 75%. Only the Cleveland dataset achieved accuracy smaller through both GA

and PSO is 53.87%. Figure 5.8 shows the comparison of mean accuracy through tuning with GA as well as PSO.

5.6.2.2. Interpretability measurement on different datasets

This section presents the interpretability of the proposed work on the different interpretability metrics covered in Table 5.2. Here, all the ten interpretability metrics are measured on each dataset considered with an explanation.

PIMA Dataset: The original PIMA Indian Diabetes Dataset has 9 features. After feature selection got 4 features namely Pregnancies, Glucose, BMI, Age, Class where Class is a target. Table 5.8 shows the interpretability measures.

Table 5.8. Interpretability measures of PIMA dataset

Metric	Value	Explanation
Rule Based Complexity		
Number of Rules	2	Two main rules for "diabetes" and "no diabetes" outcomes.
Conditions per rule	4	Each rule evaluates the 4 selected features.
Fuzzy partition Complexity		
Number of Features	4	After feature selection we received
Total Number of MFs	12	4 features with 3 membership functions each.
Rule-based Semantics		
Consistency of the Rule Base	True	Rules do not contradict each other; they evaluate "diabetes" or "no diabetes."
The number of rules fired at the same time.	1	One rule fires at any given time.
Fuzzy partition Semantics		
Completeness or coverage	True	Membership functions cover the full input range for each feature.
Normalization	True	Triangular membership functions peak at 1.0
Distinguishability	True	Membership functions are distinct across each feature.
Complementarity	True	The system covers the feature range comprehensively with complementary MFs.

Spectfheart dataset: The original Spectfheart dataset has 45 features. after feature selection got 6 features namely F3R, F9R, F17R, F18S, F20S, F22S, OVERALL_DIAGNOSIS. where OVERALL_DIAGNOSIS is the target. Table 5.9 shows the Interpretability measures of the Spectfheart dataset.

Table 5.9. Interpretability measures of the Spectfheart dataset

Metric	Value	Explanation
Rule Based Complexity		
Number of Rules	2	Two rules for predicting "abnormal" and "normal" outcomes.
Conditions per rule	6	Each rule checks each of the 6 features.
Fuzzy partition Complexity		
Number of Features	6	After feature selection we received
Total Number of MFs	18	6 features with 3 membership functions each
Rule-based Semantics		
Consistency of the Rule Base	True	Rules do not contradict each other and are independent.
The number of rules fired at the same time.	1	Only one rule can be activated at a time.
Fuzzly partition Semantics		
Completeness or coverage	True	Membership functions cover the entire input space for each feature.
Normalization	True	Triangular membership functions peak at 1.0
Distinguishability	True	Membership functions are distinct across the features.
Complementarity	True	Comprehensive coverage across the range of values for each feature.

Breast Cancer dataset: The original Breast Cancer Dataset has 33 features. After feature selection we got 12 features namely texture_mean, perimeter_mean, area_mean, concavity_mean, concave points_mean, fractal_dimension_mean, radius_se, texture_se, area_se, compactness_se, concavity_se, concave points_se, diagnosis where diagnosis is target. Table 5.10 shows its results.

Table 5.10. Interpretability measures of the Breast Cancer dataset

Metric	Value	Explanation
Rule Based Complexity		

Number of Rules	2	Two main rules for benign and malignant
Conditions per rule	12	Each rule evaluates all 12 selected features
Fuzzy partition Complexity		
Number of Features	12	After feature selection, 12 features remain
Total Number of MFs	36	12 features with 3 membership functions each
Rule-based Semantics		
Consistency of the Rule Base	True	The two rules are distinct and non-contradictory
The number of rules fired at the same time.	1	Only one rule fires per classification instance.
Fuzzy partition Semantics		
Completeness or coverage	True	Membership functions cover the full normalized range for each feature
Normalization	True	Triangular membership functions peak at 1.0
Distinguishability	True	Membership functions for each feature are distinct
Complementarity	True	MFs provide complete and complementary coverage across feature ranges

Parkinson dataset: Parkinson dataset originally have 24 features and after applying feature selection using MISFS achieved 8 features namely MDVP: Fhi(Hz), MDVP: Flo(Hz), MDVP: Jitter(%), NHR, HNR, DFA, spread2, PPE, status. Here status is the target. Its result is shown in Table 5.11.

Table 5.11. Interpretability measures of Parkinson dataset

Metric	Value	Explanation
Rule Based Complexity		
Number of Rules	2	Two main rules
Conditions per rule	8	Each rule evaluates all 8 selected features
Fuzzy partition Complexity		
Number of Features	8	Eight features are selected after feature selection
Total Number of MFs	24	8 features $\times$ 3 membership functions per feature.
Rule-based Semantics		
Consistency of the Rule Base	True	Non-contradictory rules predicting "Parkinson" vs. "Healthy".
The number of rules are fired at the same time.	1	Only one rule can be activated at a time.
Fuzzy partition Semantics		

Completeness or coverage	True	Membership functions cover the entire input space.
Normalization	True	Triangular membership functions peak at 1.0
Distinguishability	True	Membership functions have distinct intervals.
Complementarity	True	Comprehensive coverage across each feature's value range.

PCOS Dataset: The PCOS dataset originally had 45 features and after applying feature selection using MISFS achieved 15 features.

Table 5.12: Interpretability measures of the PCOS dataset

Metric	Value	Explanation
Rule Based Complexity		
Number of Rules	2	Two main rules for "PCOS" and "no PCOS" outcomes.
Conditions per rule	15	Each rule checks each of the 15 features.
Fuzzy partition Complexity		
Number of Features	15	The selected subset of features in the PCOS dataset.
Total Number of MFs	45	15 features with 3 MFs each.
Rule-based Semantics		
Consistency of the Rule Base	True	Rules are non-contradictory and separate between "PCOS" and "no PCOS" classes.
Number of rules are fired at the same time.	1	Only one rule is activated at a time.
Fuzzy partition Semantics		
Completeness or coverage	True	The MFs cover the entire universe of discourse for each feature.
Normalization	True	Triangular membership functions peak at 1.0
Distinguishability	True	The three MFs (s, m, l) for each feature are distinct.
Complementarity	True	The MFs provide complementary coverage across each feature's range.

Cleveland dataset: The Cleveland dataset originally had 14 features and after feature selection using MISFS we received 6 features. Results are shown in Table 5.13.

Table 5.13. Interpretability measures of the Cleveland dataset

Metric	Value	Explanation
Rule Based Complexity		
Number of Rules	5	Counted based on defined rules in the fuzzy system, representing 5 levels of disease severity.

Conditions per rule	4.8	Average number of conditions across rules.
Fuzzy partition Complexity		
Number of Features	6	The dataset uses 6 features (selected for this model).
Total Number of MFs	18	Each feature has 3 MFs (small, medium, large); $6 \times 3 = 18$ MFs.
Rule-based Semantics		
Consistency of the Rule Base	True	No contradictory rules, as each disease severity is mutually exclusive.
The number of rules fired at the same time.	1	At any evaluation point, only 1 rule (severity level) is active.
Fuzzy partition Semantics		
Completeness or coverage	True	Membership functions for each feature cover the universe (0 to 1) entirely.
Normalization	True	Triangular membership functions peak at 1.0
Distinguishability	True	MFs are distinct with minimal overlap beyond expected bounds.
Complementarity	True	The MFs for each feature complement each other, ensuring full coverage of the feature space.

Table 5.8 – 5.13 shows the interpretability result of the datasets PIMA, Spectfheart, Breast Cancer, Parkinson, PCOS, and Cleaveland. Summarized results of interpretability are covered in Table 5.13.

Table 5.14. Interpretability results of all datasets through the proposed LT_FRBS algorithm

Metric	Cleveland	PCOS	Parkinson	Spectf heart	PIMA	Breast Cancer
Number of Rules	5	2	2	2	2	2
Conditions per rule	6	15	8	6	4	12
Number of Features	6	15	8	6	4	12
Total Number of MFs	18	45	24	18	12	36
Consistency of the Rule Base	Yes	Yes	Yes	Yes	Yes	Yes
Numbers of rules fired at the same time	1	1	1	1	1	1
Completeness or coverage	Yes	Yes	Yes	Yes	Yes	Yes
Normalization	Yes	Yes	Yes	Yes	Yes	Yes

Distinguishability	Yes	Yes	Yes	Yes	Yes	Yes
Complementarity	Yes	Yes	Yes	Yes	Yes	Yes

The results show that the interpretability of all datasets is achieved at a higher rate. The number of rules for most of the datasets is 2 except for the Cleaveland dataset, whose number of rules is 5 as it is the multidimensional dataset. The NC per rule (complexity of the rule antecedents) is below 10 on all datasets except for breast cancer. As is already mentioned in Table 5.2, the requirements needed for the system to be interpretable. Considering the case of the number of MF needed for reducing the interpretability complexity is 7 ± 2 per feature as mentioned in Table 5.2. Considering the interpretability result of Cleaveland the NF is 6 and the MFs is 18. It means 3 MFs for each feature showing the reduction in the interpretability. Similarly, the case for all metrics of interpretability is below the requirements mentioned in Table 5.2 for reducing the complexity.

5.6.2.3. Comparative analysis with existing algorithms

This section provides comparative analysis tables (5.15 – 5.20) of the proposed LT_FRBS optimization approach with existing algorithms, focussing on accuracy, interpretability and convergence behaviour. In this context, the accuracy and interpretability of the LT_FRBS algorithm are compared to existing algorithms. As previously mentioned, the interpretability of a fuzzy model depends on several factors. Traditionally, most optimization techniques tend to improve performance by adjusting parameters (i.e., antecedents) based on the training dataset, which often results in a less interpretable model. However, Fuzzy Logic Systems offer mechanisms that preserve interpretability. Modifying these parameters in a data-driven manner can compromise the model's interpretability, potentially leading to issues such as a loss of complementarity, coverage, or distinguishability of fuzzy sets across the universe of discourse, which undermines the meaningfulness of the fuzzy sets used. By learning and tuning parameters during the fuzzification process, it is possible to maintain both interpretability and performance benefits.

Regarding the rule generation in the experiments, while different approaches have been introduced, this work follows the well-established Wang-Mendel [194] rule generation

technique. In this section, the accuracy and interpretability of LT_FRBS, on each dataset, are compared with the the existing work.

Table 5.15. Comparison of LT_FRBS with existing work on PIMA Indian Diabetes

Pima Indian Diabetes Dataset			
Authors	Techniques	Accuracy	Interpretability
[271]	Single granularity-based fuzzy classification	74.92	NR = 6.63, NC = 74.92
[272]	NSGA-II, new complexity-related interpretability measure	81.5	NR = 6.2, Nattri =3, NF = 5 Nattri/rule = 1.2
[273]	IRL algorithm	75%	NR = 4.6, NC= 10.6
[274]	Direct method to learn interval-valued fuzzy partitions	76.69%	RB = 14.59 Antecedent per rule =2.09
LT_FRBS	GA and PSO	76.95	NR = 2, NC =4, MF = 12, NF=4, C=TRUE, RF= 1, CO = TRUE, NOR=TRUE, DIS=TRUE, COM= TRUE

Table 5.16. Comparison of LT_FRBS with existing work on Spectfheart

Spectfheart Dataset			
Authors	Techniques	Accuracy	Interpretability
[275]	ANFIS, DENFIS, SOFNN	70%	NR = 246, MF = 7
[273]	IRL algorithm	79.4%	NR = 1, NC= 12.7
[274]	Direct method to learn interval-valued fuzzy partitions	80.52%	RB Size = 14.59 Antecedent per rule =2.09
LT_FRBS	GA and PSO	80.14	NR = 2, NC =6, MF = 18, NF=6, C=TRUE, RF= 1, CO = TRUE, NOR=TRUE, DIS=TRUE, COM= TRUE

Table 5.17. Comparison of LT_FRBS with existing work on breast cancer

Breast Cancer Dataset			
Authors	Techniques	Accuracy	Interpretability
[273]	IRL algorithm	73%	NR = 4.6, NC = 9.3
[276]	Hierarchical fuzzy rule structure	96.1	NR = 6
LT_FRBS	GA and PSO	92.44	NR = 2, NC =12, MF = 36, NF=12, C=TRUE, RF= 1,

			CO = TRUE, NOR=TRUE, DIS=TRUE, COM= TRUE
--	--	--	---

Table 5.18. Comparison of LT_FRBS with existing work on Parkinson

Parkinson Dataset			
Authors	Techniques	Accuracy	Interpretability
[275]	ANFIS, DENFIS, SOFNN	76%	NR = 100, MF=3
[277]	Fuzzy Inference Model	-	NR = 21
[278]	Fuzzy Inference Model	-	NR = 47
[279]	Fuzzy Inference Model	-	NR = 28
LT_FRBS	GA and PSO	72.30	NR = 2, NC =8, MF = 24, NF=8, C=TRUE, RF= 1, CO = TRUE, NOR=TRUE, DIS=TRUE, COM= TRUE

Table 5.19. Comparison of LT_FRBS with existing work on PCOS

PCOS Dataset			
Authors	Techniques	Accuracy	Interpretability
[179]	Red Deer Algorithm with RF	89.81	-
[180]	LR with cross-validation	85.022	-
[181]	KNN, SVM	90.83	-
[180]	LR	83	-
LT_FRBS	GA and PSO	76.89	NR = 2, NC =15, MF = 46, NF=15, C=TRUE, RF= 1, CO = TRUE, NOR=TRUE, DIS=TRUE, COM= TRUE

Table 5.20. Comparison of LT_FRBS with existing work on Cleaveland

Cleaveland Dataset			
Authors	Techniques	Accuracy	Interpretability
[271]	Single granularity-based fuzzy classification	55.11	NR = 20.17, NC = 56.1
[273]	IRL algorithm	56.10	NR = 8.5, NC = 10.6
[274]	Direct method to learn interval-valued fuzzy partitions	59.26	RB Size = 14.59, Antecedent per Rule = 2.09

LT_FRBS	GA and PSO	53.87	NR = 6, NC =5, MF = 18, NF=6, C=TRUE, RF= 1, CO = TRUE, NOR=TRUE, DIS=TRUE, COM= TRUE
---------	------------	-------	--

The results indicate that most existing studies focused on calculating accuracy for each dataset, but when measuring interpretability, they primarily considered the number of rules and conditions, which largely emphasized the rules themselves rather than the membership functions. In contrast, the proposed approach evaluates all parameters that contribute to interpretability, including both rule-based aspects and membership functions. Since the proposed work emphasizes the tuning of membership functions, it is essential to include parameters related to these functions in the assessment of interpretability. From Table 5.14, the interpretability is measured on all metrics and all the results are better from the existing work in Table 5.15– 5.20.

5.6.2.4. *Is the trade-off between interpretability and accuracy minimized?*

Yes, the results of interpretability and accuracy shows that the trade - off between both appears to be well-managed in this case, although it is not completely eliminated, but is reduced to its minimum level. To understand it we have to view it from the eye of interpretability and accuracy.

- **Interpretability Perspective:** It is seen from the above results that model designed helps in keeping the complexity of the system relatively low. In each dataset we see that the number of rules vary from 2 to 5 only, which is very interpretable. Moreover, we are using 3 membership functions (MFs) per feature, which is within the human cognitive limit of 7 ± 2 and enhances understandability. The semantics of the fuzzy system are preserved: the membership functions (MFs) are normalized, distinguishable, and provide full coverage of the input space. These factors make the model easy to understand and explainable.
- **Accuracy Perspective:** The accuracy of the most of the datasets is above 75% using this model, which is relatively good, with GA achieving maximum accuracy up to 91.91% accuracy and PSO achieving up to 92.44%. This shows that despite

maintaining a high level of interpretability, the model still performs well on the classification task.

- **Interpretability-Accuracy Trade-off:** Generally, increasing interpretability (e.g., reducing the NR or MFs reduces model complexity but often comes at the cost of decreased accuracy. In our case, however, we have managed to balance both Accuracy which is still very good and Interpretability is preserved through all the parameters considered at a higher rate, and no extreme complexity is introduced to achieve high accuracy.

Therefore, it could be considered through our proposed model that the trade-off between interpretability and accuracy is minimized at high level. We achieved relatively simple and interpretable model with high accuracy, meaning that our model doesn't sacrifice much accuracy for the sake of interpretability, and vice versa.

5.6.2.5. Convergence and local optima considerations

Although both GA and PSO successfully optimized the membership functions and fuzzy rule parameters, their convergence time may increase when applied to large datasets or complex fuzzy models. Similarly, PSO may occasionally exhibit premature convergence when particle diversity decreases. To overcome these limitations, future research may explore adaptive strategies such as dynamic inertia weight adjustment, mutation-based diversity maintenance, or hybrid GA–PSO models. Parallel computation can also be integrated to reduce overall runtime for large-scale datasets while maintaining global search capability.

5.7. Summary

Chapter 5 focused on optimizing fuzzy rule-based systems (FRBS) by tuning membership functions (MFs) using Genetic Algorithms (GA) and Particle Swarm Optimization (PSO). Building upon the feature selection performed in Chapter 3 with the MISFS algorithm, the proposed approach constructs fuzzy rules using the DNF/Gauaje tool and then fine-tunes the MFs to enhance classification performance while balancing interpretability.

The methodology maintains a fixed rule base, ensuring that interpretability is not compromised during optimization. GA and PSO were applied to adjust MF parameters, with the performance evaluated on multiple datasets using both accuracy and interpretability metrics. Comparative analysis showed that both GA and PSO effectively improved prediction accuracy, with PSO converging slightly faster in most cases, while GA offered robustness in exploring diverse MF configurations.

The study also addressed two key performance considerations:

1. **Convergence Time:** Population sizes, epoch numbers, and crossover/mutation parameters were carefully selected to ensure efficient convergence. Although both GA and PSO are population-based and stochastic in nature, their execution times increased moderately with the number of MFs and dataset size.
2. **Avoiding Local Optima:** Random mutation in GA and stochastic velocity updates in PSO helped the algorithms escape local optima, ensuring that global search capability was maintained.

Experimental results confirmed that the proposed MF tuning approach reduces the trade-off between accuracy and interpretability, achieving high classification performance without increasing model complexity. However, the study also noted that computational demand grows with the number of MFs and dataset size, suggesting potential for future work on adaptive or hybrid optimization strategies to further accelerate convergence and enhance scalability.

Overall, Chapter 5 demonstrates that GA and PSO-based MF tuning provides a practical and effective method to optimize FRBS performance, complementing the feature selection framework from Chapter 3 and bridging the research objectives of improving both accuracy and interpretability in real-world datasets.

CHAPTER 6

CONCLUSION AND FUTURE DIRECTIONS

6.1. Conclusion and Discussions

This research focused on enhancing the performance and interpretability of fuzzy rule-based systems (FRBS) through efficient feature selection, learning, and optimization mechanisms. The study proposed a comprehensive framework that integrates feature selection using the MISFS algorithm and learning optimization using Genetic Algorithms (GA) and Particle Swarm Optimization (PSO).

The MISFS algorithm effectively reduced data dimensionality by selecting only the most informative features, thereby improving classification accuracy and reducing computational cost. It combined the advantages of mutual information-based filtering and sequential forward selection to address redundancy and irrelevance in high-dimensional datasets.

In the second phase, the research explored and implemented learning and optimization strategies for FRBS. The use of GA and PSO for membership function tuning significantly improved predictive accuracy while maintaining interpretability. The proposed LT_FRBS model demonstrated that a balanced trade-off between interpretability and accuracy is achievable through careful tuning of fuzzy parameters and optimization of rule structures.

Overall, the research achieved the following key outcomes:

1. **Efficient Feature Selection:** The proposed MISFS algorithm combines mutual information-based filtering with sequential forward selection to reduce high-

dimensional datasets while retaining the most relevant features. This enables FRBS to achieve higher predictive accuracy with fewer inputs, improving computational efficiency.

2. **Learning and Optimization of FRBS:** A comprehensive study of learning mechanisms for FRBS was conducted, including ad-hoc covering, neural-network-based, and genetic learning methods. The work highlights how tuning membership functions (MFs) and rule parameters influences both accuracy and interpretability.
3. **Metaheuristic Tuning with GA and PSO:** The thesis demonstrates that Genetic Algorithms (GA) and Particle Swarm Optimization (PSO) can effectively fine-tune MFs, reducing the trade-off between accuracy and interpretability. Experimental evaluation on multiple datasets shows significant improvements over conventional optimization techniques, maintaining model transparency and minimizing complexity.
4. **Interpretability–Accuracy Trade-Off:** By integrating feature selection and MF optimization, the research provides a practical approach to balance interpretability and performance. Rules are kept concise and semantically meaningful, ensuring usability in real-world applications.
5. **Empirical Validation:** The proposed methods are evaluated on multiple real-world datasets, confirming their applicability and generalization capabilities in diverse scenarios.

Overall, this research demonstrates that combining hybrid feature selection, advanced learning mechanisms, and metaheuristic optimization can substantially improve FRBS performance while maintaining interpretability. The framework provides a systematic methodology for constructing, tuning, and validating FRBS in high-dimensional and complex problem domains.

6.2 Future Directions

Future research could expand on the present work by:

1. Exploring additional optimization strategies, such as hybrid GA–PSO or differential evolution, to further reduce convergence time and improve global search capability.
2. Extending evaluation to larger-scale, high-dimensional datasets to assess scalability and computational efficiency.
3. Investigating adaptive fuzzy rule generation and advanced clustering techniques to enhance the interpretability–accuracy balance.
4. Applying the framework to emerging real-world applications, including robotics, autonomous systems, recommendation engines, and healthcare decision support.
5. Developing automated pipelines for tuning both features and fuzzy system parameters, potentially integrating machine learning-based surrogate models for faster optimization.

By pursuing these directions, FRBS can be made more robust, scalable, and widely applicable in complex, uncertain, and real-time environments.

REFERENCES

- [1] L. A. Zadeh, 'Fuzzy sets', *Information and Control*, 1965.
- [2] S. Barro and R. Marín, 'A Call for a Stronger Role for Fuzzy Logic in Medicine', in *Fuzzy Logic in Medicine*, vol. 83, S. Barro and R. Marín, Eds, in *Studies in Fuzziness and Soft Computing*, vol. 83. , Heidelberg: Physica-Verlag HD, 2002, pp. 1–17. doi: 10.1007/978-3-7908-1804-8_1.
- [3] E. H. Mamdani, 'Advances in the linguistic synthesis of fuzzy controllers', *International Journal of Man-Machine Studies*, vol. 8, no. 6, pp. 669–678, 1976.
- [4] V. B. Watkar, 'Fuzzy Inference Systems: Types & Applications', *Journal of Emerging Technologies and Innovative Research (JETIR)*, 2022, Accessed: Oct. 15, 2025. [Online]. Available: <https://www.indiramahavidyalaya.com/profile/pdfdata/naac/RESEARCH%20PAPERS%20FUZZY%20INFERENCE%20SYSTEM%202022-2023.pdf>
- [5] T. Takagi and M. Sugeno, 'Fuzzy identification of systems and its applications to modeling and control', *IEEE transactions on systems, man, and cybernetics*, no. 1, pp. 116–132, 1985.
- [6] M. Sugeno and G. T. Kang, 'Structure identification of fuzzy model', *Fuzzy sets and systems*, vol. 28, no. 1, pp. 15–33, 1988.
- [7] Y. Tsukamoto, 'An approach to fuzzy reasoning method', *Advances in fuzzy set theory and applications*, 1979, Accessed: Jan. 21, 2025. [Online]. Available: <https://cir.nii.ac.jp/crid/1570854175296227328>
- [8] E. H. Mamdani and S. Assilian, 'An experiment in linguistic synthesis with a fuzzy logic controller', *International journal of man-machine studies*, vol. 7, no. 1, pp. 1–13, 1975.
- [9] A. Dvořák and M. Štěpnička, 'Mamdani-Assilian rules: with or without continuity?', in *Proceedings of the 13th International Workshop on Fuzzy Logic and Applications (WILF 2021)*, 2021. Accessed: Oct. 15, 2025. [Online]. Available: <https://ceur-ws.org/Vol-3074/paper06.pdf>
- [10] R. Alcalá, O. Cordon, J. Casillas, F. Herrera, and S. J. I. Zwir, 'Learning and tuning fuzzy rule-based systems for linguistic modeling', in *Knowledge-based systems*, Elsevier, 2000, pp. 889–941. Accessed: Jan. 21, 2025. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/B9780124438750500306>
- [11] P. Lindskog, 'Fuzzy Identification from a Grey Box Modeling Point of View', in *Fuzzy Model Identification*, H. Hellendoorn and D. Driankov, Eds, Berlin, Heidelberg: Springer Berlin Heidelberg, 1997, pp. 3–50. doi: 10.1007/978-3-642-60767-7_1.
- [12] J. Casillas, O. Cordon, F. Herrera, and L. Magdalena, 'Accuracy Improvements to Find the Balance Interpretability-Accuracy in Linguistic Fuzzy Modeling: An Overview', in *Accuracy Improvements in Linguistic Fuzzy Modeling*, vol. 129, J. Casillas, O. Cordon, F. Herrera, and L. Magdalena, Eds, in *Studies in Fuzziness*

- and Soft Computing, vol. 129. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 3–24. doi: 10.1007/978-3-540-37058-1_1.
- [13] Y. You, J. Sun, Y. Guo, Y. Tan, and J. Jiang, ‘Interpretability and accuracy trade-off in the modeling of belief rule-based systems’, *Knowledge-Based Systems*, vol. 236, p. 107491, 2022.
 - [14] A. Ishtaiwi, A. Al-Qerem, and M. A. Rajab, ‘An Efficient and Interpretable Method for Large-Scale Data Classification’, in *2025 1st International Conference on Computational Intelligence Approaches and Applications (ICCIAA)*, IEEE, 2025, pp. 1–7. Accessed: Oct. 15, 2025. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/11013412/>
 - [15] L. A. Zadeh, ‘Outline of a new approach to the analysis of complex systems and decision processes’, *IEEE Transactions on systems, Man, and Cybernetics*, no. 1, pp. 28–44, 1973.
 - [16] I. Guyon and A. Elisseeff, ‘An introduction to variable and feature selection’, *Journal of machine learning research*, vol. 3, no. Mar, pp. 1157–1182, 2003.
 - [17] H. Liu and H. Motoda, *Feature Selection for Knowledge Discovery and Data Mining*. Boston, MA: Springer US, 1998. doi: 10.1007/978-1-4615-5689-3.
 - [18] H. Liu and H. Motoda, *Computational Methods of Feature Selection*. CRC Press, 2007. Accessed: Nov. 07, 2024. [Online]. Available: <http://www.scopus.com/inward/record.url?scp=85130883648&partnerID=8YFLogxK>
 - [19] K. Koutroumbas and S. Theodoridis, *Pattern Recognition*. Academic Press, 2008.
 - [20] S. Kotsiantis, ‘Feature selection for machine learning classification problems: a recent overview’, *Artificial intelligence review*, vol. 42, no. 1, pp. 157–176, 2011.
 - [21] A. R. Webb, *Statistical pattern recognition*. John Wiley & Sons, 2003. Accessed: Jan. 21, 2025. [Online]. Available: https://books.google.com/books?hl=en&lr=&id=ivMBWCe_f0gC&oi=fnd&pg=PR7&dq=statistical+pattern+recognition&ots=HJ-rtcoc8x&sig=ClFxWhTU5W-8o5dJF0pL5xBquW0
 - [22] G. Ritter, *Robust Cluster Analysis and Variable Selection*. CRC Press, 2014.
 - [23] S. K. Pal and P. Mitra, *Pattern recognition algorithms for data mining*. Chapman and Hall/CRC, 2004. Accessed: Jan. 21, 2025. [Online]. Available: <https://www.taylorfrancis.com/books/mono/10.1201/9780203998076/pattern-recognition-algorithms-data-mining-sankar-pal-pabitra-mitra>
 - [24] I. T. Jolliffe, *Principal Component Analysis*. in Springer Series in Statistics. New York, NY: Springer, 1986. doi: 10.1007/978-1-4757-1904-8.
 - [25] J. Ye, ‘Least squares linear discriminant analysis’, in *Proceedings of the 24th international conference on Machine learning*, Corvalis Oregon USA: ACM, June 2007, pp. 1087–1093. doi: 10.1145/1273496.1273633.
 - [26] G. H. Golub and C. Reinsch, ‘Singular Value Decomposition and Least Squares Solutions’, in *Handbook for Automatic Computation*, F. L. Bauer, A. S. Householder, F. W. J. Olver, H. Rutishauser, K. Samelson, and E. Stiefel, Eds, Berlin, Heidelberg: Springer Berlin Heidelberg, 1971, pp. 134–151. doi: 10.1007/978-3-642-86940-2_10.
 - [27] E. Hancer, B. Xue, and M. Zhang, ‘Differential evolution for filter feature selection based on information theory and feature ranking’, *Knowledge-Based Systems*, vol. 140, pp. 103–119, 2018.

- [28] M. Dash and H. Liu, ‘Handling Large Unsupervised Data via Dimensionality Reduction.’, in *1999 ACM SIGMOD workshop on research issues in data mining and knowledge discovery*, 1999.
- [29] B. Xue, M. Zhang, and W. N. Browne, ‘Particle swarm optimization for feature selection in classification: A multi-objective approach’, *IEEE transactions on cybernetics*, vol. 43, no. 6, pp. 1656–1671, 2012.
- [30] H. Liu and L. Yu, ‘Toward integrating feature selection algorithms for classification and clustering’, *IEEE Transactions on knowledge and data engineering*, vol. 17, no. 4, pp. 491–502, 2005.
- [31] S. Alelyani, J. Tang, and H. Liu, ‘Feature Selection for Clustering: A Review’, in *Data Clustering*, 1st edn, C. C. Aggarwal and C. K. Reddy, Eds, Chapman and Hall/CRC, 2018, pp. 29–60. doi: 10.1201/9781315373515-2.
- [32] T. M. Cover, *Elements of information theory*. John Wiley & Sons, 1999.
- [33] F. R. Chung, *Spectral graph theory*, vol. 92. American Mathematical Soc., 1997.
- [34] U. Von Luxburg, ‘A tutorial on spectral clustering’, *Statistics and computing*, vol. 17, pp. 395–416, 2007.
- [35] Z. Zhao and H. Liu, ‘Semi-supervised Feature Selection via Spectral Analysis’, in *Proceedings of the 2007 SIAM International Conference on Data Mining*, Society for Industrial and Applied Mathematics, Apr. 2007, pp. 641–646. doi: 10.1137/1.9781611972771.75.
- [36] L. Zuo, L. Li, and C. Chen, ‘The graph based semi-supervised algorithm with ℓ_1 -regularizer’, *Neurocomputing*, vol. 149, pp. 966–974, Feb. 2015, doi: 10.1016/j.neucom.2014.07.037.
- [37] K. Zhang, L. Lan, J. T. Kwok, S. Vucetic, and B. Parvin, ‘Scaling Up Graph-Based Semisupervised Learning via Prototype Vector Machines’, *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 3, pp. 444–457, Mar. 2015, doi: 10.1109/TNNLS.2014.2315526.
- [38] N. N. Pise and P. Kulkarni, ‘A Survey of Semi-Supervised Learning Methods’, in *2008 International Conference on Computational Intelligence and Security*. doi: 10.1109/CIS.2008.204.
- [39] K. Benabdeslem and M. Hindawi, ‘Efficient Semi-Supervised Feature Selection: Constraint, Relevance, and Redundancy’, *IEEE Trans. on Knowl. and Data Eng.*, vol. 26, no. 5, pp. 1131–1143, May 2014, doi: 10.1109/TKDE.2013.86.
- [40] O. Chapelle, A. Zien, and B. Schölkopf, Eds, *Semi-supervised learning*. in Adaptive computation and machine learning series. Cambridge, Massachusetts: MIT Press, 2006.
- [41] M. A. Zare Chahooki and N. Moghadam Charkari, ‘Unsupervised manifold learning based on multiple feature spaces’, *Machine Vision and Applications*, vol. 25, pp. 1053–1065, May 2014, doi: 10.1007/s00138-014-0604-7.
- [42] M. A. Z. Chahooki and N. M. Charkari, ‘Improvement of supervised shape retrieval by learning the manifold space’, in *2011 7th Iranian Conference on Machine Vision and Image Processing*, IEEE, 2011, pp. 1–4.
- [43] A. Kumar and A. Kaur, ‘Literature Review on Development of Feature Selection and Learning Mechanism for Fuzzy Rule-Based System’, *Recent Advances in Computer Science and Communications (Formerly: Recent Patents on Computer Science)*, vol. 16, no. 4, pp. 42–62, 2023.

- [44] J. Zhao, K. Lu, and X. He, ‘Locality sensitive semi-supervised feature selection’, *Neurocomputing*, vol. 71, no. 10, pp. 1842–1849, June 2008, doi: 10.1016/j.neucom.2007.06.014.
- [45] H. Cheng, W. Deng, C. Fu, Y. Wang, and Z. Qin, ‘Graph-Based Semi-supervised Feature Selection with Application to Automatic Spam Image Identification’, in *Computer Science for Environmental Engineering and EcoInformatics*, Y. Yu, Z. Yu, and J. Zhao, Eds, Berlin, Heidelberg: Springer, 2011, pp. 259–264. doi: 10.1007/978-3-642-22691-5_45.
- [46] G. Doquire and M. Verleysen, ‘Graph Laplacian for Semi-supervised Feature Selection in Regression Problems’, in *Advances in Computational Intelligence*, J. Cabestany, I. Rojas, and G. Joya, Eds, Berlin, Heidelberg: Springer, 2011, pp. 248–255. doi: 10.1007/978-3-642-21501-8_31.
- [47] G. Doquire and M. Verleysen, ‘A graph Laplacian based approach to semi-supervised feature selection for regression problems’, *Neurocomputing*, vol. 121, pp. 5–13, Dec. 2013, doi: 10.1016/j.neucom.2012.10.028.
- [48] M. Yang, Y.-J. Chen, and G.-L. Ji, ‘Semi_Fisher Score: A semi-supervised method for feature selection’, in *2010 International Conference on Machine Learning and Cybernetics*, July 2010, pp. 527–532. doi: 10.1109/ICMLC.2010.5581007.
- [49] K. Benabdeslem and M. Hindawi, ‘Constrained Laplacian Score for Semi-supervised Feature Selection’, in *Machine Learning and Knowledge Discovery in Databases*, D. Gunopulos, T. Hofmann, D. Malerba, and M. Vazirgiannis, Eds, Berlin, Heidelberg: Springer, 2011, pp. 204–218. doi: 10.1007/978-3-642-23780-5_23.
- [50] M. Kalakech, P. Biela, L. Macaire, and D. Hamad, ‘Constraint scores for semi-supervised feature selection: A comparative study’, *Pattern Recognition Letters*, vol. 32, no. 5, pp. 656–665, Apr. 2011, doi: 10.1016/j.patrec.2010.12.014.
- [51] L. Chen, R. Huang, and W. Huang, ‘Graph-based semi-supervised weighted band selection for classification of hyperspectral data’, *Proc. Int. Conf. Audio Lang. Image Process.*, Nov. 2010, doi: 10.1109/ICALIP.2010.5685086.
- [52] R. Chen, X. Weng, B. He, and M. Yang, ‘Large graph processing in the cloud’, in *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, Indianapolis Indiana USA: ACM, June 2010, pp. 1123–1126. doi: 10.1145/1807167.1807297.
- [53] Y. Liu, F. Nie, J. Wu, and L. Chen, ‘Semi-supervised feature selection based on label propagation and subset selection’, in *2010 International Conference on Computer and Information Application*, Dec. 2010, pp. 293–296. doi: 10.1109/ICCIA.2010.6141595.
- [54] W. Yang, C. Hou, and Y. Wu, ‘A semi-supervised method for feature selection’, in *2011 International Conference on Computational and Information Sciences*, IEEE, 2011, pp. 329–332.
- [55] S. Lv, H. Jiang, L. Zhao, D. Wang, and M. Fan, ‘Manifold based fisher method for semi-supervised feature selection’, in *2013 10th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, July 2013, pp. 664–668. doi: 10.1109/FSKD.2013.6816279.

- [56] Y. Liu, F. Nie, J. Wu, and L. Chen, ‘Efficient semi-supervised feature selection with noise insensitive trace ratio criterion’, *Neurocomputing*, vol. 105, pp. 12–18, Apr. 2013, doi: 10.1016/j.neucom.2012.05.031.
- [57] Yahong Han, Yi Yang, Yan Yan, Zhigang Ma, N. Sebe, and Xiaofang Zhou, ‘Semisupervised Feature Selection via Spline Regression for Video Semantic Recognition’, *IEEE Trans. Neural Netw. Learning Syst.*, vol. 26, no. 2, pp. 252–264, Feb. 2015, doi: 10.1109/TNNLS.2014.2314123.
- [58] Y. Cheng, Y. Cai, Y. Sun, and J. Li, ‘Semi-supervised feature selection under logistic i-relief framework’, in *19th international conference on pattern recognition*, IEEE, 2008, pp. 1–4.
- [59] Y. Han, K. Park, and Y.-K. Lee, ‘Confident wrapper-type semi-supervised feature selection using an ensemble classifier’, in *2011 2nd International Conference on Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC)*, Aug. 2011, pp. 4581–4586. doi: 10.1109/AIMSEC.2011.6010202.
- [60] F. Bellal, H. Elghazel, and A. Aussem, ‘A semi-supervised feature ranking method with ensemble learning’, *Pattern Recognition Letters*, vol. 33, no. 10, pp. 1426–1433, July 2012, doi: 10.1016/j.patrec.2012.03.001.
- [61] Z. Ma, Y. Yang, F. Nie, J. Uijlings, and N. Sebe, ‘Exploiting the entire feature space with sparsity for automatic image annotation’, in *Proceedings of the 19th ACM international conference on Multimedia*, Scottsdale Arizona USA: ACM, Nov. 2011, pp. 283–292. doi: 10.1145/2072298.2072336.
- [62] Z. Ma, F. Nie, Y. Yang, J. R. R. Uijlings, N. Sebe, and A. G. Hauptmann, ‘Discriminating Joint Feature Analysis for Multimedia Data Understanding’, *IEEE Transactions on Multimedia*, vol. 14, no. 6, pp. 1662–1672, Dec. 2012, doi: 10.1109/TMM.2012.2199293.
- [63] C. Shi, Q. Ruan, and G. An, ‘Sparse feature selection based on graph Laplacian for web image annotation’, *Image and Vision Computing*, vol. 32, no. 3, pp. 189–201, Mar. 2014, doi: 10.1016/j.imavis.2013.12.013.
- [64] X. Song, J. Zhang, Y. Han, and J. Jiang, ‘Semi-supervised feature selection via hierarchical regression for web image classification’, *Multimedia Systems*, vol. 22, no. 1, pp. 41–49, Feb. 2016, doi: 10.1007/s00530-014-0390-0.
- [65] L. Yang and L. Wang, ‘Simultaneous Feature Selection and Classification via Semi-Supervised Models’, in *Third International Conference on Natural Computation (ICNC 2007)*, Aug. 2007, pp. 646–650. doi: 10.1109/ICNC.2007.666.
- [66] Z. Xu, I. King, M. R.-T. Lyu, and R. Jin, ‘Discriminative Semi-Supervised Feature Selection Via Manifold Regularization’, *IEEE Transactions on Neural Networks*, vol. 21, no. 7, pp. 1033–1047, July 2010, doi: 10.1109/TNN.2010.2047114.
- [67] K. Dai, H.-Y. Yu, and Q. Li, ‘A Semisupervised Feature Selection with Support Vector Machine’, *Journal of Applied Mathematics*, vol. 2013, pp. 1–11, 2013, doi: 10.1155/2013/416320.
- [68] J. C. Ang, H. Haron, and H. N. A. Hamed, ‘Semi-supervised SVM-based Feature Selection for Cancer Classification using Microarray Gene Expression Data’, in *Current Approaches in Applied Artificial Intelligence*, vol. 9101, Cham: Springer International Publishing, 2015, pp. 468–477. [Online]. Available: https://link.springer.com/10.1007/978-3-319-19066-2_45

- [69] R. Varshavsky, A. Gottlieb, M. Linial, and D. Horn, 'Novel unsupervised feature filtering of biological data', *Bioinformatics*, vol. 22, no. 14, pp. e507–e513, 2006.
- [70] D. Devakumari and K. Thangavel, 'Unsupervised adaptive floating search feature selection based on contribution entropy', in *2010 international conference on communication and computational intelligence (INCOCCI)*, IEEE, 2010, pp. 623–627. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/5738800/>
- [71] V. M. Rao and V. N. Sastry, 'Unsupervised feature ranking based on representation entropy', in *2012 1st International Conference on Recent Advances in Information Technology (RAIT)*, IEEE, 2012, pp. 421–425.
- [72] M. Banerjee and N. R. Pal, 'Feature selection with SVD entropy: Some modification and extension', *Information Sciences*, vol. 264, pp. 118–134, 2014.
- [73] Y. Liu *et al.*, 'A hybrid feature selection algorithm combining information gain and genetic search for intrusion detection', in *Journal of Physics: Conference Series*, IOP Publishing, 2020, p. 032048. Accessed: Jan. 23, 2025. [Online]. Available: <https://iopscience.iop.org/article/10.1088/1742-6596/1601/3/032048/meta>
- [74] X. He, D. Cai, and P. Niyogi, 'Laplacian score for feature selection', *Advances in neural information processing systems*, vol. 18, 2005, [Online]. Available: <https://proceedings.neurips.cc/paper/2909-laplacian-score-for-feature-selection>
- [75] Z. Zhao and H. Liu, 'Spectral feature selection for supervised and unsupervised learning', in *Proceedings of the 24th international conference on Machine learning*, Corvalis Oregon USA: ACM, June 2007, pp. 1151–1157. doi: 10.1145/1273496.1273641.
- [76] P. Padungweang, C. Lursinsap, and K. Sunat, 'Univariate filter technique for unsupervised feature selection using a new Laplacian score based local nearest neighbors', in *2009 Asia-Pacific Conference on Information Processing*, IEEE, 2009, pp. 196–200. Accessed: Jan. 23, 2025. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/5197170/>
- [77] S. Solorio-Fernandez, J. F. Martínez-Trinidad, and J. A. Carrasco-Ochoa, 'A new unsupervised spectral feature selection method for mixed data: a filter approach', *Pattern Recognition*, vol. 72, pp. 314–326, 2017.
- [78] M. Dash, K. Choi, P. Scheuermann, and Huan Liu, 'Feature selection for clustering - a filter solution', in *2002 IEEE International Conference on Data Mining, 2002. Proceedings.*, Maebashi City, Japan: IEEE Comput. Soc, 2002, pp. 115–122. doi: 10.1109/ICDM.2002.1183893.
- [79] L. Talavera, 'Dependency-Based Feature Selection for Clustering Symbolic Data', *Intelligent Data Analysis*, vol. 4, June 2001, doi: 10.3233/IDA-2000-4103.
- [80] P. Mitra, C. A. Murthy, and S. K. Pal, 'Unsupervised feature selection using feature similarity', *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 24, no. 3, pp. 301–312, Mar. 2002, doi: 10.1109/34.990133.
- [81] M. Haindl, P. Somol, D. Ververidis, and C. Kotropoulos, 'Feature Selection Based on Mutual Correlation', in *Progress in Pattern Recognition, Image Analysis and Applications*, vol. 4225, J. F. Martínez-Trinidad, J. A. Carrasco Ochoa, and J. Kittler, Eds, in *Lecture Notes in Computer Science*, vol. 4225. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 569–577. doi: 10.1007/11892755_59.

- [82] Y. Li, B.-L. Lu, and Z.-F. Wu, ‘Hierarchical fuzzy filter method for unsupervised feature selection’, *Journal of Intelligent & Fuzzy Systems*, vol. 18, no. 2, pp. 157–169, 2007.
- [83] C.-C. Yen, L.-C. Chen, and S.-D. Lin, ‘Unsupervised feature selection: minimize information redundancy of features’, in *2010 International Conference on Technologies and Applications of Artificial Intelligence*, IEEE, 2010, pp. 247–254. Accessed: Jan. 26, 2025. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/5695461/>
- [84] S. Wang, W. Pedrycz, Q. Zhu, and W. Zhu, ‘Unsupervised feature selection via maximum projection and minimum redundancy’, *Knowledge-Based Systems*, vol. 75, pp. 19–29, 2015.
- [85] A. J. Ferreira and M. A. Figueiredo, ‘An unsupervised approach to feature discretization and selection’, *Pattern Recognition*, vol. 45, no. 9, pp. 3048–3060, 2012.
- [86] J. Ghosh, D. Kumar, and R. Tripathi, ‘Features Extraction for Network Intrusion Detection Using Genetic Algorithm (GA)’, in *Modern Approaches in Machine Learning and Cognitive Science: A Walkthrough*, vol. 885, V. K. Gunjan, J. M. Zurada, B. Raman, and G. R. Gangadharan, Eds, in *Studies in Computational Intelligence*, vol. 885. , Cham: Springer International Publishing, 2020, pp. 13–25. doi: 10.1007/978-3-030-38445-6_2.
- [87] S. Tabakhi, P. Moradi, and F. Akhlaghian, ‘An unsupervised feature selection algorithm based on ant colony optimization’, *Engineering Applications of Artificial Intelligence*, vol. 32, pp. 112–123, June 2014, doi: 10.1016/j.engappai.2014.03.007.
- [88] S. Tabakhi and P. Moradi, ‘Relevance–redundancy feature selection based on ant colony optimization’, *Pattern Recognition*, vol. 48, no. 9, pp. 2798–2811, Sept. 2015, doi: 10.1016/j.patcog.2015.03.020.
- [89] S. Tabakhi, A. Najafi, R. Ranjbar, and P. Moradi, ‘Gene selection for microarray data classification using a novel ant colony optimization’, *Neurocomputing*, vol. 168, pp. 1024–1036, Nov. 2015, doi: 10.1016/j.neucom.2015.05.022.
- [90] B. Z. Dadaneh, H. Y. Markid, and A. Zakerolhosseini, ‘Unsupervised probabilistic feature selection using ant colony optimization’, *Expert Systems with Applications*, vol. 53, pp. 27–42, July 2016, doi: 10.1016/j.eswa.2016.01.021.
- [91] M. M. Hassan, A. Gumaei, A. Alsanad, M. Alrubaian, and G. Fortino, ‘A hybrid deep learning model for efficient intrusion detection in big data environment’, *Information Sciences*, vol. 513, pp. 386–396, 2020.
- [92] J. Chen, X. Qi, L. Chen, F. Chen, and G. Cheng, ‘Quantum-inspired ant lion optimized hybrid k-means for cluster analysis and intrusion detection’, *Knowledge-Based Systems*, vol. 203, p. 106167, June 2020, doi: 10.1016/j.knosys.2020.106167.
- [93] S. Sarvari, N. F. Mohd Sani, Z. Mohd Hanapi, and M. T. Abdullah, ‘An Efficient Anomaly Intrusion Detection Method With Feature Selection and Evolutionary Neural Network’, *IEEE Access*, vol. 8, pp. 70651–70663, 2020, doi: 10.1109/ACCESS.2020.2986217.
- [94] W. Elmasry, A. Akbulut, and A. H. Zaim, ‘Evolving deep learning architectures for network intrusion detection using a double PSO metaheuristic’, *Computer Networks*, vol. 168, p. 107042, Feb. 2020, doi: 10.1016/j.comnet.2019.107042.

- [95] D. García-García and R. Santos-Rodríguez, ‘Spectral clustering and feature selection for microarray data’, in *2009 International Conference on Machine Learning and Applications*, IEEE, 2009, pp. 425–428. Accessed: Jan. 26, 2025. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/5381488/>
- [96] S. Nijima and Y. Okuno, ‘Laplacian Linear Discriminant Analysis Approach to Unsupervised Feature Selection’, *IEEE/ACM Trans. Comput. Biol. and Bioinf.*, vol. 6, no. 4, pp. 605–614, Oct. 2009, doi: 10.1109/TCBB.2007.70257.
- [97] R. Liu, N. Yang, X. Ding, and L. Ma, ‘An Unsupervised Feature Selection Algorithm: Laplacian Score Combined with Distance-Based Entropy Measure’, in *2009 Third International Symposium on Intelligent Information Technology Application*, NanChang, China: IEEE, 2009, pp. 65–68. doi: 10.1109/IITA.2009.390.
- [98] D. Cai, C. Zhang, and X. He, ‘Unsupervised feature selection for multi-cluster data’, in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, Washington DC USA: ACM, July 2010, pp. 333–342. doi: 10.1145/1835804.1835848.
- [99] Z. Zhao, L. Wang, and H. Liu, ‘Efficient spectral feature selection with minimum redundancy’, in *Proceedings of the AAAI conference on artificial intelligence*, 2010, pp. 673–678. Accessed: Jan. 26, 2025. [Online]. Available: <https://ojs.aaai.org/index.php/AAAI/article/view/7671>
- [100] Y. Yang, H. T. Shen, Z. Ma, Z. Huang, and X. Zhou, ‘2,1-Norm Regularized Discriminative Feature Selection for Unsupervised Learning’.
- [101] Z. Li, Y. Yang, J. Liu, X. Zhou, and H. Lu, ‘Unsupervised feature selection using nonnegative spectral analysis’, in *Proceedings of the AAAI conference on artificial intelligence*, 2012, pp. 1026–1032. Accessed: Jan. 26, 2025. [Online]. Available: <https://ojs.aaai.org/index.php/AAAI/article/view/8289>
- [102] C. Hou, F. Nie, D. Yi, and Y. Wu, ‘Feature selection via joint embedding learning and sparse regression’, in *Twenty-Second international joint conference on Artificial Intelligence*, Citeseer, 2011. Accessed: Jan. 26, 2025. [Online]. Available: <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=afb5faa663016031e529ead790ec4ce44929901b>
- [103] C. Hou, F. Nie, X. Li, D. Yi, and Y. Wu, ‘Joint embedding learning and sparse regression: A framework for unsupervised feature selection’, *IEEE transactions on cybernetics*, vol. 44, no. 6, pp. 793–804, 2013.
- [104] Z. Zhao, L. Wang, H. Liu, and J. Ye, ‘On similarity preserving feature selection’, *IEEE Transactions on Knowledge and Data engineering*, vol. 25, no. 3, pp. 619–632, 2011.
- [105] Z. Li, J. Liu, Y. Yang, X. Zhou, and H. Lu, ‘Clustering-guided sparse structural learning for unsupervised feature selection’, *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 9, pp. 2138–2150, 2013.
- [106] M. Qian and C. Zhai, ‘Robust unsupervised feature selection’, in *Twenty-third international joint conference on artificial intelligence*, Citeseer, 2013.
- [107] L. Shi, L. Du, and Y.-D. Shen, ‘Robust Spectral Learning for Unsupervised Feature Selection’, in *2014 IEEE International Conference on Data Mining*, Shenzhen, China: IEEE, Dec. 2014, pp. 977–982. doi: 10.1109/ICDM.2014.58.

- [108] P. Zhu, Q. Hu, C. Zhang, and W. Zuo, ‘Coupled dictionary learning for unsupervised feature selection’, in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2016. Accessed: Jan. 26, 2025. [Online]. Available: <https://ojs.aaai.org/index.php/AAAI/article/view/10239>
- [109] F. Nie, W. Zhu, and X. Li, ‘Unsupervised feature selection with structured graph optimization’, in *Proceedings of the AAAI conference on artificial intelligence*, 2016. Accessed: Jan. 26, 2025. [Online]. Available: <https://ojs.aaai.org/index.php/AAAI/article/view/10168>
- [110] Y. Yi, W. Zhou, Y. Cao, Q. Liu, and J. Wang, ‘Unsupervised Feature Selection with Graph Regularized Nonnegative Self-representation’, in *Biometric Recognition*, vol. 9967, Z. You, J. Zhou, Y. Wang, Z. Sun, S. Shan, W. Zheng, J. Feng, and Q. Zhao, Eds, in Lecture Notes in Computer Science, vol. 9967. , Cham: Springer International Publishing, 2016, pp. 591–599. doi: 10.1007/978-3-319-46654-5_65.
- [111] X. Wang, X. Zhang, Z. Zeng, Q. Wu, and J. Zhang, ‘Unsupervised spectral feature selection with l1-norm graph’, *Neurocomputing*, vol. 200, pp. 47–54, 2016.
- [112] S. Du, Y. Ma, S. Li, and Y. Ma, ‘Robust unsupervised feature selection via matrix factorization’, *Neurocomputing*, vol. 241, pp. 115–127, 2017.
- [113] P. Zhu, W. Zuo, L. Zhang, Q. Hu, and S. C. Shiu, ‘Unsupervised feature selection by regularized self-representation’, *Pattern Recognition*, vol. 48, no. 2, pp. 438–446, 2015.
- [114] W. Zhou, C. Wu, Y. Yi, and G. Luo, ‘Structure preserving non-negative feature self-representation for unsupervised feature selection’, *IEEE Access*, vol. 5, pp. 8792–8803, 2017.
- [115] S. Wang and H. Wang, ‘Unsupervised feature selection via low-rank approximation and structure learning’, *Knowledge-Based Systems*, vol. 124, pp. 70–79, 2017.
- [116] Q. Lu, X. Li, and Y. Dong, ‘Structure preserving unsupervised feature selection’, *Neurocomputing*, vol. 301, pp. 36–45, 2018.
- [117] C. Tang *et al.*, ‘Robust unsupervised feature selection via dual self-representation and manifold regularization’, *Knowledge-Based Systems*, vol. 145, pp. 109–120, 2018.
- [118] C. Tang, X. Zhu, J. Chen, P. Wang, X. Liu, and J. Tian, ‘Robust graph regularized unsupervised feature selection’, *Expert Systems with Applications*, vol. 96, pp. 64–76, 2018.
- [119] M. Luo, F. Nie, X. Chang, Y. Yang, A. G. Hauptmann, and Q. Zheng, ‘Adaptive unsupervised feature selection with structure regularization’, *IEEE transactions on neural networks and learning systems*, vol. 29, no. 4, pp. 944–956, 2017.
- [120] Y. Shi, J. Miao, Z. Wang, P. Zhang, and L. Niu, ‘Feature selection with $\ell_{2,1}$ regularization’, *IEEE transactions on neural networks and learning systems*, vol. 29, no. 10, pp. 4967–4982, 2018.
- [121] M. Devaney and A. Ram, ‘Efficient feature selection in conceptual clustering’, in *ICML*, 1997, pp. 92–97. Accessed: Jan. 27, 2025. [Online]. Available: https://sfb876.tu-dortmund.de/PublicPublicationFiles/devaney_ram_97a.pdf
- [122] J. G. Dy and C. E. Brodley, ‘Feature selection for unsupervised learning’, *Journal of machine learning research*, vol. 5, no. Aug, pp. 845–889, 2004.

- [123] E. R. Hruschka and T. F. Covoos, 'Feature selection for cluster analysis: an approach based on the simplified Silhouette criterion', in *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)*, IEEE, 2005, pp. 32–38. Accessed: Nov. 20, 2024. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/1631238/>
- [124] M. Breaban and H. Luchian, 'A unifying criterion for unsupervised clustering and feature selection', *Pattern Recognition*, vol. 44, no. 4, pp. 854–865, 2011.
- [125] Y. Kim, W. N. Street, and F. Menczer, 'Evolutionary model selection in unsupervised learning', *IDA*, vol. 6, no. 6, pp. 531–556, Dec. 2002, doi: 10.3233/IDA-2002-6605.
- [126] D. Dutta, P. Dutta, and J. Sil, 'Simultaneous feature selection and clustering with mixed features by multi objective genetic algorithm', *International Journal of Hybrid Intelligent Systems*, vol. 11, no. 1, pp. 41–54, Feb. 2014, doi: 10.3233/HIS-130182.
- [127] S. Hosseini and B. M. H. Zade, 'New hybrid method for attack detection using combination of evolutionary algorithms, SVM, and ANN', *Computer Networks*, vol. 173, p. 107168, May 2020, doi: 10.1016/j.comnet.2020.107168.
- [128] V. Roth and T. Lange, 'Feature Selection in Clustering Problems', *Advances in Neural Information Processing Systems*, vol. 16, Mar. 2004.
- [129] M. H. C. Law, M. A. T. Figueiredo, and A. K. Jain, 'Simultaneous feature selection and clustering using mixture models', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 9, pp. 1154–1166, Sept. 2004, doi: 10.1109/TPAMI.2004.71.
- [130] H. Zeng and Y. Cheung, 'Feature Selection and Kernel Learning for Local Learning-Based Clustering', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1532–1547, Aug. 2011, doi: 10.1109/TPAMI.2010.215.
- [131] S. Wang, J. Tang, and H. Liu, 'Embedded Unsupervised Feature Selection', *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 29, no. 1, Art. no. 1, Feb. 2015, doi: 10.1609/aaai.v29i1.9211.
- [132] J. Guo, Y. Guo, X. Kong, and R. He, *Unsupervised Feature Selection with Ordinal Locality*. 2017. doi: 10.1109/ICME.2017.8019357.
- [133] J. Guo and W. Zhu, 'Dependence guided unsupervised feature selection', in *Proceedings of the AAAI conference on artificial intelligence*, 2018. Accessed: Jan. 26, 2025. [Online]. Available: <https://ojs.aaai.org/index.php/AAAI/article/view/11904>
- [134] M. Dash and H. Liu, 'Feature Selection for Clustering', in *Knowledge Discovery and Data Mining. Current Issues and New Applications*, T. Terano, H. Liu, and A. L. P. Chen, Eds, Berlin, Heidelberg: Springer, 2000, pp. 110–121. doi: 10.1007/3-540-45571-X_13.
- [135] E. R. Hruschka, T. F. Covoos, and N. F. Ebecken, 'Feature selection for clustering problems: a hybrid algorithm that iterates between k-means and a Bayesian filter', in *Fifth International Conference on Hybrid Intelligent Systems (HIS'05)*, IEEE, 2005, pp. 6-pp. Accessed: Jan. 26, 2025. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/1587781/>

- [136] Y. Li, B.-L. Lu, and Z.-F. Wu, 'A hybrid method of unsupervised feature selection based on ranking', in *18th International Conference on Pattern Recognition (ICPR'06)*, IEEE, 2006, pp. 687–690. Accessed: Jan. 26, 2025. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/1699298/>
- [137] Y. B. Kim and J. Gao, 'Unsupervised Gene Selection For High Dimensional Data', in *Sixth IEEE Symposium on BioInformatics and BioEngineering (BIBE'06)*, Oct. 2006, pp. 227–234. doi: 10.1109/BIBE.2006.253339.
- [138] S. Solorio-Fernández, J. A. Carrasco-Ochoa, and J. Fco. Martínez-Trinidad, 'A new hybrid filter–wrapper feature selection method for clustering based on ranking', *Neurocomputing*, vol. 214, pp. 866–880, Nov. 2016, doi: 10.1016/j.neucom.2016.07.026.
- [139] J. Kianat, M. A. Khan, M. Sharif, T. Akram, A. Rehman, and T. Saba, 'A joint framework of feature reduction and robust feature selection for cucumber leaf diseases recognition', *Optik*, vol. 240, p. 166566, Aug. 2021, doi: 10.1016/j.ijleo.2021.166566.
- [140] M. R. Islam, A. A. Lima, S. C. Das, M. F. Mridha, A. R. Prodeep, and Y. Watanobe, 'A Comprehensive Survey On The Process, Methods, Evaluation, and Challenges Of Feature Selection', *IEEE Access*, 2022.
- [141] L. E. Peterson, 'K-nearest neighbor', *Scholarpedia*, vol. 4, no. 2, p. 1883, 2009.
- [142] L. Breiman, 'Random forests', *Machine learning*, vol. 45, pp. 5–32, 2001.
- [143] A. Natekin and A. Knoll, 'Gradient boosting machines, a tutorial', *Frontiers in neurorobotics*, vol. 7, p. 21, 2013.
- [144] C. Cortes and V. Vapnik, 'Support vector machine', *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.
- [145] Y. Freund and R. E. Schapire, 'A desicion-theoretic generalization of on-line learning and an application to boosting', in *Computational Learning Theory: Second European Conference, EuroCOLT'95 Barcelona, Spain, March 13–15, 1995 Proceedings 2*, Springer, 1995, pp. 23–37.
- [146] A. Hashemi, M. B. Dowlatshahi, and H. Nezamabadi-pour, 'Ensemble of feature selection algorithms: a multi-criteria decision-making approach', *International Journal of Machine Learning and Cybernetics*, vol. 13, no. 1, pp. 49–69, 2022.
- [147] A. Kumar, A. Kaur, P. Singh, M. Driss, and W. Boulila, 'Efficient multiclass classification using feature selection in high-dimensional datasets', *Electronics*, vol. 12, no. 10, p. 2290, 2023.
- [148] H. Liu, J. Li, and L. Wong, 'A comparative study on feature selection and classification methods using gene expression profiles and proteomic patterns', *Genome informatics*, vol. 13, pp. 51–60, 2002.
- [149] V. Bolón-Canedo, N. Sánchez-Marono, and A. Alonso-Betanzos, 'A review of feature selection methods on synthetic data', *Knowledge and information systems*, vol. 34, pp. 483–519, 2013.
- [150] I. Inza, P. Larranaga, R. Blanco, and A. J. Cerrolaza, 'Filter versus wrapper gene selection approaches in DNA microarray domains', *Artificial intelligence in medicine*, vol. 31, no. 2, pp. 91–103, 2004.
- [151] G. Ansari, T. Ahmad, and M. N. Doja, 'Ensemble of feature ranking methods using hesitant fuzzy sets for sentiment classification', *International Journal of Machine Learning and Computing*, vol. 9, no. 5, pp. 599–608, 2019.

- [152] N. Sambyal, P. Saini, and R. Syal, 'A review of statistical and machine learning techniques for microvascular complications in type 2 diabetes', *Current Diabetes Reviews*, vol. 17, no. 2, pp. 143–155, 2021.
- [153] S. Mirza, S. Mittal, and M. Zaman, 'Decision support predictive model for prognosis of diabetes using SMOTE and decision tree', *International Journal of Applied Engineering Research*, vol. 13, no. 11, pp. 9277–9282, 2018.
- [154] D. K. Choubey, P. Kumar, S. Tripathi, and S. Kumar, 'Performance evaluation of classification methods with PCA and PSO for diabetes', *Network Modeling Analysis in Health Informatics and Bioinformatics*, vol. 9, pp. 1–30, 2020.
- [155] M. Fatima and M. Pasha, 'Survey of machine learning algorithms for disease diagnostic', *Journal of Intelligent Learning Systems and Applications*, vol. 9, no. 01, p. 1, 2017.
- [156] S. Hasan and S. M. Shamsuddin, 'Multi-strategy learning and deep harmony memory improvisation for self-organizing neurons', *Soft Computing*, vol. 23, pp. 285–303, 2019.
- [157] D. D. Lewis, 'Feature selection and feature extraction for text categorization', in *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992*, 1992.
- [158] R. Battiti, 'Using mutual information for selecting features in supervised neural net learning', *IEEE Transactions on neural networks*, vol. 5, no. 4, pp. 537–550, 1994.
- [159] H. Salem, G. Attiya, and N. El-Fishawy, 'Classification of human cancer diseases by gene expression profiles', *Applied Soft Computing*, vol. 50, pp. 124–134, 2017.
- [160] B. Sahu, 'A combo feature selection method (filter+ wrapper) for microarray gene classification', *International Journal of Pure and Applied Mathematics*, vol. 118, no. 16, pp. 389–401, 2018.
- [161] B. Subanya and R. R. Rajalaxmi, 'Feature selection using Artificial Bee Colony for cardiovascular disease classification', in *2014 International conference on electronics and communication systems (ICECS)*, IEEE, 2014, pp. 1–6.
- [162] B. Subanya and R. Rajalaxmi, 'A novel feature selection algorithm for heart disease classification', *International Journal of Computational Intelligence and Informatic*, vol. 4, no. 2, 2014.
- [163] S. Ojagh, F. Cauteruccio, G. Terracina, and S. H. Liang, 'Enhanced air quality prediction by edge-based spatiotemporal data preprocessing', *Computers & Electrical Engineering*, vol. 96, p. 107572, 2021.
- [164] P. Theerthagiri, A. U. Ruby, and J. Vidya, 'Diagnosis and Classification of the Diabetes Using Machine Learning Algorithms', *SN Computer Science*, vol. 4, no. 1, p. 72, 2022.
- [165] S. P. Chatrati *et al.*, 'Smart home health monitoring system for predicting type 2 diabetes and hypertension', *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 3, pp. 862–870, Mar. 2022, doi: 10.1016/j.jksuci.2020.01.010.
- [166] S. Apoorva, K. Aditya S, P. Snigdha, P. Darshini, and H. A. Sanjay, 'Prediction of diabetes mellitus type-2 using machine learning', in *Computational Vision and Bio-Inspired Computing: ICCVBIC 2019*, Springer, 2020, pp. 364–370.

- [167] Y. Ding, H. Zhu, R. Chen, and R. Li, 'An efficient AdaBoost algorithm with the multiple thresholds classification', *Applied Sciences*, vol. 12, no. 12, p. 5872, 2022.
- [168] X. Cui, Y. Li, J. Fan, and T. Wang, 'A novel filter feature selection algorithm based on relief', *Applied Intelligence*, vol. 52, no. 5, pp. 5063–5081, 2022.
- [169] Y. Qu, Y. Fang, and F. Yan, 'Feature selection algorithm based on association rules', in *Journal of Physics: Conference Series*, IOP Publishing, 2019, p. 052012.
- [170] K. Deep, 'A Random Walk Grey Wolf Optimizer based on dispersion factor for feature selection on Chronic Disease Prediction', *Expert Systems with Applications*, vol. 206, p. 117864, 2022.
- [171] M. A. Elsadig, A. Altigani, and H. T. Elshoush, 'Breast cancer detection using machine learning approaches: a comparative study.', *International Journal of Electrical & Computer Engineering (2088-8708)*, vol. 13, no. 1, 2023.
- [172] R. R. Kadhim and M. Y. Kamil, 'Comparison of machine learning models for breast cancer diagnosis', *IAES International Journal of Artificial Intelligence*, vol. 12, no. 1, p. 415, 2023.
- [173] N. Al-Azzam and I. Shatnawi, 'Comparing supervised and semi-supervised machine learning models on diagnosing breast cancer', *Annals of Medicine and Surgery*, vol. 62, pp. 53–64, 2021.
- [174] F. Khan *et al.*, 'Cloud-based breast cancer prediction empowered with soft computing approaches', *Journal of healthcare engineering*, vol. 2020, 2020.
- [175] T.-N.-Q. Nguyen, H.-T.-T. Vo, H. A. Nguyen, and T. Van Huynh, 'Machine Learning in Classification of Parkinson's Disease Using Electroencephalogram with Simon's Conflict', in *Computational Intelligence Methods for Green Technology and Sustainable Development: Proceedings of the International Conference GTSD2022*, Springer, 2022, pp. 110–122.
- [176] B. Devi, S. Srivastava, and V. K. Verma, 'Multiclass-Based Support Vector Machine for Parkinson's Disease Detection on Speech Data', in *Information Systems and Management Science: Conference Proceedings of 4th International Conference on Information Systems and Management Science (ISMS) 2021*, Springer, 2022, pp. 540–557.
- [177] R. Lamba, T. Gulati, H. F. Alharbi, and A. Jain, 'A hybrid system for Parkinson's disease diagnosis using machine learning techniques', *International Journal of Speech Technology*, pp. 1–11, 2021.
- [178] Z. K. Senturk, 'Early diagnosis of Parkinson's disease using machine learning algorithms', *Medical hypotheses*, vol. 138, p. 109603, 2020.
- [179] S. Sreejith, H. K. Nehemiah, and A. Kannan, 'A clinical decision support system for polycystic ovarian syndrome using red deer algorithm and random forest classifier', *Healthcare Analytics*, vol. 2, p. 100102, 2022.
- [180] S. Bharati, P. Podder, M. R. H. Mondal, V. B. Surya Prasath, and N. Gandhi, 'Ensemble Learning for Data-Driven Diagnosis of Polycystic Ovary Syndrome', in *Intelligent Systems Design and Applications: 21st International Conference on Intelligent Systems Design and Applications (ISDA 2021) Held During December 13–15, 2021*, Springer, 2022, pp. 1250–1259.
- [181] S. C. Nandipati, X. Chew, and K. W. Khaw, 'Polycystic Ovarian Syndrome (PCOS) classification and feature selection by machine learning techniques', *Appl Math Comput Intell*, vol. 9, pp. 65–74, 2020.

- [182] M. E. Cintra, H. A. Camargo, and M. C. Monard, ‘Genetic generation of fuzzy systems with rule extraction using formal concept analysis’, *Information Sciences*, vol. 349, pp. 199–215, 2016.
- [183] S. M. Mousavi, M. Tavana, N. Alikar, and M. Zandieh, ‘A tuned hybrid intelligent fruit fly optimization algorithm for fuzzy rule generation and classification’, *Neural Comput & Applic*, vol. 31, no. 3, pp. 873–885, Mar. 2019, doi: 10.1007/s00521-017-3115-4.
- [184] J. A. Sanz, A. Fernandez, H. Bustince, and F. Herrera, ‘IVTURS: A linguistic fuzzy rule-based classification system based on a new interval-valued fuzzy reasoning method with tuning and rule selection’, *IEEE Transactions on Fuzzy Systems*, vol. 21, no. 3, pp. 399–411, 2013.
- [185] Bardossy A. ., and L. Duckstein, *Fuzzy Rule-Based Modeling with Applications to Geophysical, Biological, and Engineering Systems.*, 1995th edn.
- [186] E. H. Mamdani, ‘Application of fuzzy algorithms for control of simple dynamic plant’, *Proc. Inst. Electr. Eng. UK*, vol. 121, no. 12, p. 1585, 1974, doi: 10.1049/piee.1974.0328.
- [187] O. Cordón and F. Herrera, ‘A proposal for improving the accuracy of linguistic modeling’, *IEEE Transactions on Fuzzy Systems*, vol. 8, no. 3, pp. 335–344, 2000.
- [188] O. Cordón and F. Herrera, ‘A three-stage evolutionary process for learning descriptive and approximate fuzzy-logic-controller knowledge bases from examples’, *International Journal of Approximate Reasoning*, vol. 17, no. 4, pp. 369–407, 1997.
- [189] A. Kumar and A. Kaur, ‘Study of Various Fuzzy Approaches for Disease Diagnosis’, presented at the Eleventh International Conference on Advances in Information Technology and Mobile Communication, AIM-2021, Mumbai, India: Greenze International Journal of Engineering and Technology, GJET, Feb. 2021, pp. 789–794.
- [190] F. Mansourypoor and S. Asadi, ‘Development of a reinforcement learning-based evolutionary fuzzy rule-based system for diabetes diagnosis’, *Computers in biology and medicine*, vol. 91, pp. 337–352, 2017.
- [191] A. Kumar, A. Kaur, and P. Singh, ‘Advancement in Learning and Optimization Techniques for Fuzzy Rule-Based System’, presented at the Hinweis International Conference on Recent Trends in Engineering and Technology, RTET - 2024, Grenze International Journal of Engineering and Technology (GIJET), Dec. 2024, pp. 3749–3757.
- [192] L. Zadeh, ‘Fuzzy logic (abstract): issues, contentions and perspectives’, in *Proceedings of the 22nd annual ACM computer science conference on Scaling up: meeting the challenge of complexity in real-world computing applications: meeting the challenge of complexity in real-world computing applications*, in CSC ’94. New York, NY, USA: Association for Computing Machinery, Mar. 1994, p. 407. doi: 10.1145/197530.197667.
- [193] R. Alcalá, J. Casillas, O. Cordón, F. Herrera, and I. Zwir, ‘Techniques for learning and tuning fuzzy rule-based systems for linguistic modeling and their application’, *Knowledge Engineering Systems, Techniques and Applications*, vol. 3, pp. 889–941, 1999.

- [194] L.-X. Wang and J. M. Mendel, ‘Generating fuzzy rules by learning from examples’, *IEEE Transactions on systems, man, and cybernetics*, vol. 22, no. 6, pp. 1414–1427, 1992.
- [195] K. Nozaki, H. Ishibuchi, and H. Tanaka, ‘A simple but powerful heuristic method for generating fuzzy rules from numerical data’, *Fuzzy sets and systems*, vol. 86, no. 3, pp. 251–270, 1997.
- [196] J.-J. Shann and H. C. Fu, ‘A fuzzy neural network for rule acquiring on fuzzy control systems’, *Fuzzy Sets and Systems*, vol. 71, no. 3, pp. 345–357, 1995.
- [197] J.-S. Jang, ‘ANFIS: adaptive-network-based fuzzy inference system’, *IEEE transactions on systems, man, and cybernetics*, vol. 23, no. 3, pp. 665–685, 1993.
- [198] O. Cord, *Genetic fuzzy systems: evolutionary tuning and learning of fuzzy knowledge bases*, vol. 19. World Scientific, 2001. Accessed: Jan. 27, 2025. [Online]. Available: https://books.google.com/books?hl=en&lr=&id=BWmSV-38fKAC&oi=fnd&pg=PR7&dq=genetic+fuzzy+systems:+evolutionary+tuning+and+learning+of+fuzzy+knowledge&ots=SPhilq8c9D&sig=XUHzg0_Kw6FKegBVGIJIMM4Lzdo
- [199] P. R. Thrift, ‘Fuzzy Logic Synthesis with Genetic Algorithms.’, in *ICGA*, 1991, pp. 509–513. Accessed: Jan. 27, 2025. [Online]. Available: <https://www.academia.edu/download/36706667/1991-Thrift-ICGA91.pdf>
- [200] J. Liska and S. S. Melsheimer, ‘Complete design of fuzzy logic systems using genetic algorithms’, in *Proceedings of 1994 IEEE 3rd International Fuzzy Systems Conference*, IEEE, 1994, pp. 1377–1382. Accessed: Jan. 27, 2025. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/343611/>
- [201] A. González, R. Pérez, and J. L. Verdegay, ‘Learning the structure of a fuzzy rule: a genetic approach’, *Fuzzy Systems and Artificial Intelligence*, vol. 3, no. 1, pp. 57–70, 1994.
- [202] A. Gonzalez and R. Perez, ‘Completeness and consistency conditions for learning fuzzy rules’, *Fuzzy Sets and Systems*, vol. 96, no. 1, pp. 37–51, May 1998, doi: 10.1016/S0165-0114(96)00280-1.
- [203] O. Cordo, M. J. Del Jesus, F. Herrera, and M. Lozano, ‘MOGUL: A methodology to obtain genetic fuzzy rule-based systems under the iterative rule learning approach’, *Int. J. Intell. Syst.*, vol. 14, no. 11, pp. 1123–1153, Nov. 1999, doi: 10.1002/(SICI)1098-111X(199911)14:11%3C1123::AID-INT4%3E3.0.CO;2-6.
- [204] J. Casillas, O. Cordon, F. Herrera, and L. Magdalena, ‘Interpretability Improvements to Find the Balance Interpretability-Accuracy in Fuzzy Modeling: An Overview’, in *Interpretability Issues in Fuzzy Modeling*, vol. 128, J. Casillas, O. Cordon, F. Herrera, and L. Magdalena, Eds, in *Studies in Fuzziness and Soft Computing*, vol. 128. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 3–22. doi: 10.1007/978-3-540-37057-4_1.
- [205] V. Gupta, H. Gaur, S. Vashishtha, U. Das, V. K. Singh, and D. J. Hemanth, ‘A fuzzy rule-based system with decision tree for breast cancer detection’, *IET Image Processing*, vol. 17, no. 7, pp. 2083–2096, May 2023, doi: 10.1049/ipr2.12774.
- [206] S. Porebski, ‘Evaluation of fuzzy membership functions for linguistic rule-based classifier focused on explainability, interpretability and reliability’, *Expert Systems with Applications*, vol. 199, p. 117116, Aug. 2022, doi: 10.1016/j.eswa.2022.117116.

- [207] A. Mazouzi, N. Hadroug, W. Alayed, A. Hafaifa, A. Iratni, and A. Kouzou, 'Comprehensive optimization of fuzzy logic-based energy management system for fuel-cell hybrid electric vehicle using genetic algorithm', *International Journal of Hydrogen Energy*, vol. 81, pp. 889–905, Sept. 2024, doi: 10.1016/j.ijhydene.2024.07.237.
- [208] P. Dutta and D. Doley, 'Medical Diagnosis Under Uncertain Environment Through Bipolar-Valued Fuzzy Sets', in *Computer Vision and Machine Intelligence in Medical Image Analysis*, vol. 992, M. Gupta, D. Konar, S. Bhattacharyya, and S. Biswas, Eds, in *Advances in Intelligent Systems and Computing*, vol. 992, Singapore: Springer Singapore, 2020, pp. 127–135. doi: 10.1007/978-981-13-8798-2_13.
- [209] M. J. Gacto, R. Alcalá, and F. Herrera, 'Interpretability of linguistic fuzzy rule-based systems: An overview of interpretability measures', *Information Sciences*, vol. 181, no. 20, pp. 4340–4360, Oct. 2011, doi: 10.1016/j.ins.2011.02.021.
- [210] S.-M. Zhou and J. Q. Gan, 'Low-level interpretability and high-level interpretability: a unified view of data-driven interpretable fuzzy system modelling', *Fuzzy sets and systems*, vol. 159, no. 23, pp. 3091–3131, 2008.
- [211] R. Alcalá, O. Cordon, J. Casillas, F. Herrera, and S. J. I. Zwir, '29 - Learning and Tuning Fuzzy Rule-Based Systems for Linguistic Modeling', in *Knowledge-Based Systems*, C. T. Leondes, Ed., San Diego: Academic Press, 2000, pp. 889–941. doi: 10.1016/B978-012443875-0/50030-6.
- [212] D. Wang, D. Tan, and L. Liu, 'Particle swarm optimization algorithm: an overview', *Soft Comput*, vol. 22, no. 2, pp. 387–408, Jan. 2018, doi: 10.1007/s00500-016-2474-6.
- [213] D. Nauck and R. Kruse, 'Obtaining interpretable fuzzy classification rules from medical data', *Artificial intelligence in medicine*, vol. 16, no. 2, pp. 149–169, 1999.
- [214] F. Liu, G. S. Ng, C. Quek, and T. F. Loh, 'Artificial ventilation modeling using neuro-fuzzy hybrid system', in *The 2006 IEEE International Joint Conference on Neural Network Proceedings*, IEEE, 2006, pp. 2859–2864. Accessed: Jan. 14, 2025. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/1716485/>
- [215] E. Yargholi and S. Parvaneh, 'Novel cardiac risk factor stratification using neuro-fuzzy tool', in *2008 International Conference on Computational Intelligence for Modelling Control & Automation*, IEEE, 2008, pp. 1199–1204. Accessed: Jan. 14, 2025. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/5172796/>
- [216] S. Naghibi, M. Teshnehlab, and M. A. Shoorehdeli, 'Breast Cancer Classification Based on Advanced Multi Dimensional Fuzzy Neural Network', *J Med Syst*, vol. 36, no. 5, pp. 2713–2720, Oct. 2012, doi: 10.1007/s10916-011-9747-5.
- [217] B. Fatima and C. M. Amine, 'A neuro-fuzzy inference model for breast cancer recognition', *International Journal of Computer Science & Information Technology*, vol. 4, no. 5, p. 163, 2012.
- [218] B. J. Jacob, E. Y. Cheu, J. Tan, and C. Quek, 'Self-reorganizing TSK fuzzy inference system with BCM theory of meta-plasticity', in *The 2012 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2012, pp. 1–8. Accessed:

- Jan. 12, 2025. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/6252527/>
- [219] W.-M. Tan and H.-C. Quek, 'Adaptive training schema in Mamdani-type neuro-fuzzy models for data-analysis in dynamic system forecasting', in *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, IEEE, 2008, pp. 1733–1738. Accessed: Jan. 12, 2025. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/4634032/>
- [220] K. K. Ang and C. Quek, 'RSPOP: Rough set-based pseudo outer-product Fuzzy rule identification algorithm', *Neural computation*, vol. 17, no. 1, pp. 205–243, 2005.
- [221] Y. Jin, 'Fuzzy modeling of high-dimensional systems: complexity reduction and interpretability improvement', *IEEE Transactions on Fuzzy Systems*, vol. 8, no. 2, pp. 212–221, 2000.
- [222] J. M. Alonso Moral, C. Castiello, L. Magdalena, and C. Mencar, 'Interpretability Constraints and Criteria for Fuzzy Systems', in *Explainable Fuzzy Systems*, vol. 970, in *Studies in Computational Intelligence*, vol. 970. , Cham: Springer International Publishing, 2021, pp. 49–89. doi: 10.1007/978-3-030-71098-9_3.
- [223] K. V. Shihabudheen and G. N. Pillai, 'Recent advances in neuro-fuzzy system: A survey', *Knowledge-Based Systems*, vol. 152, pp. 136–162, 2018.
- [224] M. M. Ahmed and N. A. M. Isa, 'Knowledge base to fuzzy information granule: A review from the interpretability-accuracy perspective', *Applied Soft Computing*, vol. 54, pp. 121–140, 2017.
- [225] H. Ishibuchi, T. Murata, and I. B. Türkşen, 'Single-objective and two-objective genetic algorithms for selecting linguistic rules for pattern classification problems', *Fuzzy sets and systems*, vol. 89, no. 2, pp. 135–150, 1997.
- [226] H. Ishibuchi, K. Nozaki, N. Yamamoto, and H. Tanaka, 'Selecting fuzzy if-then rules for classification problems using genetic algorithms', *IEEE Transactions on fuzzy systems*, vol. 3, no. 3, pp. 260–270, 1995.
- [227] W. Pedrycz and J. V. de Oliveira, 'Optimization of fuzzy models', *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 26, no. 4, pp. 627–636, 1996.
- [228] J. V. De Oliveira, 'Semantic constraints for membership function optimization', *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 29, no. 1, pp. 128–138, 1999.
- [229] J. V. De Oliveira, 'Towards neuro-linguistic modeling: constraints for optimization of membership functions', *Fuzzy Sets and Systems*, vol. 106, no. 3, pp. 357–380, 1999.
- [230] Y. Jin, 'Generating distinguishable, complete, consistent and compact fuzzy systems using evolutionary algorithms', in *Accuracy Improvements in Linguistic Fuzzy Modeling*, vol. 129, J. Casillas, O. Cordón, F. Herrera, and L. Magdalena, Eds, in *Studies in Fuzziness and Soft Computing*, vol. 129. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 100–118. doi: 10.1007/978-3-540-37058-1_5.
- [231] F. Cheong and R. Lai, 'Constrained optimization of genetic fuzzy systems', in *Accuracy Improvements in Linguistic Fuzzy Modeling*, vol. 129, J. Casillas, O.

- Cordón, F. Herrera, and L. Magdalena, Eds, in *Studies in Fuzziness and Soft Computing*, vol. 129. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 46–71. doi: 10.1007/978-3-540-37058-1_3.
- [232] J. J. Espinosa and J. Vandewalle, ‘Constructing Fuzzy Models with Linguistic Integrity -AFRELI Algorithm’.
- [233] O. Cordón, F. Herrera, and P. Villar, ‘Generating the knowledge base of a fuzzy rule-based system by the genetic learning of the data base’, *IEEE Transactions on fuzzy systems*, vol. 9, no. 4, pp. 667–674, 2001.
- [234] H. Ishibuchi, T. Nakashima, and T. Murata, ‘Three-objective genetics-based machine learning for linguistic rule extraction’, *Information sciences*, vol. 136, no. 1–4, pp. 109–133, 2001.
- [235] T. Suzuki, T. Kodama, T. Furuhashi, and H. Tsutsui, ‘Fuzzy modeling using genetic algorithms with fuzzy entropy as conciseness measure’, *Information Sciences*, vol. 136, no. 1–4, pp. 53–67, 2001.
- [236] O. Cordón, M. J. Del Jesus, F. Herrera, L. Magdalena, and P. Villar, ‘A Multiobjective Genetic Learning Process for joint Feature Selection and Granularity and Contexts Learning in Fuzzy Rule-Based Classification Systems’, in *Interpretability Issues in Fuzzy Modeling*, vol. 128, J. Casillas, O. Cordón, F. Herrera, and L. Magdalena, Eds, in *Studies in Fuzziness and Soft Computing*, vol. 128. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 79–99. doi: 10.1007/978-3-540-37057-4_4.
- [237] S. Guillaume and B. Charnomordic, ‘A new method for inducing a set of interpretable fuzzy partitions and fuzzy inference systems from data’, in *Interpretability Issues in Fuzzy Modeling*, vol. 128, J. Casillas, O. Cordón, F. Herrera, and L. Magdalena, Eds, in *Studies in Fuzziness and Soft Computing*, vol. 128. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 148–175. doi: 10.1007/978-3-540-37057-4_7.
- [238] H. Ishibuchi and T. Yamamoto, ‘Trade-off between the Number of Fuzzy Rules and Their Classification Performance’, in *Accuracy Improvements in Linguistic Fuzzy Modeling*, vol. 129, J. Casillas, O. Cordón, F. Herrera, and L. Magdalena, Eds, in *Studies in Fuzziness and Soft Computing*, vol. 129. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 72–99. doi: 10.1007/978-3-540-37058-1_4.
- [239] H. Ishibuchi and T. Yamamoto, ‘Interpretability Issues in Fuzzy Genetics-Based Machine Learning for Linguistic Modelling’, in *Modelling with Words*, vol. 2873, J. Lawry, J. Shanahan, and A. L. Ralescu, Eds, in *Lecture Notes in Computer Science*, vol. 2873. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 209–228. doi: 10.1007/978-3-540-39906-3_11.
- [240] D. D. Nauck, ‘Measuring interpretability in rule-based classification systems’, in *The 12th IEEE International Conference on Fuzzy Systems, 2003. FUZZ’03.*, IEEE, 2003, pp. 196–201. Accessed: Feb. 06, 2025. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/1209361/>
- [241] W. Pedrycz, ‘Expressing Relevance Interpretability and Accuracy of Rule-Based Systems’, in *Interpretability Issues in Fuzzy Modeling*, vol. 128, J. Casillas, O. Cordón, F. Herrera, and L. Magdalena, Eds, in *Studies in Fuzziness and Soft Computing*, vol. 128. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 546–567. doi: 10.1007/978-3-540-37057-4_23.

- [242] C.-A. Peña-Reyes and M. Sipper, ‘Fuzzy CoCo: Balancing Accuracy and Interpretability of Fuzzy Models by Means of Coevolution’, in *Accuracy Improvements in Linguistic Fuzzy Modeling*, vol. 129, J. Casillas, O. Cordón, F. Herrera, and L. Magdalena, Eds, in *Studies in Fuzziness and Soft Computing*, vol. 129. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 119–146. doi: 10.1007/978-3-540-37058-1_6.
- [243] D. Tikk, T. D. Gedeon, and K. W. Wong, ‘A Feature Ranking Algorithm for Fuzzy Modelling Problems’, in *Interpretability Issues in Fuzzy Modeling*, vol. 128, J. Casillas, O. Cordón, F. Herrera, and L. Magdalena, Eds, in *Studies in Fuzziness and Soft Computing*, vol. 128. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 176–192. doi: 10.1007/978-3-540-37057-4_8.
- [244] V. Vanhoucke and R. Silipo, ‘Interpretability in Multidimensional Classification’, in *Interpretability Issues in Fuzzy Modeling*, vol. 128, J. Casillas, O. Cordón, F. Herrera, and L. Magdalena, Eds, in *Studies in Fuzziness and Soft Computing*, vol. 128. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 193–217. doi: 10.1007/978-3-540-37057-4_9.
- [245] S. Guillaume and B. Charnomordic, ‘Generating an interpretable family of fuzzy partitions from data’, *IEEE transactions on fuzzy systems*, vol. 12, no. 3, pp. 324–335, 2004.
- [246] H. Ishibuchi and T. Yamamoto, ‘Fuzzy rule selection by multi-objective genetic local search algorithms and rule evaluation measures in data mining’, *Fuzzy sets and systems*, vol. 141, no. 1, pp. 59–88, 2004.
- [247] J. Casillas, O. Cordón, M. J. Del Jesus, and F. Herrera, ‘Genetic tuning of fuzzy rule deep structures preserving interpretability and its interaction with fuzzy rule set reduction’, *IEEE Transactions on Fuzzy Systems*, vol. 13, no. 1, pp. 13–29, 2005.
- [248] K. Narukawa, Y. Nojima, and H. Ishibuchi, ‘Modification of evolutionary multiobjective optimization algorithms for multiobjective design of fuzzy rule-based classification systems’, in *The 14th IEEE International Conference on Fuzzy Systems, 2005. FUZZ’05.*, IEEE, 2005, pp. 809–814. Accessed: Feb. 06, 2025. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/1452498/>
- [249] R. Mikut, J. Jäkel, and L. Gröll, ‘Interpretability issues in data-based learning of fuzzy systems’, *Fuzzy sets and systems*, vol. 150, no. 2, pp. 179–197, 2005.
- [250] R. Alcalá, J. Alcalá-Fdez, F. Herrera, and J. Otero, ‘Genetic learning of accurate and compact fuzzy rule based systems based on the 2-tuples linguistic representation’, *International Journal of Approximate Reasoning*, vol. 44, no. 1, pp. 45–64, 2007.
- [251] R. Alcalá, J. Alcalá-Fdez, and F. Herrera, ‘A proposal for the genetic lateral tuning of linguistic fuzzy systems and its interaction with rule selection’, *IEEE Transactions on Fuzzy Systems*, vol. 15, no. 4, pp. 616–635, 2007.
- [252] R. Alcalá, M. J. Gacto, F. Herrera, and J. Alcalá-Fdez, ‘A MULTI-OBJECTIVE GENETIC ALGORITHM FOR TUNING AND RULE SELECTION TO OBTAIN ACCURATE AND COMPACT LINGUISTIC FUZZY RULE-BASED SYSTEMS’, *Int. J. Unc. Fuzz. Knowl. Based Syst.*, vol. 15, no. 05, pp. 539–557, Oct. 2007, doi: 10.1142/S0218488507004868.

- [253] M. Cococcioni, P. Ducange, B. Lazzerini, and F. Marcelloni, 'A Pareto-based multi-objective evolutionary approach to the identification of Mamdani fuzzy systems', *Soft Comput*, vol. 11, no. 11, pp. 1013–1031, Sept. 2007, doi: 10.1007/s00500-007-0150-6.
- [254] P. Fazendeiro, J. V. de Oliveira, and W. Pedrycz, 'A multiobjective design of a patient and anaesthetist-friendly neuromuscular blockade controller', *IEEE Transactions on Biomedical Engineering*, vol. 54, no. 9, pp. 1667–1678, 2007.
- [255] H. Ishibuchi and Y. Nojima, 'Analysis of interpretability-accuracy tradeoff of fuzzy systems by multiobjective fuzzy genetics-based machine learning', *International Journal of Approximate Reasoning*, vol. 44, no. 1, pp. 4–31, 2007.
- [256] F. Liu, C. Quek, and G. S. Ng, 'A novel generic hebbian ordering-based fuzzy rule base reduction approach to Mamdani neuro-fuzzy system', *Neural Computation*, vol. 19, no. 6, pp. 1656–1680, 2007.
- [257] C. Mencar, G. Castellano, and A. M. Fanelli, 'Distinguishability quantification of fuzzy sets', *Information Sciences*, vol. 177, no. 1, pp. 130–149, 2007.
- [258] J. M. Alonso, L. Magdalena, and S. Guillaume, 'HILK: A new methodology for designing highly interpretable linguistic knowledge bases using the fuzzy logic formalism', *Int. J. Intell. Syst.*, vol. 23, no. 7, pp. 761–794, July 2008, doi: 10.1002/int.20288.
- [259] P. Pulkkinen, J. Hytönen, and H. Koivisto, 'Developing a bioaerosol detector using hybrid genetic fuzzy systems', *Engineering Applications of Artificial Intelligence*, vol. 21, no. 8, pp. 1330–1346, 2008.
- [260] P. Pulkkinen and H. Koivisto, 'Fuzzy classifier identification using decision tree and multiobjective evolutionary algorithms', *International Journal of Approximate Reasoning*, vol. 48, no. 2, pp. 526–543, 2008.
- [261] R. Alcalá, P. Ducange, F. Herrera, B. Lazzerini, and F. Marcelloni, 'A multiobjective evolutionary approach to concurrently learn rule and data bases of linguistic fuzzy-rule-based systems', *IEEE Transactions on fuzzy systems*, vol. 17, no. 5, pp. 1106–1122, 2009.
- [262] J. M. Alonso, L. Magdalena, and G. González-Rodríguez, 'Looking for a good fuzzy system interpretability index: An experimental approach', *International Journal of Approximate Reasoning*, vol. 51, no. 1, pp. 115–134, 2009.
- [263] A. Botta, B. Lazzerini, F. Marcelloni, and D. C. Stefanescu, 'Context adaptation of fuzzy systems through a multi-objective evolutionary approach based on a novel interpretability index', *Soft Computing*, vol. 13, pp. 437–449, 2009.
- [264] M. J. Gacto, R. Alcalá, and F. Herrera, 'Adaptation and application of multi-objective evolutionary algorithms for rule reduction and parameter tuning of fuzzy rule-based systems', *Soft Comput*, vol. 13, no. 5, pp. 419–436, Mar. 2009, doi: 10.1007/s00500-008-0359-z.
- [265] P. Pulkkinen and H. Koivisto, 'A dynamically constrained multiobjective genetic fuzzy system for regression problems', *IEEE Transactions on Fuzzy Systems*, vol. 18, no. 1, pp. 161–177, 2009.
- [266] J. M. Alonso, L. Magdalena, and O. Cordón, 'Embedding HILK in a three-objective evolutionary algorithm with the aim of modeling highly interpretable fuzzy rule-based classifiers', in *2010 4th International Workshop on Genetic and Evolutionary Fuzzy Systems (GEFS)*, IEEE, 2010, pp. 15–20. Accessed: Feb. 06,

2025. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/5454165/>
- [267] J. M. Alonso and L. Magdalena, ‘HILK++: an interpretability-guided fuzzy modeling methodology for learning readable and comprehensible fuzzy rule-based classifiers’, *Soft Computing*, vol. 15, pp. 1959–1980, 2011.
- [268] M. J. Gacto, R. Alcalá, and F. Herrera, ‘Integration of an index to preserve the semantic interpretability in the multiobjective evolutionary rule selection and tuning of linguistic fuzzy systems’, *IEEE Transactions on Fuzzy Systems*, vol. 18, no. 3, pp. 515–531, 2010.
- [269] A. A. Márquez, F. A. Márquez, and A. Peregrín, ‘A multi-objective evolutionary algorithm with an interpretability improvement mechanism for linguistic fuzzy systems with adaptive defuzzification’, in *International Conference on Fuzzy Systems*, IEEE, 2010, pp. 1–7. Accessed: Feb. 06, 2025. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/5584294/>
- [270] C. Mencar, C. Castiello, R. Cannone, and A. M. Fanelli, ‘Interpretability assessment of fuzzy knowledge bases: A cointension based approach’, *International Journal of Approximate Reasoning*, vol. 52, no. 4, pp. 501–518, 2011.
- [271] R. Alcalá, Y. Nojima, F. Herrera, and H. Ishibuchi, ‘Generating single granularity-based fuzzy classification rules for multiobjective genetic fuzzy rule selection’, in *2009 IEEE International Conference on Fuzzy Systems*, IEEE, 2009, pp. 1718–1723. Accessed: Feb. 06, 2025. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/5277369/>
- [272] M. B. Gorzałczany and F. Rudziński, ‘Interpretable and accurate medical data classification—a multi-objective genetic-fuzzy optimization approach’, *Expert Systems with Applications*, vol. 71, pp. 26–39, 2017.
- [273] D. García, J. C. Gámez, A. González, and R. Pérez, ‘An interpretability improvement for fuzzy rule bases obtained by the iterative rule learning approach’, *International Journal of Approximate Reasoning*, vol. 67, pp. 37–58, 2015.
- [274] J. A. Sanz and H. Bustince, ‘A wrapper methodology to learn interval-valued fuzzy rule-based classification systems’, *Applied Soft Computing*, vol. 104, p. 107249, 2021.
- [275] H. Ouifak and A. Idri, ‘On the performance and interpretability of Mamdani and Takagi-Sugeno-Kang based neuro-fuzzy systems for medical diagnosis’, *Scientific African*, vol. 20, p. e01610, 2023.
- [276] I. Gadaras and L. Mikhailov, ‘An interpretable fuzzy rule-based classification methodology for medical diagnosis’, *Artificial intelligence in medicine*, vol. 47, no. 1, pp. 25–41, 2009.
- [277] C. Ornelas-Vences, L. P. Sanchez-Fernandez, L. A. Sanchez-Perez, A. Garza-Rodriguez, and A. Villegas-Bastida, ‘Fuzzy inference model evaluating turn for Parkinson’s disease patients’, *Computers in biology and medicine*, vol. 89, pp. 379–388, 2017.
- [278] A. Garza-Rodríguez, L. P. Sánchez-Fernández, L. A. Sánchez-Pérez, C. Ornelas-Vences, and M. Ehrenberg-Inzunza, ‘Pronation and supination analysis based on biomechanical signals from Parkinson’s disease patients’, *Artificial*

- Intelligence in Medicine*, vol. 84, pp. 7–22, Jan. 2018, doi: 10.1016/j.artmed.2017.10.001.
- [279] A. Garza-Rodríguez, L. P. Sánchez-Fernández, L. A. Sánchez-Pérez, and J. J. Carbajal Hernández, ‘Fuzzy inference model based on triaxial signals for pronation and supination assessment in Parkinson’s disease patients’, *Artificial Intelligence in Medicine*, vol. 105, p. 101873, May 2020, doi: 10.1016/j.artmed.2020.101873.
-

LIST OF PUBLICATIONS (JOURNALS)

- Ankur Kumar, Avinash Kaur (2023). Literature Review on Development of Feature Selection and Learning Mechanism for Fuzzy Rule-Based System. *Recent Advances in Computer Science and Communications (Formerly: Recent Patents on Computer Science)*, 16(4), 42-62. (Published, Indexing: Scopus)
 - Kumar, A., Kaur, A., Singh, P., Driss, M., & Boulila, W. (2023). Efficient multiclass classification using feature selection in high-dimensional datasets. *Electronics*, 12(10), 2290. (Published, Indexing: Scopus, SCIE, IF 2.6)
 - Ankur Kumar, Avinash Kaur, Parminder Singh, Evolutionary Swarm Optimization of FRBS for High-Dimensional Interpretability-Accuracy Balance” (Accepted, *International Journal of Fuzzy Systems*, Indexing: Scopus, SCIE, Impact Factor: 3.6)
-

LIST OF CONFERENCES

- Ankur Kumar, Avinash Kaur (2021). Study of various fuzzy approaches for disease diagnosis. 12th International Conference on Advances in Computing, Control, and Telecommunication Technologies, ACT 2021, 2021-August, 789-794. (Published, Conference, Scopus)
 - Ankur Kumar, Avinash Kaur, Parminder Singh (2024), Advancement in Learning and Optimization Techniques for Fuzzy Rule-Based System, Hinweis International Conference on Recent Trends in Engineering and Technology RTET - 2024, Greenze International Journal of Engineering and Technology (GIJET), Dec. 2024, pp. 3749–3757 (Published, Conference)
-