UNIVERSITAT ROVIRA I VIRGILI

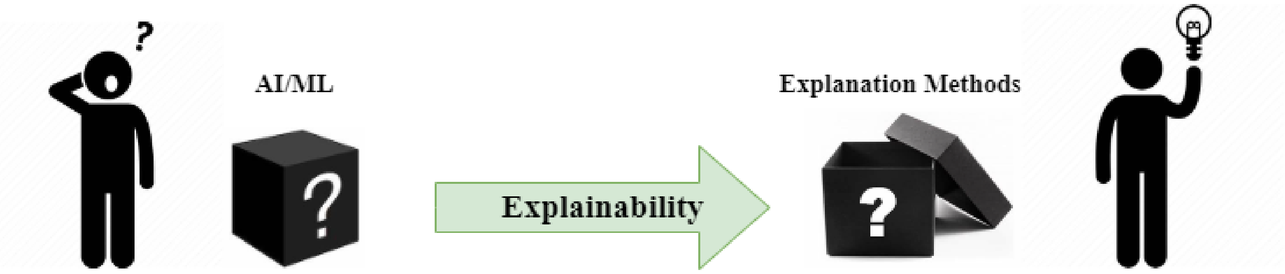
DEVELOPMENT OF EXPLAINABLE METHODS FOR FUZZY DECISION SUPPORT SYSTEMS

Najlaa Maaroof wahib AL Ziyadi

UNIVERSITAT
ROVIRA i VIRGILI

Development of explainable methods for fuzzy decision support systems

Author:
Najlaa MAAROOF



DOCTORAL THESIS
2023

Najlaa MAAROOF

Development of explainable methods for fuzzy decision support systems

DOCTORAL THESIS

Supervisors:
Prof. Dr. Antonio MORENO RIBAS

Department of Computer Engineering and Mathematics

UNIVERSITAT ROVIRA i VIRGILI

May, 2023

UNIVERSITAT ROVIRA i VIRGILI

I STATE that the present study, entitled “Development of explainable methods for fuzzy decision support systems," presented by Najlaa Maaroof Wahib Al Ziyadi for the award of the degree of doctor, has been carried out under my supervision at the Department of Computer Engineering and Mathematics of this university.

Tarragona, May 29th, 2023.

Doctoral Thesis Supervisor,
Prof. Dr. Antonio Moreno Ribas

Abstract

Recent advances in Artificial Intelligence (AI) have led to the widespread adoption of machine learning systems in various domains. However, the increased complexity of these systems has made it difficult to understand their decision-making processes, leading to their "black box" nature. This ambiguity has become problematic, especially in critical domains like healthcare, where trust in the decisions of machine learning-based systems is essential. Therefore, the development of Explainable Artificial Intelligence (XAI) has emerged as a new research field that seeks to develop more transparent and interpretable AI models. Furthermore, XAI is critical for building trustworthy AI systems that can be effectively audited and monitored, especially in healthcare, where high precision is not enough to convince society to trust the decisions of ML-based systems.

The main goal of this Ph.D thesis is to develop effective methods for explaining fuzzy decision support systems. As these systems become more ubiquitous in real-world applications, their interpretability has become a significant challenge, particularly in critical domains such as healthcare and legal decision-making.

To achieve the goal of explaining fuzzy systems, we propose a novel method that focuses on searching for neighbors in the input space. This method represents our work's first contribution, as the neighborhood generation step is crucial for producing robust and reliable explanations. The second contribution is introducing a new technique for generating explanations of fuzzy attributes in Machine Learning (ML) systems based on fuzzy logic. This technique uses knowledge about the fuzzy sets associated with each attribute to develop effective explanation methods. By using this technique, we can provide a more detailed and accurate understanding of the role of individual fuzzy attributes in the overall decision-making process of the model. This can enhance the interpretability and transparency of machine learning models, especially in critical domains where trust and accountability are crucial. Next, the thesis deals with a comparative study of two rule-based explanation methods for diabetic retinopathy risk assessment. This study aimed to evaluate one of our proposed methods and ascertain their effectiveness and usability in explicating the decision-making process of a fuzzy system designed for diabetic retinopathy. In addition, it provides valuable insights into the potential for enhancing this method and its clinical application. The last contribution of this thesis is developing a novel approach for extracting local and counterfactual explanations using fuzzy decision trees suitable for both binary and multiclass classification problems. This method serves as an alternative to classical decision trees. It is specifically designed to provide interpretable and actionable insights into fuzzy systems' decision-making process by providing explanations tailored to the user's preferences and easy to understand. The user-centric nature of this approach is a significant contribution, as it highlights the importance of designing machine learning systems that are transparent and accountable to their users.

Keywords: Artificial Intelligence; Explainable artificial intelligence; Machine learning.

Acknowledgements

The author was supported by a predoctoral grant provided by URV Martí Franquès.

This thesis' studies were funded by the Research projects PI21/00064 and PI18/00169 from ISCIII & FEDER funds and URV grants 2020PFR-B2-61 & 2019PFR-B2-61.

I would like to acknowledge and express my sincere gratitude to my supervisor, Dr. Antonio Moreno Ribas, for his invaluable guidance, insightful comments, and unwavering encouragement throughout the process of completing this thesis. His continuous support, guidance, and professionalism were critical to the success of this research work. Furthermore, I would like to extend my appreciation to Dr. Aida Valls and Dr. Mohammed Jabreel for their valuable comments and constructive suggestions, which greatly enriched the quality and depth of this thesis. Their contributions have been instrumental in advancing the research.

Contents

List of Figures

List of Tables

List of Abbreviations

AI	Artificial Intelligence
DL	Deep Learning
DR	Diabetic Retinopathy
XAI	Explainable Artificial Intelligence
ML	Machine Learning
DRSA	Dominance-based Rough Set Approach
CDSS	clinical decision support systems
DT	Decision Tree
FRF	Fuzzy Random Forest
FDT	Fuzzy Decision Tree
LIME	Local Interpretable Model-agnostic Explanations
LORE	Local Rule-Based Explanations
DM	Diabetes Mellitus
HVDM	Heterogeneous Value Difference Metric
NHS	National Health Surveys
GDPR	General Data Protection Regulation

I would like to dedicate this thesis to
my parents, my daughter Lyan, my son Loay
for their endless love, support and encouragement.
And to my husband Mohammed.
You made my life so much better in so many ways
that it is hard to imagine doing this without you.

Chapter 1

Introduction

1.1 Motivation

1.1.1 The Challenge of Explainability in Machine Learning

Artificial Intelligence (AI) and Machine Learning (ML) are rapidly advancing fields that have enabled many applications and innovations across various domains. AI is the science and engineering of creating intelligent machines and systems that can perform tasks that normally require human intelligence, such as perception, reasoning, learning, decision-making, and natural language processing [1]. ML is a subfield of AI that focuses on designing algorithms and systems that can learn from data and improve their performance over time [2]. ML models can perform tasks such as image recognition, natural language processing, recommendation systems, and more.

The availability of large amounts of data and computational resources has facilitated the development and deployment of complex ML models that can achieve high levels of accuracy and performance on various tasks. For example, Deep Learning (DL), a branch of ML that uses multiple layers of artificial neural networks to learn from data, has achieved remarkable results in domains such as computer vision, natural language processing, speech recognition, and natural language generation [3]. However, these models often lack explainability and transparency, meaning that they do not provide understandable and meaningful reasons for their decisions or outputs. This poses a challenge for users and stakeholders who need to trust, verify, and understand the ML models [4].

The lack of explainability and transparency of ML models also limits their potential for further innovation and improvement. For example, debugging, optimising, or generalising a model to new situations or data without knowing how and why it works is not easy. Moreover, it is hard to incorporate human feedback or domain knowledge into the model to enhance its performance or robustness. Furthermore, it is challenging to ensure ethical and legal compliance with the model concerning fairness, privacy, accountability, or safety [5]. Therefore, there is a need to develop methods and techniques that can explain ML models' behaviour and outcomes.

Explainability is becoming essential for AI products, particularly in high-stakes industries such as healthcare and finance. In healthcare, explainable AI can help clinicians make better decisions for diagnosis, prognosis, treatment planning, and patient education [6]. For example, an AI system that can detect diabetic retinopathy should also provide an explanation of how it reached its diagnosis and what features it used. In finance, explainable AI can help investors, regulators, auditors, and customers understand the risks and opportunities of financial products and services [7]. For example, an AI system recommending optimal portfolios or loan approvals should also explain how it calculated the expected returns or credit scores [8].

To address the challenge of explainability and transparency of ML models, various approaches have been proposed to provide explanations for their decisions or outputs. These approaches can be broadly classified into two categories: ante-hoc and post-hoc [4]. Ante-hoc approaches aim to design ML models that are inherently interpretable and transparent, such as decision trees, rule-based systems, or linear models. Post-hoc approaches aim to generate explanations for existing ML models that are not interpretable by themselves, such as deep neural networks or ensemble methods. Post-hoc approaches can be further divided into model-specific and model-agnostic methods. Model-specific methods exploit a specific ML model's internal structure and parameters to generate explanations, such as saliency maps or activation maximisation for deep neural networks [9]. Model-agnostic methods can be applied to any ML model, regardless of its complexity or architecture, by analysing the input-output pairs of the model and extracting relevant features or patterns, such as Local Interpretable Model-agnostic Explanations (LIME) or SHAP [10, 11]. Post-hoc approaches are more flexible and widely applicable than ante-hoc approaches, as they can provide explanations for any ML model without sacrificing its performance or accuracy.

In addition to the classification of post-hoc approaches as model-specific and model-agnostic, another important distinction is between local and global explanations. Local explanations aim to provide insights into how the model arrived at a specific prediction for a particular input instance, while global explanations aim to provide insights into the model's overall behaviour and decision-making process across all input instances.

Post-hoc local explanation methods generate explanations for ML models after they have been trained and deployed. One common approach for these methods is to generate neighbours for the instance to explain, i.e., similar instances with slightly perturbed features, and use them for training an interpretable ML model, such as a linear model or a decision tree. Then, the explanation for the instance can be extracted from the interpretable model, such as the coefficients or the rules. These methods have gained much attention recently due to their effectiveness, flexibility, and compatibility with existing ML models. However, they also have limitations and challenges, such as the trade-off between fidelity and interpretability, their dependence on the black-box models' outputs, their sensitivity to the intelligent design of the neighbour's generation method and the choice of the interpretable ML model, and their potential for misinterpretation or manipulation [12, 13]. Therefore, further research is needed to evaluate and improve these methods' reliability, accuracy, and usability.

Furthermore, one of the main challenges of XAI is to provide both factual and counterfactual explanations for the users of AI systems. Factual explanations reveal the reasons behind a decision made by a black-box classifier, while counterfactual explanations suggest how to change the input instance to obtain a different outcome.

1.1.2 Fuzzy-based Machine Learning Systems: Advantages and Challenges in Explainability

Fuzzy-based ML systems are a class of ML models that use fuzzy logic to handle uncertainty and imprecision in data [14]. Fuzzy logic is a type of multi-valued logic that permits degrees of truth instead of binary true or false values. It can capture intricate and nonlinear relationships in data and provide interpretable rules or linguistic terms for their decisions [15]. Fuzzy-based ML systems include fuzzy decision trees, fuzzy rule-based systems, fuzzy clustering, fuzzy neural networks, and fuzzy random forests, to name a few examples.

Additionally, fuzzy-based ML systems offer several advantages over traditional ML systems. First, they can handle vague, incomplete, or noisy data by utilising membership functions and similarity measures to assign degrees of belonging to fuzzy sets [15]. Second, they can incorporate human expertise and knowledge into the model by using linguistic variables and fuzzy rules [16]. Third, they can adapt to dynamic environments or data by using learning mechanisms or tuning methods. Fourth, to some extent, they can produce transparent and explainable outputs based on defuzzification techniques or aggregation operators [17].

Hence, fuzzy-based ML systems have been applied to various domains, especially in healthcare, where uncertainty and interpretability are crucial. A recent example of fuzzy-based ML systems in healthcare is RETIPROGRAM, a system developed by our research group to help clinicians estimate the personalized risk of developing diabetic retinopathy as early as possible. The AI core of RETIPROGRAM is a fuzzy random forest composed of 100 fuzzy decision trees [18].

Nevertheless, ML systems that employ fuzzy logic are not exempt from the challenge of explainability. As these systems become more complex and advanced, their interpretability and transparency may diminish. For example, fuzzy decision trees may grow too large or have too many branches to be easily understood by humans. Fuzzy rule-based systems may have conflicting or redundant rules affecting consistency and reliability. Fuzzy clustering may produce clusters that need to be more well-defined or meaningful for the domain [17].

In our research group, we have discovered that RETIPROGRAM is a complex system that can be challenging to understand, as it consists of 100 fuzzy decision trees, and the ultimate decision is determined by complex non-invertible aggregation techniques. Consequently, there is a requirement to establish explanation techniques for ML systems that employ fuzzy logic to enhance their transparency and interpretability.

Although XAI has recently attracted researchers and there are numerous works that can be found in the literature, only a few works have been done towards developing methods to explain ML systems that use fuzzy logic or to utilize fuzzy techniques to develop explanation methods. Nevertheless, these works are limited to specific types of ML systems that use fuzzy logic and do not provide a general framework or methodology for XAI in fuzzy logic. Therefore, a gap in the literature needs to be filled by proposing novel explanation methods for ML systems that use fuzzy logic.

1.2 Objectives

The main objective of this thesis is to develop novel explanation methods for fuzzy-based ML systems that can provide both factual and counterfactual reliable explanations. Therefore, we propose to follow the post-hoc explanation framework focusing on its two main components, namely, the neighbour generation method and the selection of the interpretable ML model. To accomplish this objective, we have formulated the following specific objectives:

1. To analyse the behaviour of the neighbour generation methods used by the LORE method and other XAI methods, identify their shortcomings, and propose novel generation methods that can improve the quality of the explanations.

2. To compare the proposed generation methods with different explainable ML models on real-world applications, namely RETIPROGRAM, the ML system based on fuzzy logic for predicting the risk of diabetic retinopathy.

3. To propose a new explanation framework for XAI that employs fuzzy techniques to provide factual and counterfactual explanations for various types of fuzzy-based ML systems.

4. To develop a new method that uses fuzzy logic to explain complex multi-class classifiers, such as fuzzy random forests.

5. To evaluate the effectiveness and usefulness of the proposed explanation methods using real-world datasets and case studies in the healthcare domain.

6. To compare and contrast the proposed explanation methods with existing XAI methods and provide insights and recommendations for future research.

1.3 Contributions and Scientific Dissemination

In response to the main objective of this thesis, which is to develop novel explanation methods for fuzzy-based ML systems that can provide both factual and counterfactual explanations, the main contributions of this PhD thesis are as follows:

1. We proposed two novel explanation methods, namely Guided-LORE and Contextualized-LORE, both of which focus on the first component of the post-hoc explanation framework, the generation of neighbours method. In Guided-LORE, we formulated the generation process as a search problem and developed an algorithm to solve it using Uniform Cost Search. Compared to other generation methods in the literature, this approach has an advantage in terms of utilizing knowledge about the characteristics of the input features, resulting in the generation of neighbours that are dense, compact, and have a clear decision boundary. On the other hand, Contextualized-LORE is designed particularly to target fuzzy-based ML systems, which is the main objective of this thesis, but it can also be used for other ML systems. It is a variation of the Guided-LORE method that explicitly considers cases where the attributes that define the objects are fuzzy. **Chapter 3** describes and explains these methods in detail.

   The outcomes of this research have been disseminated through publications in the following papers:

   - Najlaa Maaroof, Antonio Moreno, Aida Valls, and Mohammed Jabreel. "Guided-LORE: Improving LORE with a Focused Search of Neighbours", In Trustworthy AI-Integrating Learning, Optimization and Reasoning: First International Workshop, TAILOR 2020, Virtual Event, Springer. (pp. 114-127).

- Najlaa Maaroof, Antonio Moreno, Aida Valls, and Mohammed Jabreel. "Contextualized LORE for Fuzzy Attributes", In Artificial Intelligence Research and Development, IOS Press, 2021. (pp. 435-444).

2. We studied and analysed two distinct methods for generating rules in the C-LORE-F method. One method utilised crisp decision trees, while the other involved constructing preferential decision rules using the Dominance-Based Rough Set Approach (DRSA) [19]. Both methods were used to generate explanations for the RETIPROGRAM classifier and evaluate the effectiveness of the proposed neighbours' generation method in Contextualized-LORE when we change the interpretable ML model. We also provided a comparative study of two rule-based explanation methods for assessing the risk of Diabetic Retinopathy. **Chapter 4** provides a detailed explanation of this work.

   The outcomes of this study were published in the following paper:

   - Najlaa Maaroof, Antonio Moreno, Aida Valls, Mohammed Jabreel, and Marcin Szeląg. "A Comparative Study of Two Rule-Based Explanation Methods for Diabetic Retinopathy Risk Assessment", Applied Sciences, 12.7 (2022).

3. We introduced a novel approach, called Fuzzy-LORE, to enhance the quality of explanations for fuzzy-based ML systems. Fuzzy-LORE builds upon our previous method, i.e., Contextualized-LORE, by employing the fuzzy decision tree as an ML interpretable model instead of the classical decision tree. Therefore, the Fuzzy-LORE method fully incorporates fuzzy logic techniques throughout the process, from generating neighbours to extracting a meaningful explanation that includes decision rules, counterfactual rules, and counterfactual examples. **Chapter 5** provides a detailed explanation of this method.

The findings of this research have been disseminated through a publication in which the method was presented and received the **Best Paper Award** at the 24th International Conference of the Catalan Association for Artificial Intelligence (CCIA 2022). The publication is as follows [20]:

- Najlaa Maaroof, Antonio Moreno, Mohammed Jabreel, and Aida Valls. "Fuzzy-LORE: A Method for Extracting Local and Counterfactual Explanations Using Fuzzy Decision Trees", In Artificial Intelligence Research and Development, IOS Press, 2022, (pp. 345–354).

4. We proposed a novel method called multi-class Fuzzy-LORE (mcFuzzy-LORE), which extends Fuzzy-LORE to provide explanations for multi-class fuzzy-based classifiers such as fuzzy random forests. It could also be applied to explain binary fuzzy-based classifiers, as they are a special case of multi-class classifiers. We evaluated the proposed method on a private dataset that was utilised for training an FRF-based multi-class classifier that assesses the risk of developing diabetic retinopathy in diabetic patients. **Chapter 6** describes and explains this method in detail.

   The outcomes of this research have been disseminated through publications in the following papers:

   - Najlaa Maaroof, Antonio Moreno, Aida Valls, Mohammed Jabreel, and Pedro Romero. "multi-class Fuzzy-LORE : A method for Extracting Local and Counterfactual Explanations using Fuzzy Decision Trees", In Electronics. 2023; 12(10):2215.

The following chapter, namely **Chapter 2**, of this thesis provides an introduction to Explainable Artificial Intelligence (XAI). It presents a comprehensive overview of its significance in various applications, including healthcare, finance, and autonomous vehicles. This chapter also covers the different categories of XAI techniques, such as model-specific and model-agnostic methods. Furthermore, it explains how they can be utilised to generate interpretable explanations for ML models. Additionally, the chapter introduces the main concepts used throughout the thesis, briefly reviews related works in the field of XAI, and discusses their limitations and shortcomings, which motivate our research. Finally, it defines the evaluation metrics used in this work.

The thesis concludes with **Chapter 7**, which summarises the main contributions and achievements of our research and discusses the limitations and advantages of our proposed methods, as well as potential future research directions in the field of XAI for fuzzy-based ML systems.

Chapter 2

Background

This chapter presents a comprehensive overview of relevant concepts that contextualize the contributions of this thesis. It begins by outlining the diverse definitions of explainability and exploring the underlying reasons and motivations for pursuing it. The chapter also discusses the general challenges in Explainable Artificial Intelligence (XAI) and outlines the methodologies employed to attain it.

2.1 Explainability in Artificial Intelligence

Explainability has emerged as a critical challenge in developing and deploying AI systems. Despite early efforts to develop explainable AI systems, the rapid advances in ML in recent years have shifted the focus of AI research towards developing models and algorithms that prioritise predictive accuracy over interpretability. This has led to an increased demand for solutions that address the lack of transparency and complexity associated with these systems.

One proposed solution is the development of XAI techniques, which aim to create more transparent AI systems that enable end-users to understand, trust and manage these systems effectively. The term "XAI" was first introduced by Van Lent, Fisher, and Mancuso in 2004 [22], although the problem of explainability has been studied since the mid-1970s when researchers explored explanation mechanisms for expert systems [23]. According to [24], XAI focuses on developing explainable techniques that empower end-users in comprehending, trusting, and efficiently managing the new era of AI systems. One can use the definition of XAI coined by D. Gunning [25] "XAI will create a suite of machine learning techniques that enables human users to understand, appropriately trust, and effectively manage the emerging generation of artificially intelligent partners."

It is important to note that, despite the various definitions put forward for explainability, there remains considerable ambiguity and lack of clarity regarding the definitions of key terms such as "interpretability" and "explainability," which are used interchangeably by many researchers. However, several attempts have been made to clarify these two terms and related concepts such as comprehensibility. For example, Lombrozo [26] presented a definition for explanations grounded in psychology, proposing that explanations facilitate the exchange of beliefs among individuals. Specifically, they defined *explanations* as "the currency in which we exchange beliefs." According to FAT [27], the goal of enabling ML explainability "is to ensure that algorithmic decisions and the corresponding data can be conveyed to end-users and other stakeholders using non-technical terminology." One of the most widespread definitions of interpretability is that of Doshi-Velez and Kim [28], who defined it as "the ability to explain or to present in understandable terms to a human."

In this thesis, we have adopted the terminology used by Rudin [29], who distinguish between interpretable and explainable ML. Interpretable ML refers to models that are inherently transparent and easy to understand, while explainable ML focuses on providing post hoc explanations for opaque or proprietary black-box models.

2.1.1 Why is Explainable AI Needed?

Nowadays, black-box AI systems have become ubiquitous in decision-making domains such as customer services, social networks, and medical systems. Unfortunately, most of these systems make decisions without providing explanations for them.

However, Adadi and Berrada [5] argued that not all black-box AI systems need to provide explanations for their decisions, as doing so could result in increased costs and reduced efficiency. Specifically, they outlined two situations in which explainability and interpretability are not always necessary: (1) when results that are deemed unacceptable do not lead to significant consequences, such as recommendation systems; and (2) when the problem has been thoroughly researched and tested in practice, making the decision made by the black-box system trustworthy, such as with the extensively researched and rigorously validated language translation system, which generates highly reliable outputs.

Hence, it is crucial to consider the cases under which explanations can be beneficial [5]. The XAI literature offers a range of viewpoints on the necessity of explainable AI, as illustrated in Figure 2.1. These perspectives are, to some extent, distinct, although there may be some overlap between them. Nevertheless, they emphasise the most significant rationales for why XAI is essential.

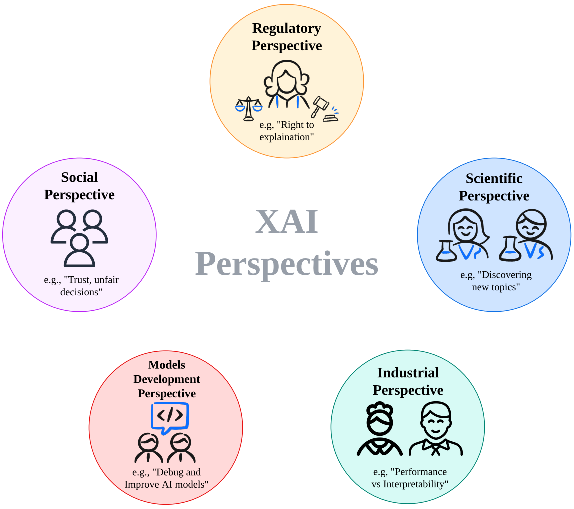

Regulatory Perspective
e.g, "Right to explaination"
Social Perspective
e.g., "Trust, unfair decisions"
XAI Perspectives
Scientific Perspective
e.g, "Discovering new topics"
Models Development Perspective
e.g., "Debug and Improve AI models"
Industrial Perspective
e.g, "Performance vs Interpretability"

FIGURE 2.1: The five main perspectives for the need for XAI.

Regulatory Perspective: The widespread use of AI systems across various industries has raised concerns about compliance with regulations and laws, particularly those related to data privacy and automated decision-making. The European Union's General Data Protection Regulation (GDPR) is a prime example of why XAI is needed from a regulatory standpoint. These regulations mandate the "right to explanation," which enables a user to demand an explanation regarding an algorithm's decision that significantly affects them [30], a challenging requirement for black-box AI systems.

XAI can aid in ensuring compliance with these regulations and laws by providing clear and understandable explanations of how an AI system arrived at a decision or output. For instance, consider an AI system that assesses job applications and makes hiring decisions. An applicant may request justifications for the decision made to ensure that it aligns with other regulations and laws [31]. With XAI techniques, the system can explain how it reached a particular decision, such as why one candidate was chosen over another. This explanation can be used to adhere to regulations that necessitate transparency in automated decision-making and can also aid in establishing trust with job applicants who are concerned about bias or discrimination.

Model's Developmental Perspective: Understanding the decision-making processes in black-box AI systems is crucial, particularly when their outputs can impact human lives. Several factors, such as limited training data, biased training data, outliers, adversarial data, and model overfitting, can lead to inappropriate results. XAI can help to understand what black-box AI systems have learned, debug them, and enhance their robustness, user trust, and safety while minimizing or preventing discrimination, bias, and faulty behaviour [32].

For instance, XAI can help in identifying and mitigating biases that can arise during the development of ML models. By explaining the model's decisions, XAI can enable data scientists to identify the sources of bias and modify the model's features to mitigate them. This can help prevent the model from making unfair or discriminatory decisions.

Furthermore, the development of ML models involves stakeholders with different expertise, such as data scientists, domain experts, and business analysts. XAI can facilitate communication between these stakeholders and help them understand the model's decision-making process. This, in turn, can improve collaboration and enable stakeholders to contribute to the model development process effectively.

In addition, when comparing models with similar performance, XAI can assist in model selection by revealing the features that different models use to make decisions [33, 34]. Using XAI techniques, it is possible to identify the strengths and weaknesses of each model and select the one that aligns best with the problem at hand. This can also help to ensure that the selected model is transparent and interpretable, providing insights into its decision-making process and improving user trust.

Scientific Perspective: From a scientific perspective, XAI is crucial because it can facilitate a deeper understanding of black-box AI models and reveal the scientific knowledge they extract. The purpose of constructing black-box AI models is to generate an approximate function to solve specific problems. However, these models contain information not explicitly present in the input data but inferred by the model during training, making it difficult to understand the factors contributing to its decision-making process [35].

By applying XAI techniques, researchers can comprehend the inner workings of black-box AI models and identify the scientific knowledge they extract. This can enable the discovery of innovative concepts across diverse scientific domains. For example, XAI can help identify hidden patterns and relationships in large datasets, which can be used to develop new scientific theories or support existing ones. Additionally, XAI can enable researchers to identify errors and inconsistencies in data that might otherwise go unnoticed, thereby improving the accuracy and reliability of scientific models.

Overall, XAI is critical from a scientific perspective because it can help researchers understand the factors influencing the decisions made by AI models, reveal the knowledge extracted by these models, and enable the discovery of innovative concepts and theories across diverse scientific domains.

Industrial Perspective: In the industrial domain, black-box AI models are increasingly utilised for a range of applications, including image recognition, natural language processing and predictive maintenance. Despite providing high accuracy and performance, these models also pose significant risks to the reliability, safety and compliance of systems. This is particularly concerning in regulated industries such as healthcare and autonomous vehicles, where model outputs can impact human lives. Therefore, adopting explainable AI techniques can help enhance transparency and accountability of black-box AI models, thus improving user trust [35].

Consequently, the use of explainable AI techniques can also facilitate the adoption of black-box AI models within industry. By understanding the model's decision-making process, stakeholders can make informed decisions about the model's suitability for a particular application. Additionally, XAI can help identify and address biases in the model, ensuring fairness and mitigating potential legal issues.

Furthermore, XAI can aid in model maintenance and monitoring. As the model's input data and operating environment change over time, its performance can deteriorate, and its outputs may become unreliable. By utilising XAI techniques to understand the model's inner workings, it is possible to identify sources of underperformance and update the model accordingly, ensuring continued accuracy and reliability.

End-user and Social Perspective: End-users and society, in general, are also essential stakeholders in developing and deploying AI systems. Therefore, using black-box models in critical domains such as healthcare, finance, and criminal justice can have significant social and ethical implications. For example, a wrong diagnosis or decision made by an AI system could result in misdiagnosis, wrongful conviction, or a discriminatory outcome [36].

Thus, the need for transparency and accountability in the decision-making process of AI systems is critical to ensuring that end-users and society trust the technology. XAI can provide users with insights into the model's inner workings, thereby enabling them to understand the system's decision-making process [36, 37].

Furthermore, XAI can help promote responsible AI development and deployment. By providing end-users with insights into the model's decision-making process, they can detect and address any potential risks or issues arising from the system's use [38].

Now that we have discussed the need for XAI from various perspectives let us examine the challenges associated with XAI.

2.1.2 XAI Challenges

Despite the potential benefits of XAI, there are significant challenges that need addressing to develop effective and trustworthy explainable AI systems.

This subsection discusses these challenges and highlights the need for interdisciplinary research and collaboration between computer scientists, ethicists, legal scholars, and other experts to address the complex and multifaceted issues related to XAI.

Standardisation and formalisation: This is one of the major challenges highlighted in the literature on XAI [5, 36, 39, 40].

Despite extensive research on explainability from various perspectives, there is still no consensus on a standardized definition of explainability within the research community. This can make it challenging for researchers and technicians to develop and design XAI systems and solutions that are guided by a broad base. It is crucial to have a shared understanding of what explainability means and entails to enable effective communication and collaboration between stakeholders in different domains.

One approach to addressing this challenge is to develop a framework that characterises explainability from different angles, including data, model, learning strategy, outcomes, and gradient information. This will provide a more comprehensive understanding of the different aspects of explainability and enable researchers to identify, classify, and evaluate sub-issues of explainability systematically. Such a framework will also provide a basis for standardising the evaluation of XAI techniques, making it easier to compare and select between different methods [5].

Moreover, the lack of standardisation and formalisation also makes it difficult to integrate explainability methods into a standardised framework. Existing research provides valuable information on explainability, but evaluating them systematically can take time and effort. Therefore, there is a need to develop a unified framework that can enable researchers to integrate explainability methods into AI systems in a standardised way [39].

Overall, standardisation and formalisation are crucial challenges that need to be addressed to enable the development and deployment of trustworthy and effective XAI systems.

Interpretability vs. Performance Trade-off: Another challenge facing the development of XAI is the trade-off between interpretability and performance [39]. In Machine Learning, accuracy and interpretability are often competing objectives, and there is often a trade-off between the two. Highly accurate models may be too complex to interpret and explain, making it difficult for end-users to understand and trust the system. Conversely, highly interpretable models may sacrifice some accuracy, reducing their effectiveness in certain applications. A visual representation of the trade-off between explainability and performance is provided in Figure 2.2.

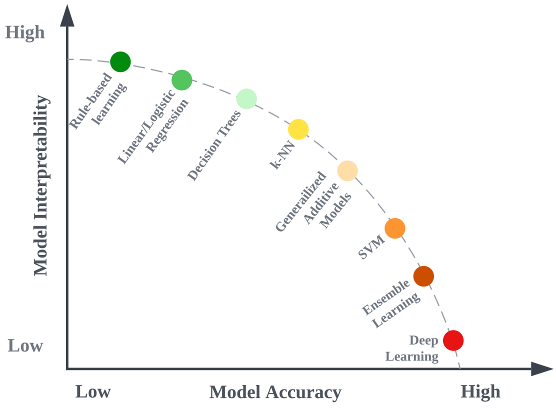


FIGURE 2.2: Trade-off between Explainability and Performance.

This trade-off is particularly relevant in sensitive domains such as healthcare and finance, where the decisions made by the AI system can have significant consequences. In these domains, end-users need to understand why a particular decision was made, and they must have confidence in the system's reliability. However, achieving high accuracy in these applications is also critical. For instance, accurate predictions can be essential in helping healthcare professionals diagnose and treat patients effectively.

To address this challenge, researchers are exploring ways to balance the need for accuracy with the need for interpretability. One approach is to develop hybrid models that combine the strengths of both interpretable and complex models [40]. For example, an interpretable model can be used to provide explanations for the decisions made by a complex model. Another approach is to use ensemble methods, where multiple models with different levels of interpretability are combined to achieve high accuracy while maintaining interpretability.

Challenges in Evaluating Explainability in AI: The lack of standardisation and evaluation frameworks for XAI techniques is a significant challenge facing the development of explainable AI. Several studies have called for a common set of evaluation metrics and benchmarks to assess the performance and effectiveness of different explainability techniques [5, 41, 39].

There are two main approaches for evaluating the interpretability of machine learning models: objective metrics and human-centred evaluation methods [39]. While objective metrics are quantifiable mathematical metrics, human-centred evaluations rely on feedback from users (usually domain experts) [39]. However, there is no consensus on which metrics should be used or how they should be applied, and the lack of standardisation makes comparing and selecting different XAI approaches difficult. Furthermore, rigorous evaluations are challenging in most cases due to the lack of ground truth [39, 42]. In addition, the quality of an explanation is subjective and context-dependent, making it challenging to define what constitutes a "good" or "sufficient" explanation [4]. An explanation that is adequate for one user may not be sufficient for another, and it may also depend on the complexity of the underlying model.

Standardised evaluation frameworks and metrics for XAI techniques are necessary to advance the field and build trust in AI systems. By establishing common standards for evaluating interpretability, researchers, and practitioners can compare and select XAI approaches more effectively, leading to more transparent and trustworthy AI systems. To achieve this, there is a need to define objective metrics for evaluating XAI approaches in various contexts, models, and applications [4]. In addition, a model-agnostic framework that suggests the most appropriate explanation considering the problem domain, the use case, and the user type could also be beneficial [43].

Preserving Privacy in Explainable AI: One of the primary benefits of XAI algorithms is their interpretability, which makes them more transparent and easier for humans to understand. However, this feature can also lead to potential privacy breaches as the interpretability of XAI algorithms may reveal sensitive information about individuals [44]. Therefore, it is crucial to investigate and ensure that XAI algorithms do not compromise data privacy during training or inference.

To address the challenge of the privacy-explainability trade-off, new methods are needed for evaluating the privacy impact of XAI algorithms. These methods should provide an automated way to assess the severity of privacy breaches [45]. By using this metric, it is possible to develop new privacy-preserving techniques and regulations that can mitigate the risks posed by XAI algorithms to individuals' privacy.

This work focuses on addressing the interpretability-performance trade-off, which is a challenge XAI encounters (the second category stated earlier). The goal is to develop post-hoc local explanation methods for fuzzy-based ML systems to increase interpretability while maintaining the model's performance. This is done by keeping the original model unchanged and incorporating further interpretability.

The next subsection will examine several techniques that can be utilised to overcome the challenges mentioned earlier.

2.2 Techniques and Approaches for XAI

In the field of XAI, there is a clear distinction between models that are designed to be interpretable and those that require external techniques to be explained. This is sometimes referred to as the difference between interpretable models and model explainability techniques, or more commonly as transparent models and post-hoc explainability [36, 5, 46, 4, 47, 43].

The idea of transparent models is that they are designed to be inherently interpretable, with their inner workings and decision-making processes easily understandable by humans. Examples of transparent models include decision trees, rules [48, 49], additive models [50], or sparse linear models[51].

On the other hand, post-hoc explainability techniques aim to explain the decision-making process of black-box models after they have been trained and deployed. These techniques often involve generating visualizations or feature importance rankings to help users understand how the model arrived at its decision.

2.2.1 Transparent Machine Learning Models

Transparent machine learning models, also referred to as interpretable models, are designed to be inherently understandable by humans. These models are generally less complex than their black-box counterparts and have a clear decision-making process that can be traced and understood. Decision trees, linear regression, and logistic regression are some examples of transparent models.

One of the main advantages of transparent models is their interpretability. Users can easily understand why a specific decision was made and identify any biases or errors in the model since the decision-making process is transparent. This transparency also enables users to make informed decisions about the use of the model in various applications.

However, transparent models also have their limitations. They may not always provide the highest accuracy, particularly in complex tasks where non-linear relationships are present. Additionally, they may not be well-suited for processing large datasets or handling high-dimensional data.

Despite these limitations, transparent models remain an important area of research in XAI. They offer a valuable alternative to black-box models and can be particularly useful in applications where interpretability is crucial.

2.2.2 Post-hoc Explanation Techniques

In cases where ML models do not meet the criteria for being transparent, post-hoc explanation techniques are employed. These techniques aim to provide understandable information on how a model generates predictions for a given input. This is achieved by developing a separate method to explain the decision-making process of the model after it has been developed.

Most post-hoc explanation methods generate, in some way, a set of inputs, analyze the answers provided by the black box to them, and then construct a simpler model form in which an explanation can be inferred. The taxonomy of post-hoc explanation techniques can be divided based on the type of algorithms that could be applied. If their application is only restricted to a specific family of algorithms, then these methods are called model-specific. In contrast, the methods that could be applied in every possible algorithm are called model agnostic.

Model-dependent techniques are those that are specifically designed for a certain type of model, or that rely on specific assumptions about the model's structure or behaviour. For example, decision trees and rule extraction are model-dependent techniques, as they rely on the assumption that the model is a decision tree or can be represented as a set of rules.

Model-agnostic techniques can be applied to any machine learning model, regardless of its architecture, complexity, or nature of the data it uses.

In addition to the classification of post-hoc approaches as model-specific and model-agnostic, another important distinction is between global and local explanations.

Global explanation methods explain the overall behaviour of the model. These methods provide insights into how the model makes decisions across the entire dataset and can help identify patterns and trends in the data.

Local explanation methods, on the other hand, are used to explain individual predictions made by the model. These methods provide explanations that help to understand how the model arrived at a particular decision for a specific instance.

LIME [10] is widely recognized as one of the most prominent local explanation methods. The main concept behind LIME involves creating a locally-weighted interpretable linear model within the vicinity of a specific observation. When explaining the ML prediction of a particular example, LIME constructs a simple linear model around that prediction. To train this model, random data points are generated from the distribution of the training dataset. These data points are weighted based on their distance from the reference point, which is the prediction being explained by LIME. To ensure clarity and simplicity, feature selection is applied, resulting in the retention of only the most important variables. The coefficients of these selected variables are then considered as explanations for the prediction.

Another popular local explanation methods in the literature is Local

Rule-Based Explanations (LORE) [52]. LORE generates simple, human-readable rules that explain how a model's predictions are made for a specific instance. These rules can help build trust in black box models and provide insights into decision-making processes. LORE is a model-agnostic method that uses decision trees to generate local explanations for black box binary classifiers. A local explanation consists of a factual rule that states the reasons for the decision and a set of counterfactual rules that suggest how to change it. LORE uses a genetic algorithm to generate synthetic instances that mimic the behaviour of the black box locally and then learns a decision tree from them. Finally, the decision tree is compiled into a set of rules for extracting an explanation. In this thesis, LORE serves as the foundation of the methods presented. So, in the following chapter, we explain in details the main steps of LORE including our novel proposed changes.

2.3 Chapter summary

This chapter introduced fundamental concepts and background related to the thesis topic, such as Explainable AI and its motivation, challenges, and techniques. The chapter provided a comprehensive overview of the diverse definitions of explainability and explored the underlying reasons and motivations for pursuing it. The challenges in XAI were discussed, and the methodologies employed to overcome them were outlined. The following chapters will explain the contribution of this thesis towards developing XAI methods that can provide high-quality factual and counterfactual explanations specifically for fuzzy-based ML systems.

Chapter 3

Neighborhood Generation Techniques for Improved Local Surrogate Models

3.1 Introduction

One of the most commonly used techniques for providing local post-hoc explanations involves creating an interpretable surrogate model that imitates the behaviour of the black-box model and explains its predictions for a particular input. This approach often involves generating neighbourhoods around a specific input point, which are constructed by perturbing the input features in a meaningful way to create new data points that are close to the original one. By training the surrogate model on these neighbourhoods, it can capture the behaviour of the original model in the vicinity of the input point and provide insights into how the model arrived at its decision. Therefore, generating neighbourhoods is a crucial step in constructing local interpretable surrogate models for post-hoc explanations.

However, traditional approaches such as LIME [10] and LORE [52] have limitations when it comes to generating neighbourhoods, which can affect the quality of the explanations provided. Specifically, LIME uses a random sampling method to generate neighbourhoods, which may result in "instability" in the generated explanations, where for the same prediction, different explanations can be generated [53, 54, 55]. On the other hand, LORE relies on genetic algorithms, which can be computationally expensive and time-consuming, particularly when dealing with high-dimensional data. Furthermore, similar to LIME, it may suffer from instability or inconsistency due to the randomness of the genetic algorithm. In addition, the neighbours generated using either LIME or LORE may contain features with out-of-bounds values. For instance, in some cases, the value of the "age" feature may be negative. Finally, the generation process of both methods is not knowledge-driven and lacks context, particularly the characteristics of the features. For example, in some cases, it may be more beneficial to keep some features unchangeable (e.g. gender) or allow only positive changes to some features (e.g. age).

This chapter presents two novel methods designed to improve the quality of generated neighbours, which are crucial in constructing local interpretable surrogate models and deriving accurate post-hoc explanations. By generating high-quality neighbourhoods, the surrogate models produced by these methods can more accurately capture the behaviour of the original model and provide more meaningful insights into its decision-making process.

We start by defining the black box outcome explanation problem in section 3.2. The first method, described in 3.3, is called "Guided LORE," which utilises a guided search to identify relevant features and perturb them in a meaningful way, resulting in improved quality of generated neighbours. The second method, described in 3.4, is "Contextualized LORE for Fuzzy Attributes," which builds upon the Guided LORE method but extends it to handle fuzzy attributes. Fuzzy attributes present a significant challenge to existing post-hoc explanation techniques due to their imprecise and ambiguous nature. This method is specifically designed to address this challenge and generate high-quality neighbourhoods for models that involve fuzzy attributes.

3.2 Problem Definition

Given a black box classifier b and an instance x, our goal is to provide an explanation for the decision $y = b(x)$ in the form of a triplet $(\mathcal{R}, \Delta, \mathcal{C})$, where:

- $\mathcal{R}$ is the decision rule that covers the instance x. This rule tells which are the sufficient conditions to be satisfied by the object for being classified as y, so they indicate the minimal reasons for belonging to that class.
- Δ is the set of counterfactual rules that lead to an outcome different than the one of x. They indicate the minimal number of conditions that should be simultaneously changed in the object for not being in class y.
- $\mathcal{C}$ is a set of counterfactual instances that represent examples of objects that belong to a different class and have the minimum changes with respect to the original example x.

3.3 Guided-LORE: improving LORE with a Focused Search of Neighbours

We propose a variation of LORE, called "Guided-LORE", which introduces a new method for generating neighbours and constructing local explanations for complex black box predictors. Like LORE, Guided-LORE follows a set of steps, as shown in Algorithm 1, that aim to generate an explanation for the decision made by a black box system b, based on a given example x and its outcome y. Firstly, we utilise the uniform-cost search algorithm (described in Algorithm 2) to generate two sets of examples: the positive set D^+, which contains examples that are close to x and belong to the same class, and the negative set D^-, which contains examples that are close to x but have a different class.

As it will be seen later, a neighbour of a point p will be generated by making a small positive or negative change in the value of a feature of p, allowing for an exhaustive search of all possible points in the vicinity of x. To obtain the negative set, we look at an auxiliary set T and identify the closest example to x, x^-, with a different label than y using the procedure *FindDiffExample* on line 3. T can either be the training set used to train the black-box model or any other dataset from the same distribution. Once we have obtained x^-, we pass it to Algorithm 2 to generate the negative set.

Algorithm 1: Guided-LORE

```
Input  : $x$: an instance to explain, $T$: an auxiliary set, $b$: a
         black-box model, $L$: maximum level of exploration, and
         $KB$: knowledge base.
Output: $E$: the explanation of the decision of $b$ on $x$
1 $y \longleftarrow b(x)$
2 $D^+ \longleftarrow GetNeighbours(x, y, b, L, KB)$
3 $x^-, y^- \longleftarrow FindDiffExample(x, y, b, T)$
4 $D^- \longleftarrow GetNeighbours(x^-, y^-, b, L, KB)$
5 $D \longleftarrow D^+ \cup D^-$
6 $t \longleftarrow BuildTree(D)$
7 $\mathcal{R} = (p \rightarrow y) \longleftarrow ExtractRule(x, t)$
8 $\Delta, \mathcal{C} \longleftarrow ExtractCounterfactuals(x, \mathcal{R}, t)$
9 $E \longleftarrow (\mathcal{R}, \Delta, \mathcal{C})$
```

Once we have the two sets, we merge them to obtain the final set D. We then use the standard LORE process to train a decision tree t, which is used to produce the explanation, including the rule used by the decision tree to classify x and the set of counterfactual rules of the decision tree that produce a different outcome. It is worth noting that Guided-LORE's generation method labels the generated neighbours on-the-fly. As a result, each generated example is assigned a label obtained from the black-box model in the GetNeighbours procedure. The next subsection provides a more detailed explanation of the proposed generation method.

3.3.1 Neighbours Generation

The task of generating neighbours is viewed as a search problem, with the aim of exploring the neighbourhood space of a point x_0. To accomplish this, we employ a Uniform Cost search based on the Heterogeneous Value Difference Metric (HVDM) [56], utilizing some knowledge (KB) about the attributes (including the maximum and minimum values, as well as the step needed to modify the value in the attribute positively or negatively in this case).

The neighbourhood generation procedure, GetNeighbours, is described by first formalising the problem as a search problem and then explaining its steps in Algorithm 2.

The neighbours generation problem can be formulated as a search problem as follows:

- **State Space**: the set of all possible examples S. If $\mathcal{F}$ is the set of features and Y is the set of labels in our problem, then we can define $S = \{(x,y) | x = (x_{f_1}, x_{f_2}, ...), for\ all\ k\ in\ 1..|\mathcal{F}|\ x_{f_k} \in range(f_k), y \in Y\}$. The range of a feature f_k depends on its type. We consider three types of attributes: nominal, numerical and ordinal.
- **Initial State**: (x_0, y_0), where x_0 is the instance of which we want to generate its neighbours and y_0 is the label of this instance calculated by the black box model.
- **Operators**: The operators represent the available actions that can be performed to obtain a neighboring instance by modifying the value of a single attribute (feature). Each feature can be associated with one or more actions, which utilize domain knowledge stored

in the KB parameter. In our case, we define three types of actions: *forward*, *backward*, and *choose*.

The *forward* and *backward* actions are used with numerical and ordinal attributes. For these attributes, we have knowledge about their range (minimum and maximum values) and a *step* value. The *step* value is used to generate the closest neighbors of an instance by adding or subtracting this value from the attribute's current value. Therefore, a numerical attribute can be increased or decreased by the *step* value, while an ordinal attribute can be changed to the next or previous value based on the selected action.

For nominal attributes, we employ the *choose* action, which generates a neighbour of an instance by randomly changing the value of the nominal attribute to another allowed value. The knowledge stored in the KB includes the set of permitted values for these attributes.

- **Transition Model**: returns a new instance in which the value of a feature $f_k \in \mathcal{F}$ is incremented by $step$ if the action is *forward*, decremented by $step$ if the action is *backward* or chosen from the set of possible values if the action is *choose*.

- **Goal Test**: This condition checks, for each generated individual, if, according to the black box, it has the same label as x_0, y_0. If that is the case, we will generate the neighbours of the individual in the same way (i.e. applying one positive/negative change in the value of a single attribute); otherwise, we have found an individual close to x_0 that belongs to another class; thus, we have reached a boundary of y_0, and we will not continue the search from that instance.

- **Path Cost**: The path cost of each node is calculated by measuring the HVDM distance between the generated example and x_0.

Algorithm 2: Guided Neighbours Generator

```
Input  : An example $x_0$, its output $y_0$, a black-box model $b$, the
         maximum level of exploration $L$, and a knowledge-base
         $KB$.
Output : The set of neighbours, $\hat{D}$.
$root \longleftarrow node\,(x_i, NULL, 0), root.label = y_0 \longleftarrow b\,(x_0)$
$q \longleftarrow [root], \hat{D} \longleftarrow []$
while Not need to stop do
    $n \longleftarrow head\,[q]$
    if $n.label = root.label$ and $n.level \leq L$ then
        add $n$ to $\hat{D}$
        foreach feature $f \in KB$ do
            if $f$ is Nominal then
                $x_c \longleftarrow copy(n.x)$
                $x_c[f] \longleftarrow choose(KB[f][range])$
                $n_c \longleftarrow node(x_c, n, n.level + 1)$
                $n_c.label \longleftarrow b(x_c)$
                $n_c.d \longleftarrow distance(x_c, x_0)$
                add $n_c$ to $q$;
            else
                $step \longleftarrow KB[f][[step]$
                $max\_value \longleftarrow KB[f][max]$
                $min\_value \longleftarrow KB[f][min]$
                if $n.x[f] + step \leq max\_value$ then
                    $x_l \longleftarrow copy(n.x)$
                    $x_l[f] \longleftarrow n.x[f] + step$
                    $n_l \longleftarrow node(x_l, n, n.level + 1)$
                    $n_l.label \longleftarrow b(x_l)$
                    $n_l.d \longleftarrow distance(x_l, x_0)$
                    add $n_l$ to $q$
                end if
                if $n.x[f] - step \geq min\_value$ then
                    $x_r \longleftarrow copy(n.x)$
                    $x_r[f] \longleftarrow n.r[f] - step$
                    $n_r \longleftarrow node(x_r, n, n.level + 1)$
                    $n_r.label \longleftarrow b(x_r)$
                    $n_r.d \longleftarrow distance(x_r, x_0)$
                    add $n_r$ to $q$;
                end if
            end if
        end foreach
    else
    end if
end while
```

Algorithm 2 shows the generation of the closest neighbours of an example x_0. As shown in Figure 3.1, the search tree starts from this example, and all the available actions to move from one example to another are applied in each node of the search tree.

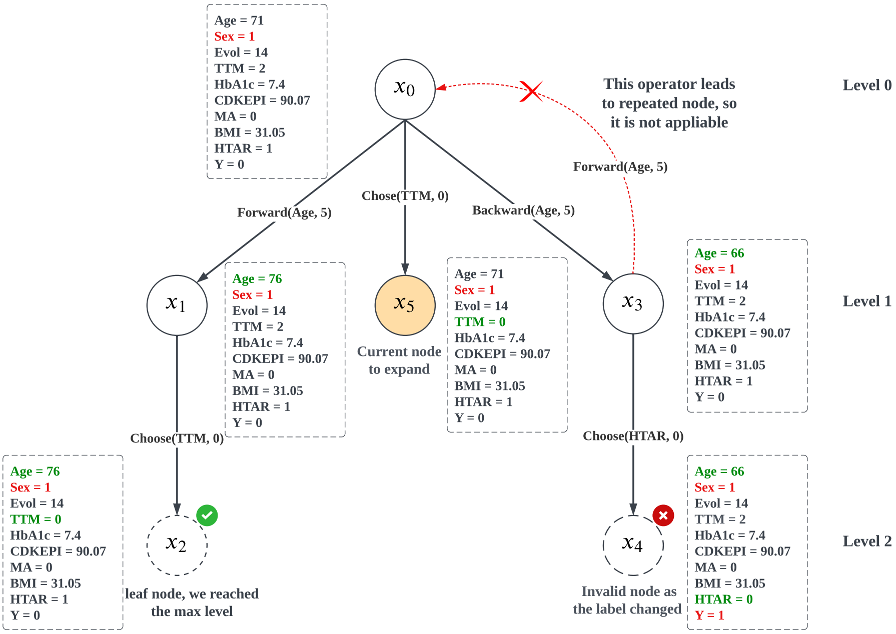


FIGURE 3.1: Illustration of the tree search. Node index refers to the order of visiting the node based on the distance to the root. Attributes in red colour are unchangeable. Attributes in green colour changed with respect to the original value. Red dashed arrow refers to inapplicable action to avoid cycles.

Each action only changes one feature by taking its value and adding some positive/negative quantity or retrieving the next/previous value (forward/backward action) or replacing it by another value (choose action). The step value for the ordinal variables is always set to one which means the forward action takes the next value whereas the backward action takes the previous one. So, in the neighbours in the first level of the tree, one feature will be changed whilst the rest remain the same. In this case, if we have, for example, five numerical features and five nominal features, we will get a maximum of fifteen neighbours (each value of a nominal feature can be changed, and the value of each numerical feature can be increased or decreased, if the new value is still in the range of the feature). If the outcome of the black box model changes in one of these nodes, then it is a leaf of the tree, and we do not expand that node further. Otherwise, we expand that node. Consequently, on the second level, we would have changes in two attributes or double changes in the same attribute, and so on. The node to be expanded in each step is the one that has the shortest path cost to the initial one, x_0. The generation process is terminated when there are no more nodes to be expanded (all the leaves have led to changes in the initial classification) or when all the nodes at the maximum level of exploration L have been expanded. Repeated nodes are ignored to avoid cycles.

3.3.2 Explanation Extraction

In this step, we utilise the generated neighbours D to train a decision tree t that mimics the local behaviour of the black-box model b on D. Subsequently, we can extract the explanation of its decision towards x and employ it to interpret the decision of b. The explanation extraction methods are similar to those of the LORE technique [52]. Initially, we extract all decision rules from the t in the form of: *IF* $condition_1$ *AND* $condition_2$ ... *AND* $condition_n$ *THEN* $decision_k$. In this case, the first element of the explanation, $\mathcal{R}$, represents a single rule activated by x.

The procedure of extracting the counterfactual rules, i.e., Δ, is described in Algorithm 3. It searches for all rules that result in a decision

Algorithm 3: Extraction of counterfactual rules

```
Input  : $\mathcal{R}$: The set of decision rules, $x$: instance to explain, and
         $y$: the decision of $x$
Output: $\Delta$: set of counterfactual rules
$Q \longleftarrow GetRulesWithDifferentDecision(\mathcal{R}, y)$
$\Delta \longleftarrow \emptyset$
$min \longleftarrow +\infty$
foreach $rule\ q \in Q$ do
    $qlen \longleftarrow nf(q, x)$
    if $qlen < min$ then
        $\Delta \longleftarrow q$
        $min \longleftarrow qlen$
    else
        if $qlen = min$ then
            $\Delta \longleftarrow \Delta \cup q$
        end if
    end if
end foreach
return $\Delta$
```

different from y and selects the rules with the fewest conditions not satisfied by x, which are returned by the function nf.

To obtain the counterfactual examples, i.e., $\mathcal{C}$, we use the counterfactual rules Δ and the original input x. Given a counterfactual rule $\hat{r} : q \rightarrow \hat{y}$ and x, we find the instance that requires the minimum changes in x to fulfil the conditions q. We identify all the attributes in the conditions q that are not satisfied by x and make the smallest modification (up or down) to the values of these attributes to satisfy the conditions in q.

As an example, let us consider the explanation for the instance presented in the top row of Table 3.1. This particular instance is chosen from the Diabetic Retinopathy dataset, which is described in detail in section 3.3.3. Assuming that this instance is classified as class 1, our approach will generate the rules depicted in Figure 3.3 from the decision tree shown in Figure 3.2.

Age	Sex	EVOL	TTM	HbA1c	CDKEPI	MA	BMI	HTAR
71.0	1	14.0	2	7.4	90.07	0.0	31.05	1
—	—	—	—	6.5	—	—	—	—
—	—	—	0	—	—	—	—	—
—	—	—	1	—	—	—	—	—

TABLE 3.1: Patient example and counterfactual instances.

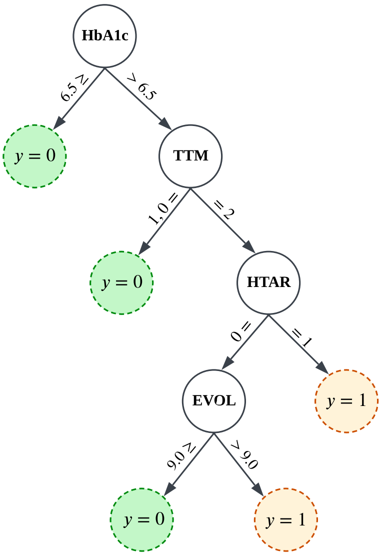


FIGURE 3.2: The decision tree for x. Dashed circles represent leaves.

$$\begin{array}{l}
R1: \{HbA1c \leq 6.5\} \rightarrow \{y = 0\} \\
R2: \{HbA1c > 6.5 \,\&\, TTM = 0\} \rightarrow \{y = 0\} \\
R3: \{HbA1c > 6.5 \,\&\, TTM = 1\} \rightarrow \{y = 0\} \\
R4: \{HbA1c > 6.5 \,\&\, TTM = 2 \,\&\, HTAR = 0 \,\&\, EVOL \leq 9.0\} \rightarrow \{y = 0\} \\
R5: \{HbA1c > 6.5 \,\&\, TTM = 2 \,\&\, HTAR = 0 \,\&\, EVOL > 9.0\} \rightarrow \{y = 1\} \\
\boxed{R6: \{HbA1c > 6.5 \,\&\, TTM = 2 \,\&\, HTAR = 1\} \rightarrow \{y = 1\}}
\end{array}$$

FIGURE 3.3: The rules constructed for x.

The activated rule, representing the first element of the explanation $\mathcal{R}$, is $R6$, which is applicable to the given example. There are four rules that lead to the opposite decision, denoted as $Q = R1, R2, R3, R4$. As the values of $nf(R1, x)$, $nf(R2, x)$, and $nf(R3, x)$ are 1, and $nf(R4, x)$ is 2, the set of counterfactual rules Δ is $R1, R2, R3$.

The final step of the explanation extraction process is constructing the set of counterfactual examples $\mathcal{C}$. We start by taking rule $R1$ and changing the values of the attributes in its condition $(HbA1c \leq 6.5)$ to the smallest value below the upper bound of HbA1c, which is 6.5, as this condition was false for patient x. We repeat this process for the remaining rules in Δ, resulting in two other examples that only require one change in the TTM variable, as the remaining conditions are already satisfied by x. The obtained counterfactual instances are shown in rows 2-4 of Table 3.1, where empty cells indicate that the initial value of the attribute has not been changed. We can observe that the number of changes in the counterexamples is very small in this case (just 1). While we could generate other CF examples with two changes, such as setting HTAR to 0 and Evol to 9, this example may not hold practical significance. The reason is that the attribute "Evol" represents a measure that always increases (time ellapsed since the first diagnosis of diabetes). Consequently, decreasing its value in a CF instance would not align with its inherent nature. Therefore, this CF example is not actionable or meaningful in a real-world context where Evol should logically exhibit positive growth or advancement.

3.3.3 Experiments and Results

Experimental Setup

We evaluated the effectiveness of *Guided-LORE* by comparing it with LIME and LORE, arguably two of the most popular explanation methods at the moment. We used three publicly available datasets: adult, german and compas. The first two datasets are available in the well-known UCI Machine Learning Repository. The adult dataset includes 48,842 records and 14 attributes. The german dataset is composed by 1,000 individuals, which are classified as "good" or "bad" creditors according to 20 categorical and numerical features. The compas dataset from ProPublica, that contains 1000 instances, was used by the COMPAS algorithm for scoring the risk of defendants (Low, Medium and High). In this final experiment we followed the work in LORE and considered a binary classification version with the two classes "Low-Medium" and "High". In addition to that, we compared our method with LORE on a private dataset for the assessment of risk of developing diabetic retinopathy (DR) for diabetic patients. It is composed of 2,323 examples of binary classification. The Diabetic-Retinopathy data set was used to develop a fuzzy random forest-based system, called RETIPROGRAM, which is currently being used in the Hospital de Sant Joan in Reus (Tarragona). Each instance in the data set is defined by nine attributes: current age, sex, years since diabetes detection, type of diabetes treatment, good or bad control of arterial hypertension, HbA1c level, glomerular filtrate rate estimated by the CKD-EPI value, microalbuminuria, and body mass index. The data was split into a training set of 1,212 examples and a test set of 1,111 examples. The classification model used in RETIPROGRAM achieves an accuracy of 80%, with a sensitivity of 81.3% and specificity of 79.7% [57]

In the case of the diabetic retinopathy risk assessment problem we directly used our fuzzy random forest-based system, i.e., RETIPROGRAM.

Considering the public datasets, we followed the experimental setup described in [52] to make the comparisons fair. We randomly split each dataset into a training set with 80% of the instances, and a test set, i.e. the set of instances for which the black-box decision has to be explained, with 20% of the instances. We used the former to train the black box predictors, whereas the latter was used to evaluate the systems. The black-box predictors used in the test were the following: a Support Vector Machine (SVM) with RBF kernel, a Random Forest classifier (RF) with 100 trees, and a Neural Network (NN) with two layers (the first one has 100 neurons and the last one has one neuron) and the *lbfg* solver. Table 3.2 shows the number of training and testing examples used in the test for each data set.

	Train	Test	Total
Adult	39,074	9768	48,842
Compas	800	200	1,000
German	8,000	2,000	10,000
Diabetic Retinopathy	1,212	1,111	2,323

TABLE 3.2: Datasets employed in the evaluation.

Evaluation Metrics

The following evaluation metrics were used to evaluate the quality of the explanations generated by the proposed method and compare it to the state-of-the-art methods, namely LIME and LORE.

- Hit: this metric computes the similarity between the output of the explanation model and the black-box, b, for all the testing instances. It returns 1 if they are equal and 0 otherwise.
- **Fidelity**: this metric measures to which extent the explanation model can accurately reproduce the black-box predictor for the particular case of instance x. It answers the question of how good is

the explanation model at mimicking the behaviour of the black-box by comparing its predictions and the ones of the black-box on the instances that are neighbours of x, which are in D.

- **l-Fidelity**: it is similar to the *fidelity*; however, it is computed on the subset of instances from D covered by the explanation rules, $\mathcal{R}$. It is used to measure to what extent these rules are good at mimicking the black-box model on similar data of the same class.
- **c-Hit**: this metric compares the predictions of the explanation model and the black-box model on all the counterfactual instances of x, $\mathcal{C}$.
- **cl-fidelity**: it is also similar to the $fidelity$; however, it is computed on the set of instances covered by the the counterfactual rules in a local explanation for x.

Results and Discussion

Table 3.3 shows the results of *Guided-LORE*, LIME and LORE on the three public datasets. Specifically, it reports the mean and standard deviation of the *hit* score of each black-box model. It may be seen that *Guided-LORE* outperforms LORE and LIME. In most of the cases it obtains the best *hit* score and, in those cases where LORE or LIME is better, it shows a very close performance to them. In the case of the Random Forest black-box model, *Guided-LORE* gives the best performance with the three datasets, and the worst one is LIME. Thus, it seems that decision trees can effectively mimic the performance of random forests more than linear regression models. Our method gives the best score for the three black-boxes with the Compas dataset. In the case of the NN black-box model, LIME gives the best performance, and LORE is the worst one. *Guided-LORE* and LIME show a similar performance with the German data set.

Method	BlackBox	Datasets		
		Compas	German	Adult
A. Guided-LORE	SVM	**1.0 ± 0.0**	**1.0 ± 0.0**	**0.96 ± 0.2**
	NN	**1.0 ± 0.0**	0.99 ± 0.01	0.94 ± 0.2
	RF	**1.0 ± 0.0**	0.98 ± 0.1	0.91 ± 0.3
	Average	**1.0 ± 0.0**	**0.99 ± 0.03**	**0.955 ± 0.23**
B. LORE	SVM	**0.99 ± 0.1**	**1.0 ± 0.0**	**0.98 ± 0.1**
	NN	0.98 ± 0.1	0.98 ± 0.1	0.91 ± 0.3
	RF	0.94 ± 0.2	0.92 ± 0.2	0.90 ± 0.3
	Average	0.97±0.1	0.97 ± 0.13	0.93 ± 0.23
C. LIME	SVM	0.82 ± 0.4	0.96 ± 0.1	**0.98 ± 0.1**
	NN	**0.90 ± 0.3**	**1.0 ± 0.0**	**0.98 ± 0.1**
	RF	0.82 ± 0.6	0.88 ± 0.3	0.82 ± 0.4
	Average	0.85 ± 0.4	0.95 ± 0.1	0.93 ± 0.2

TABLE 3.3: The comparison of the three evaluated systems on the hit score.

In Figure 3.4 and Table 3.4, we compare the performance of *Guided-LORE* with LORE, as they share the same extraction explanation process. The values in Table 3.4 report the evaluation results of the application of these two methods on the DR dataset and the FRF black-box model. The reported results reveal that *Guided-LORE* outperforms LORE in all the metrics. Such finding can be ascribed to the fact that as the backbone of the generation function of LORE is a genetic algorithm, it may tend to generate very similar examples to the original instance by making only small changes. In a classical numerical feature space, such small differences may be relevant enough. However, when we deal with fuzzy-based models, the transformation from the classical numerical space to the fuzzy feature space may remove these differences and lead to almost identical instances. In that case, we lose the diversity in the generated neighbours. For example, LORE (or even LIME) may generate an instance that only differs from the original one by changing the age from being 55 to 55.5. Such an example is likely to be identical to the original one in the fuzzy feature space. In our case, as we propose a guided generation process, we avoid this problem by generating examples that are different but close to the original one, in terms of both the classical and fuzzy feature spaces.

	hit	fidelity	l-fidelity	c-hit	cl-fidelity
Guided-LORE	**0.996 ± 0.06**	**0.991 ± 0.04**	**0.989 ± 0.07**	**0.795 ± 0.4**	**0.836 ± 0.32**
LORE	0.963 ± 0.18	0.953 ± 0.03	0.943 ± 0.18	0.765 ± 0.4	0.789 ± 0.31

TABLE 3.4: Guided-LORE vs LORE (DR dataset).

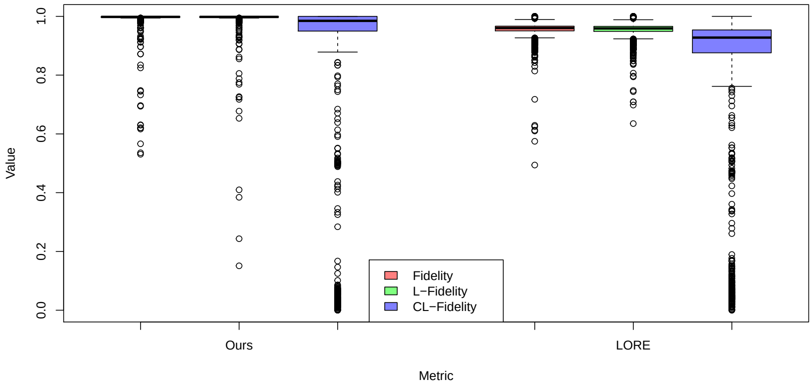


FIGURE 3.4: Comparing the neighbourhood generation method in Guided-LORE and LORE

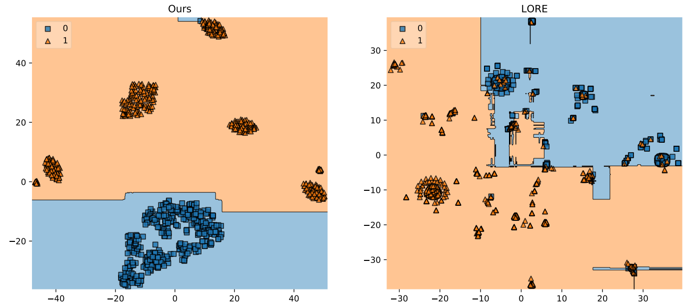


FIGURE 3.5: Decision boundaries of the neighbours generated for an instance using *Guided-LORE* (left) and LORE (right).

Figure 3.4 shows the box-plots of the *fidelity*, *l-fidelity* and *cl-fidelity* measures for *Guided-LORE* and LORE. The former has the highest mean and median values for the three measures, and the lowest variability.

Concerning the outliers, both *Guided-LORE* and LORE show similar performance. Such findings confirm our claim that *Guided-LORE* performs a focused analysis of the neighbourhood of the initial individual x, trying to find the closest "frontier" between the class of x and the other classes and as a result produces a decision boundary that is clear and simple. Figure 3.5 shows a multi-dimensional scaling of the neighbourhood of a sample instance from the DR dataset generated by the two methods. In general, the neighbours generated by *Guided-LORE* are more separable, more compact and denser than the ones generated by LORE. Considering that we are interested in searching the boundary of the predicted class in the state space, we can find that the decision boundary is clear.

3.4 Contextualized LORE for Fuzzy Attributes

The new method we proposed for generating neighbours in Guided-LORE has shown significant improvements compared to the conventional methods of LORE and LIME. By formalising the generation of neighbours as a search problem and utilising Uniform Cost Search, we were able to obtain outstanding results. However, similar to LIME and LORE, our method does not explicitly consider scenarios in which the attributes that define objects are fuzzy. For instance, in Guided-LORE, a neighbouring point is created by increasing or decreasing the value of a numerical attribute by a fixed amount, known as the *step*. This method may be appropriate if the fuzzy sets associated with the linguistic labels are evenly and uniformly distributed across the domain. Nevertheless, this assumption does not hold in many situations.

This section presents our new method *C-LORE-F* (Contextualized LORE for Fuzzy attributes), a variant of Guided-LORE that addresses those issues. Our first motivation is that, if we know that an attribute is fuzzy and we have the information on its fuzzy labels and their associated fuzzy sets, we can profit from that knowledge to make a more focused neighbourhood generation. More precisely, we can generalise its step from being a fixed value to being a function that depends on that knowledge. In that way the proposed method is more general, and it works in the cases in which the fuzzy sets associated with the linguistic labels are uniformly or non-uniformly distributed. To the best of our knowledge, this work is the first one that utilises such knowledge to develop explanation methods for ML systems based on fuzzy logic.

The second novel point of the system is the use of the knowledge about the *type* of attribute to guide the neighbourhood generation process and search for *actionable* explanations. For example, if we have an attribute like age, which automatically increases in time, it probably does not make much sense to look for neighbours that have a lower value in this attribute, as it would not be very interesting for the user to receive an explanation like "if you were 10 years younger, the prediction of the system would be different" (even if this explanation was technically correct). This knowledge about the type of attribute is not currently being used in the most popular explanation systems.

3.4.1 Proposed Method

C-LORE-F uses contextual information (the type of attribute and the fuzzy sets associated to the linguistic values of the fuzzy attributes) to produce explanations mainly for fuzzy-based ML models.

As a first change with respect to LORE and Guided-LORE, we have defined the following types of attributes.

- Attributes with a fixed value (e.g. sex).
- Attributes whose value increases in time (e.g. age).
- Attributes whose value decreases in time (e.g. years left until retirement).
- Variable attributes, that can change positively and negatively (e.g. weight).

Like Guided-LORE, the neighbourhood generation is defined as a search problem in which we explore the neighbourhood space of a point x_0 by applying a Uniform Cost Search based on the Heterogeneous Value Difference Metric (HVDM, [56]), using some contextual information about the attributes. However, the main change in the definition of the generation problem is the introduction of new operators:

Operators: Modifications of the value of a single attribute (feature). These actions leverage some contextual information about the feature to make the desired changes to generate new neighbours. In our case we define two types of actions, *next* and *prev*, described later.

Based on these changes, we have proposed Algorithm 4 which shows how the neighbours of a given instance, x_0, are generated. The search tree starts in x_0, and in each node all the possible actions to move from one instance to another are applied. For each feature $f \in \mathcal{F}$, the number of possible actions can be zero (f is *Fixed*), one (either *next* if the feature is temporally increasing or *prev* if it is temporally decreasing) or both, if f is variable. Each action only changes the value of one feature. The candidate node to be expanded, n, is the one closest to x_0, based on the path cost. If the outcome of the black-box model for n is different from y_0, then it is a leaf of the tree and we do not expand that node further. Otherwise, we expand that node. Consequently, for each node in the second level, we would have changes in two attributes or double changes in the same attribute, and so on. The expanding process finishes when we reach a predefined max-level, or when there are no more nodes to be expanded (all the leaves have led to changes in the initial classification). Repeated instances are ignored to avoid cycles.

The expanding process is done by cloning the instance of the current node, i.e., $n.x$ (lines 13 and 21 in Algorithm 4) and applying the *next* and/or *prev* actions. After that, we pass the obtained instance to the black box model f to get its corresponding label. To apply the actions *next* and *prev* for a given attribute we consider some *separate zones* based

Algorithm 4: C-LORE-F Neighbours Generator

```
Input  : An example $x_0$, its output $y_0$, a black-box model $b$, the
         maximum level of exploration $L$, and the set of
         attributes $\mathcal{F}$.
Output: The set of neighbours, $\hat{D}$.
$root \longleftarrow node\,(x_i, NULL, 0)$, $root.label = y_0 \longleftarrow b\,(x_0)$
$q \longleftarrow [root]$, $\hat{D} \longleftarrow []$
while Not need to stop do
    $n \longleftarrow head\,[q]$
    if $n.label = root.label$ and $n.level \leq L$ then
        foreach attribute $f \in \mathcal{F}$ do
            if $f$ is Fixed then
                // no action to apply
                continue;
            else
                if $f$ can increase then
                    // next action
                    $x_l \longleftarrow copy(n.x)$
                    $x_l[f] \longleftarrow next(x_l, n, n.level + 1)$
                    $n_l.label \longleftarrow b(x_l)$
                    $n_l.d \longleftarrow distance(x_l, x_0)$
                    add $n_l$ to $q$ and $\hat{D}$
                end if
                if $f$ can decrease then
                    // prev action
                    $x_r \longleftarrow copy(n.x)$
                    $x_r[f] \longleftarrow next(x_r, n, n.level + 1)$
                    $n_r.label \longleftarrow b(x_r)$
                    $n_r.d \longleftarrow distance(x_r, x_0)$
                    add $n_r$ to $q$ and $\hat{D}$
                end if
            end if
        end foreach
    end if
end while
```

on its fuzzy sets, which are defined as shown in Figure 3.6, taking into account the intersection point between two consecutive fuzzy sets and the intervals of maximum activation. In Figure 3.6 the zones would be 0-5, 5-10, 10-15, 15-20, 20-25, 25-40, 40-50, 50-60, 60-75, 75-90 and 90-100. Given the value of the attribute, we locate its zone, and then we take the middle of the previous zone as the lower neighbour (the result of the *prev* action), and the middle of the next zone as the upper neighbour (the result of the *next* action). Figure 3.6 shows an example. The input value is 22, which belongs to the zone 20-25. Thus, the middle of the previous zone is the lower neighbour, (15 + 20)/2 = 17.5, and the middle of the next zone is the upper neighbour, (25 + 40)/2 = 32.5. We might end up applying only either the *next* action, if the located zone was the first one, or the *prev* action, if it was the last one.

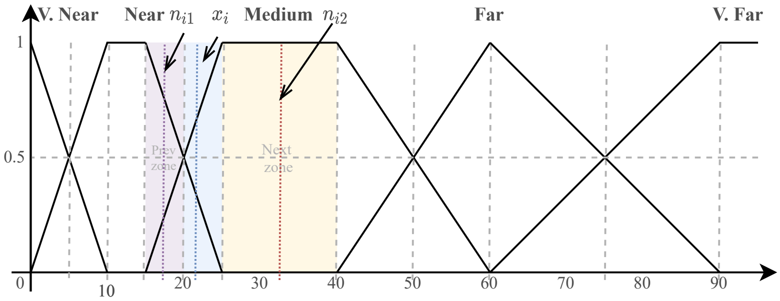


FIGURE 3.6: Illustration of the *next* and *prev* actions.

3.4.2 Experiments and Results

We used the same experimental setup described in 3.3.3 to evaluate the C-LORE-F method. Guided-LORE is referred to as G-LORE.

Table 3.5 shows the means and standard deviations of the metrics for C-LORE-F, LORE and G-LORE on the three data sets with the FRF and FDT models. In general, C-LORE-F outperforms the other methods in all metrics with the FRF model. In the case of FDT, it shows better performance than LORE and G-LORE in *hit* and *fidelity*, and a very good performance on l-fidelity. LORE is the worst in most cases, especially with the FRF model. The reason is that G-LORE and C-LORE-F try to find the closest "frontier" between the class of x_0 and the other classes, producing a clearer decision boundary.

Focusing on the black-box dimensions, all the methods show a better performance with the FRF in the *hit*, *fidelity* and *l-fidelity* metrics. This can lead us to conclude that the accuracy of a model is crucial in getting a better explanation.

At the data sets level, as shown in Figure 3.7, on average, the best performance is obtained by Diabetic-Retinopathy, followed by Adult-Income.

The reason is that all the explanation methods are sensitive to the accuracy of the black-box model. The more accurate is the model, the best is the obtained explanation. In terms of c-hit and cl-fidelity, the best results are obtained with the Diabetic Retinopathy data set. We can attribute this fact to the quality design of the fuzzy sets in this problem. The fuzzy sets of the Diabetic-Retinopathy data set were defined by an expert of the domain, whereas the fuzzy sets of Adult-Income and German-Credit were obtained automatically by applying a fuzzification algorithm [58]. The argument here is that these two metrics rely on the quality of the counterfactual examples that are used to generate counterfactual rules (which may be affected by the generated neighbours). Moreover, the intelligent design of the fuzzy sets is also a key factor in C-LORE-F as it utilises them as contextual information in the neighbourhood generation process. This can be confirmed by comparing the results of C-LORE-F vs others on the Diabetic Retinopathy data set and comparing the performance of *C-LORE-F* method on the Diabetic Retinopathy data set vs the other data sets. C-LORE-F outperforms LORE and G-LORE in almost all evaluation metrics. The cl-fidelity and c-hit are exceptions with the FDT and German-Credit case. In general, all the explanation methods

Model	Method	*hit*	*fidelity*	*l-fidelity*	*cl-fidelity*	*c-hit*
FRF	LORE	0.96±0.19	0.98±0.02	0.97±0.07	0.45±0.43	0.43±0.39
	G-LORE	0.99±0.02	0.99±0.02	0.99±0.03	0.52±0.43	0.47±0.40
	C-LORE-F	**1.00±0.0**	**0.99±0.0**	**0.99±0.0**	**0.59±0.39**	**0.58±0.42**
FDT	LORE	0.95±0.22	0.98±0.03	**0.98±0.03**	0.48±0.41	0.45±0.44
	G-LORE	0.98±0.10	0.98±0.01	0.85±0.24	**0.54±0.45**	**0.50±0.48**
	C-LORE-F	**0.99±0.05**	**0.99±0.0**	0.97±0.08	0.43±0.43	0.41±0.46

TABLE 3.5: The results on the three datasets.

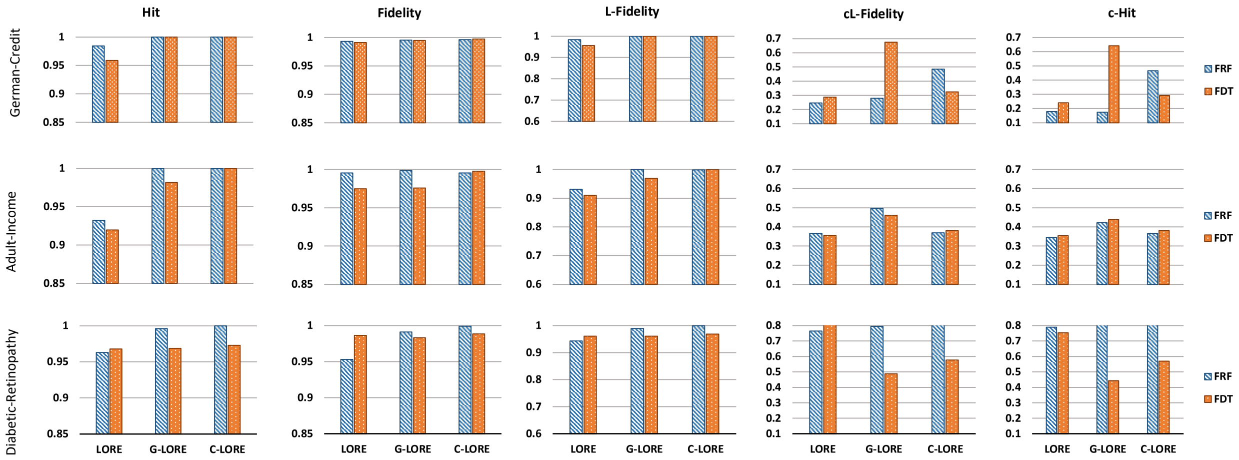


FIGURE 3.7: Comparison results: **LORE** vs **G-LORE** vs **C-LORE-F**.

showed a poor performance in terms of cl-fidelity and c-hit. That may be due to the bad quality of the counterfactual examples, a topic that will be further commented in the next chapters.

3.5 Summary

In this chapter, we have described our two novel methods, which focus on improving the generation of neighbours to yield better explanations of ML systems. The first method, Guided-LORE, is a variant of LORE and aims to clarify the decisions of black-box classifiers. The generation of neighbours of the point being explained is the critical component of our method, which we propose to formulate as a search problem. We use uniform cost search to locate the closest neighbours which result in a change in the predicted class. This method generates compact, dense neighbours with a clear decision boundary. Experimental results show that our method outperforms the state-of-the-art methods LIME and LORE in several metrics.

The second method, C-LORE-F, is a novel technique to explain the decisions of fuzzy-based systems. C-LORE-F leverages the information about the fuzzy sets that define the meaning of the linguistic values of the fuzzy attributes. It also considers the attribute's character, such as whether its value is fixed, increasing, decreasing or variable. One of its primary advantages over similar methods is that the generation process of neighbours for a point x is more informed due to the use of contextual information. We also search for boundaries with meaningful interpretations for the user, for instance, to avoid creating counterfactuals that depend on the change of a fixed attribute or on the positive change of an attribute that only decreases over time. Experimental results on various datasets demonstrate the effectiveness of our proposed method, which outperforms the state-of-the-art methods in several metrics.

Chapter 4

A Comparative Study of Two Rule-based Explanation Methods for Diabetic Retinopathy Risk Assessment

4.1 Introduction

Healthcare costs are continuously raising, due to the increase of life expectancy, the improvements in the management of chronic diseases and the development of new treatments. Diabetes Mellitus (DM), suffered by 382 million adults worldwide, is one of the most important chronic diseases. DM patients are estimated to increase up to 592 million adults by 2035 [59]. Moreover, specialists estimate that around 46% of diabetic patients have not been diagnosed [59]. DM has been growing steadily in the last few years. In Spain, the National Health Surveys (NHS) detected that diabetes increased from 4.1% of the population in 1993 to 6.4% in

2009. Specialists predict an incidence of more than 3 million DM patients in Spain by 2030 [60].

Diabetic Retinopathy (DR) is an ocular disease related to DM. It is the main cause of blindness and visual impairment worldwide and the most common among working-aged adults [61]. Overall, DR affects 30% of diabetic patients, 11% show some degree of vision loss (sight-threatening diabetic retinopathy [62]), and 4% lose their sight completely. However, early detection through periodic screening can reduce this risk by as much as 95%.

AI techniques may improve the screening quality by identifying the patient's risk of developing DR using information from the Electronic Health Record. In the healthcare domain, it is common to build clinical decision support systems (CDSS) using ML tools and algorithms. These intelligent CDSS assist clinicians in diagnosing diseases and choosing treatment decisions.

Recently, we have observed significant and continuous success in the development of ML-based systems in many domains, including healthcare. In line with this progress, our research group and the Ophthalmology Unit of the University Hospital Sant Joan (Reus, Tarragona) have developed a CDSS called RETIPROGRAM ([63],[64]), that helps clinicians to estimate the personalised risk of developing DR as early as possible. The AI core of RETIPROGRAM is a Fuzzy Random Forest (FRF) composed of 100 Fuzzy Decision Trees (FDTs). A FDT is a hierarchical structure that classifies patients based on the values of a set of attributes related to DR risk factors. Each node of the tree represents an attribute. A branch of a node is associated to a possible value of that attribute. Finally, the tree leaves assign patients to two categories: patients with/without risk of developing DR. Each branch is a rule, that provides a result if the attributes have certain values. Experimental results have shown that the system could be incorporated in DR screening programs and improve the quality of screening models [65].

However, as it is well known that, in domains like healthcare, high precision is not enough to convince society to trust the decisions of ML-based systems, we started, as explained in Chapter 3, to develop methods to derive explanations for the predictions of RETIPROGRAM, namely *Guided-LORE* and *C-LORE-F*.

In this chapter, we study two different ways of generating rules in the C-LORE-F method. On one hand, we use the classic crisp decision trees. On the other hand, we propose the construction of preferential decision rules based on rough sets (using the Dominance-Based Rough Set Approach - DRSA, [19]). Both methods are used to generate explanations for the RETIPROGRAM classifier.

The rest of this chapter is structured as follows. Section 4.2 provides an overview of the Dominance-based Rough Set Approach (DRSA) method, one of the rule explanation methods used in this comparative work. Section 4.3 presents a general framework for generating counterfactual-based explanations for the RETIPROGRAM classifier. In Section 4.4, we describe the experimentation, including several metrics to evaluate and compare the performance of both methods. Finally, we conclude the chapter in section 4.5.

4.2 Dominance-based Rough Set Approach

The DRSA method [19] can be used to construct a set of decision rules. The main difference of these rules with respect to those obtained from a decision tree is that they are classification rules with a set of conditions that take into account the preference directions of the input variables. In this section, we explain how the rules are constructed from the set of examples $\mathcal{D}$.

Rough set theory (RST) [66] is a formal theory derived from fundamental research on the logical properties of information systems. The main goal of the rough set analysis is the approximation of concepts. In addition, it offers mathematical tools to discover patterns hidden in data. As a result, it has a wide range of applications, including feature and pattern extraction, data reduction and decision rules generation (our goal in this work). It can also identify partial or total dependencies in data, among other things.

Rough set analysis concerns data stored in a table known as an *Information Table*. Each row represents an object x_i, evaluated with respect to multiple attributes representing different points of view; the information table is defined as a pair $(X, \mathcal{F})$, where X is a non-empty finite set of objects and $\mathcal{F}$ is a non-empty finite set of attributes. A special kind of information table is a Decision Table $(X, \mathcal{F} \cup Dec)$, where the attributes are divided into *condition* attributes $\mathcal{F}$ and *decision* attributes Dec. The former are related to features of objects, while the latter relate to decisions about objects. Often there is just a single decision attribute $\mathcal{Y}$. Distinct values y_k of this attribute, called class labels, induce a partition of the set of objects into so-called decision classes Cl_k.

DRSA is an extension of RST, suitable for analysis of decision tables where both condition attributes from $\mathcal{F}$ and the output decision variable (decision attribute) $\mathcal{Y}$ are ordinal, and there exist monotonic relationships between attributes from $\mathcal{F}$ and $\mathcal{Y}$. A positive relationship means that the greater the value of the condition attribute, the higher the class label. A negative relationship means that the greater the value of the condition attribute, the lower the class label. Both types of relationships are captured by induced decision rules. In general, in DRSA the number of decision classes can be more than two. Then, one has to consider upward and downward unions of decision classes. However, in the case of RETIPROGRAM, we only have two classes: 0 for the absence of DR risk and 1 for the presence of DR risk. Thus, using DRSA, we calculate rough approximations and induce decision rules for exactly these two classes.

While DRSA is primarily designed for ordinal attributes, it can be adapted to handle hybrid attributes of continuous, categorical, and ordinal types , which is the case of RETIPROGRAM. Continuous attributes can be discretized into ordinal categories. This can be done using various methods such as equal-width binning, equal-frequency binning, or more advanced techniques like decision tree-based discretization. Once discretized, these attributes can be treated as ordinal attributes in the DRSA framework. Categorical attributes can be transformed into multiple binary attributes using a technique called one-hot encoding. Each category of the categorical attribute becomes a separate binary attribute, taking the value 1 if the original attribute has that category and 0 otherwise. These binary attributes can then be treated as ordinal attributes in the DRSA framework.

Rules are constructed using elementary building blocks, known as *dominance cones*, with origins in each object in the attribute space. Based on the rough set concept, rules for a lower or/and an upper approximation of each decision class are obtained from a training set ($\mathcal{D}$ in our case) [67]. The choice of DRSA for explainability in the Diabetic Retinopathy disease is motivated by the fact that the values of the attributes are mainly ordinal, and a change from one value to another may be an indicator of the risk of developing DR. Moreover, using the VC-DomLEM algorithm [68], one can induce a set of rules being a minimal cover of consistent objects from both classes. This enables to efficiently distinguish between the two possible decision outputs [68], which is one of the aims of a surrogate model. Two types of rules may be distinguished:

1. $\mathcal{Y} \geq$ decision rules, providing lower profile descriptions for objects belonging at least to class Cl_k (so they belong to Cl_k or a better class, $Cl_{k+1}, Cl_{k+2}, \ldots$):

   IF $f_1 \geq v_1$ AND $f_2 \geq v_2$ AND ... $f_n \geq v_n$ THEN $y \geq y_k$

2. $\mathcal{Y} \leq$ decision rules, providing upper profile descriptions for objects belonging at most to class Cl_k (so they belong to Cl_k or a lower class, $Cl_{k-1}, Cl_{k-2}, \ldots$):

   IF $f_1 \leq v_1$ AND $f_2 \leq v_2$ AND ... $f_n \leq v_n$ THEN $y \leq y_k$.

In this notation, we must take into account that all condition attributes in $\mathcal{F}$ are considered to be maximisation functions (the higher the value, the higher the class label), which are called *Gain* attributes. In case an attribute has to be minimised, it is called a *Cost* attribute, and the lower its value, the higher the class label. It is also possible to introduce a criterion as both Cost and Gain. In this case, the attribute may appear twice in the rule and define an interval of values.

An important feature of the DRSA method coupled with the VC-DomLEM algorithm is the fact that particular rules are minimal (without redundant conditions) and the whole set of rules is non-redundant (if any rule would be removed, some consistent objects would not be covered by any rule).

Algorithm 5 shows the main steps of the DRSA method, assuming that only certain decision rules are considered (as in our case).

Algorithm 5: DRSA method

```
  Input  : $\mathcal{D}$ – training set of objects (decision table)
  Output : $\gamma$ – quality of classification,
           $R$ – set of decision rules generated on $\mathcal{D}$
1 $X^{\geq} \leftarrow CalculateUpwardClassUnions(\mathcal{D})$
2 $X^{\leq} \leftarrow CalculateDownwardClassUnions(\mathcal{D})$
3 foreach $X \in \{X^{\geq}, X^{\leq}\}$ do
4     $X.LowerApproximation \leftarrow$
      $CalculateLowerApproximation(X, \mathcal{D})$
5 end foreach
6 $\gamma = CalculateQualityOfClassification(X^{\geq}, X^{\leq}, \mathcal{D})$
7 $R^{\geq} \leftarrow VC\text{-}DomLEM(X^{\geq})$
8 $R^{\leq} \leftarrow VC\text{-}DomLEM(X^{\leq})$
9 $R \leftarrow R^{\geq} \cup R^{\leq}$
```

In lines 1-2, all upward and downward unions of decision classes are identified, depending on the class labels of $\mathcal{Y}$. In the loop defined in the following lines 3-5, for each upward/downward union its lower approximation is calculated. These approximations are stored inside objects representing particular unions of classes. In line 6, the quality of the classification is calculated. This is a typical rough set descriptor related to consistency of data, defined as a ratio of the number of consistent objects and all objects in $\mathcal{D}$. During calculation of γ, one takes into account the lower approximations calculated previously. In line 7, the VC-DomLEM algorithm is invoked for the upward unions of classes to induce decision rules. It generates rules describing objects from the lower approximations of subsequent unions, iterating from the most specific to the least specific union to control rule minimality. Suppose decision attribute $\mathcal{Y}$ has labels $1, 2, 3, 4, 5$, and the higher the label, the more preferred the respective decision class. Then, VC-DomLEM will first generate rules for class Cl_5, then for upward union of classes $Cl_4^{\geq} = Cl_4 \cup Cl_5$, then for upward union $Cl_3^{\geq}$, and finally for upward union $Cl_2^{\geq}$. Obviously, considering union $Cl_1^{\geq}$ does not make sense (set of all objects). In line 8, the VC-DomLEM algorithm is invoked to induce decision rules for the downward unions of classes. This is realized analogously, with the only difference that this time first class Cl_1 will be taken into account, then downward union of classes $Cl_2^{\leq} = Cl_1 \cup Cl_2$, next downward union $Cl_3^{\leq}$, and finally downward union $Cl_4^{\leq}$. Remark that VC-DomLEM algorithm was introduced for the Variable Consistency DRSA (VC-DRSA), being an extension of the classical DRSA. In [68], there are four input parameters: a set of upward or downward unions of classes, a *rule consistency measure*, a set of *consistency thresholds* for particular unions, and an *object covering option s* (strategy). When invoking the algorithm, we set measure $\widehat{\epsilon}$ [69] for rule consistency measure, supply a set of consistency thresholds all equal to zero (which forces the classical DRSA), and choose 1 for object covering option (indicating that a rule induced for any upward/downward union of classes is allowed to cover only objects from the lower approximation of that union). Moreover, in our problem (binary classification) there is just one upward union $Cl_1^{\geq} = Cl_1$ and one downward union $Cl_0^{\leq} = Cl_0$. Finally, in line 9 the resulting set of decision rules is built by adding sets of rules induced for upward and downward unions of classes.

In the experiments described in this paper, we used the implementations of the DRSA method and the VC-DomLEM algorithm available in the open source *ruleLearn* library[1].

4.3 Explanation Generation System

This section presents the explanation generation system, as shown in Figure 4.1. The input x, a patient record, is first passed to RETIPROGRAM to obtain a class y. Then, the input x and the corresponding output y are passed to the explanation unit (shown in blue at the bottom of Figure 4.1) to extract an explanation for the decision. The explanation unit consists of the neighbours' generation module, the training module, and the explanation extraction module. Finally, the obtained results are forwarded to the evaluation part of the system for performance analysis.

In this chapter, we focus on describing the explanation and evaluation parts in detail. The RETIPROGRAM and its development and evaluation are discussed in previous papers of our research group [57, 70, 71].

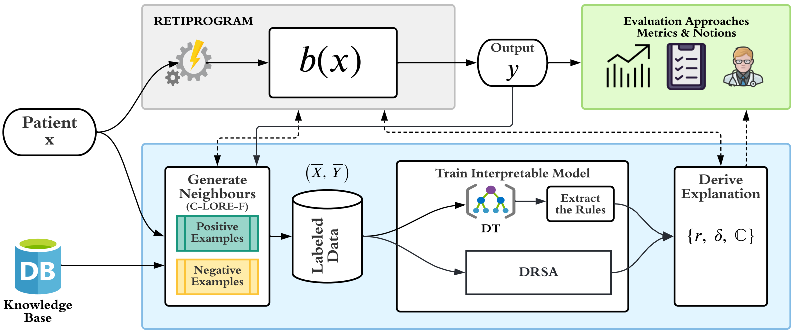


FIGURE 4.1: Architecture of the proposed explanation generation methodology.

The neighbours' generation module is based on the C-LORE-F method, which was presented in Section 3.4. This module applies a Uniform

[1] https://github.com/ruleLearn/rulelearn

Cost Search based on the HVDM distance metric, using contextual information about the features to generate positive and negative neighbours for x. All these examples are labelled using the RETIPROGRAM system, combined into one set D, and fed to the interpretable model training module to build a surrogate interpretable model that mimics the behaviour of the RETIPROGRAM locally on D. Two interpretable models are considered: a Decision Tree and Ordinal Decision Rules.

The explanation extraction module employs the technique described in Section 3.3, where we first extract all decision rules from the interpretable model. In the case of the Ordinal Decision Rules, we use DRSA and obtain a minimal set of rules as the output of the model training. In the case of the Decision Tree, we derive such decision rules just by following the conditions of each branch of the tree. In both cases, the generated models are simpler than the original Fuzzy Random Forest (which has 100 trees, each one with several rules). Once we obtain the set of rules, we can follow the same procedures described in Section 3.3.2 to obtain the explanation output, $\{\mathcal{R}, \Delta, \mathcal{C}\}$.

4.4 Experiments and Results

4.4.1 Experimental Setup

We used the test split of the diabetic retinopathy private data set to evaluate the effectiveness of the proposed explanation system. The description of the data set was shown in subsection 3.3.3 and table 3.2.

4.4.2 Evaluation of the Explanation Results

As we mentioned above, the explanation contains two main parts: first, the explanation decision rule $\mathcal{R}$, and second, a set of counterfactual rules Δ, from which we can derive the counterfactual examples, $\mathcal{C}$. These components are obtained from a set of rules, that we call the explanation model. In this section we want to compare the quality of the rules generated by the two methods. We will denote as C-LORE-F the method using typical decision trees, and we will name as DRSA the version of the same method using rules generated with Dominance-based Rough Sets.

Table 4.1 shows the means and standard deviations of the metrics for the C-LORE-F and DRSA explanation methods on the test set. It may be seen that C-LORE-F outperforms DRSA in all metrics. Let us look at the Fidelity and l-Fidelity for the DRSA method. We can find a difference of 10% in favour of l-Fidelity, which means that most of the disagreements between RETIPROGRAM and DRSA occurred with the examples with a different outcome than the original input. So, the rules describing the opposite classes are worse in DRSA than in C-LORE-F. We can also observe that in both C-LORE-F and DRSA, the cHit and cl-Fidelity show lower performance than the other metrics. This can be attributed to the quality of the generated counterfactual examples (which are evaluated in more depth later in subsection 5.4).

4.4.3 Evaluating the Locality of the Methods

The proposed explanation system is local, because it focuses on the behaviour of RETIPROGRAM around the specific instance x. The Fidelity metrics defined above validate the models' performance in terms of locality with respect to the generated neighbours and the instance to be explained. Assuming that we have access to the test set used to evaluate the black-box model, we can validate the locality of the model with respect to the test set by defining a new metric, the **xt-Fidelity**. It is the *fidelity* measure computed on the set of instances from the test set with a distance to the instance x less than or equal a threshold t. The overall xt-Fidelity on a set X given a threshold t is computed by taking the average of xt-Fidelity for all $x \in X$. We use it to measure the locality vs the globality of the explanation method. It is expected that a local

	Hit	Fidelity	l-Fidelity	cHit	cl-Fidelity
C-LORE-F	**1.00 ± 0.00**	**0.99 ± 0.002**	**0.99 ± 0.002**	**0.89 ± 0.290**	**0.88 ± 0.282**
DRSA	0.97 ± 0.152	0.831 ± 0.32	0.93 ± 0.176	0.830 ± 0.315	0.83 ± 0.298

TABLE 4.1: Evaluation results of the C-LORE-F and DRSA explanation methods.

method shows a degradation in its performance with large thresholds, as a significant number of the selected instances will belong to subspaces different than the one used to build the explanation model.

We compared the xt-Fidelity results of both C-LORE-F and DRSA under different thresholds as illustrated in Figure 4.2. In general, we can find that, as the threshold increases, the overall score decreases, which means we lose the models' locality. In other words, the input space turns out to be more global, and the model fails to cover that space. The degradation in the performance is obvious in the case of DRSA. On the other hand, the C-LORE-F method mostly preserves the performance (the degradation is minor than in DRSA). We can attribute that to the fact that the multiple and small decision trees of C-LORE-F were trained on subspaces of the global space and formed a random forest model. Such a model can show comparable performance on the test set to the performance of RETIPROGRAM (a fuzzy random forest model built from multiple fuzzy decision trees). Hence, if the locality of the explanation model is more important than the globality, we can choose DRSA. Otherwise, C-LORE-F is ideal.

4.4.4 Evaluation of the Counterfactual Examples

Counterfactual examples help to understand what changes are needed to obtain a different outcome. This is particularly interesting in healthcare applications. Hence, it is important to have counterfactual examples that balance a wide range of suggested modifications (diversity) and the relative facility of adopting those modifications (proximity to the actual input). Moreover, counterfactual examples must be actionable, e.g., people can not reduce their age or change their race.

We used the following metrics to evaluate the quality of the counterfactual examples:

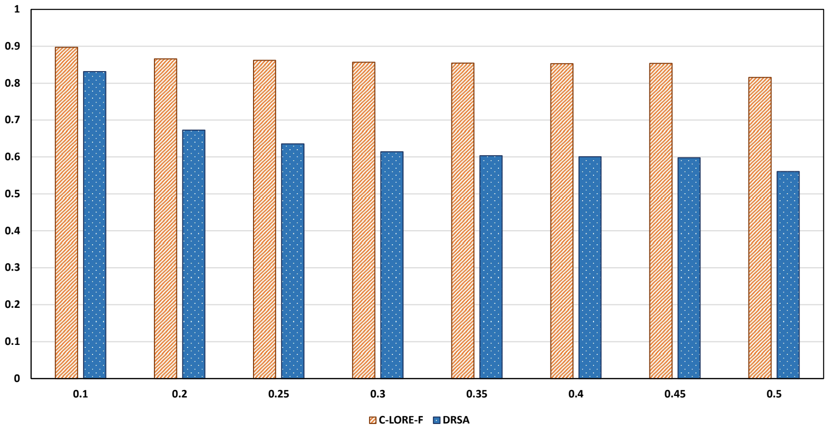


FIGURE 4.2: Degree of locality vs globality of the explanation models. The x-axis represents the thresholds and the y-axis represents the xt-Fidelity.

- **c-Hit**: this metric compares the predictions of the explanation model and the black-box model on all the counterfactual instances of x, $\mathcal{C}$.

- **Validity**: it is the number of counterfactual examples with a different outcome than the original input, i.e., x, divided by the total number of counterfactual examples.

$$\text{Validity} = \frac{|\hat{x} \in \mathcal{C}\ s.t.b(x) \neq b(\hat{x})|}{|\mathcal{C}|} \tag{4.1}$$

- **Proximity**: it is the average feature-wise normalized distance between all counterfactual instances $\mathcal{C}$ and the original input x.

$$\text{Proximity} = 1 - \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} dist(c, x) \tag{4.2}$$

where $dist(c, x)$ is defined as the sum of differences between the corresponding feature values of c and x for each feature f_i:

$$dist(c,x) = 0.5 * \sum_{i=1}^{|\mathcal{F}|} diff(c_{f_i}, x_{f_i}) \tag{4.3}$$

The function $diff(c_{f_i}, x_{f_i})$ returns the difference between the values of c and x at the i-th feature, taking into consideration whether it is categorical or continuous:

$$diff(c_{f_i}, x_{f_i}) = \begin{cases} \frac{1}{N_{cat}} \mathbb{1}[c_{f_i} \neq x_{f_i}] & : \ f_i \text{ is categorical} \\ \frac{1}{N_{cont}} \frac{|c_{f_i} - x_{f_i}|}{MAD_{f_i}} & : \ f_i \text{ is continuous} \end{cases} \tag{4.4}$$

where N_{cat} and N_{cont} denote the number of categorical and continuous features, respectively. The indicator function $\mathbb{1}[.]$ returns 1 if the condition is true, and 0 otherwise. The term MAD_{f_i} represents the median absolute deviation for the i-th continuous feature [72].

- **Sparsity**: it quantifies the average magnitude of changes in attribute values between a counterfactual example and the original input

$$\text{Sparsity} = 1 - \frac{1}{|\mathcal{C}| * |\mathcal{F}|} \sum_{c \in \mathcal{C}} \sum_{f \in \mathcal{F}} \mathbb{1}[c_f \neq x_f] \tag{4.5}$$

- **Diversity**: it is similar to proximity, but it measures the average distance between all pairs of counterfactual instances.

$$\text{Diversity} = \frac{2}{|\mathcal{C}|(|\mathcal{C}| - 1)} \sum_{i=1}^{|\mathcal{C}|} \sum_{j=i+1}^{|\mathcal{C}|} dist(c_i, c_j) \tag{4.6}$$

where *dist* is the distance function defined in equation 4.3.

Results Discussion

Figure 4.3 shows the results of the Validity and Sparsity metrics for C-LORE-F and DRSA. C-LORE-F generates better valid counterfactual examples than DRSA. For both of them, when they generate a single counterfactual example, it is valid (near 100%). Notice that C-LORE-F has never generated more than 5 counterfactual examples. On the contrary, DRSA is able to generate more counterfactuals but the validity decreases and sparsity keeps similar.

Both C-LORE-F and DRSA show outstanding performance on Sparsity (DRSA is slightly better) with an average of 0.9.

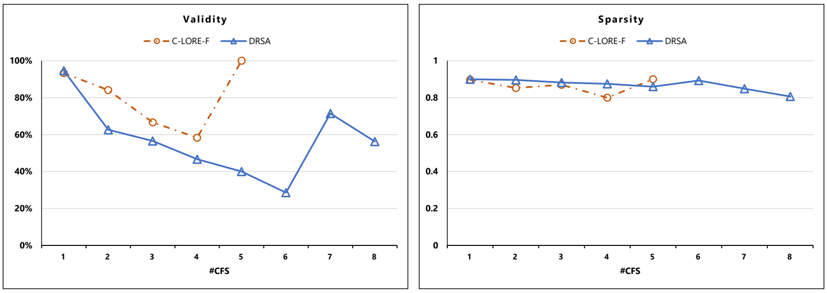


FIGURE 4.3: Validity and Sparsity of counterfactual examples. The numbers in the y-axis represent the metrics values (the higher values the better performance), while the x-axis represents the number of generated counterfactual instances.

Looking at the Diversity results (Figure 4.4), we can find that C-LORE-F generates more diverse examples with respect to the categorical features and diversity increases as the number of counterfactual examples increases. In the case of continuous features, DRSA is slightly better than C-LORE-F.

The Proximity results are shown in Figure 4.5. C-LORE-F generates counterfactual examples with lower proximity than DRSA for both the categorical and continuous features. Moreover, the inherent trade-off

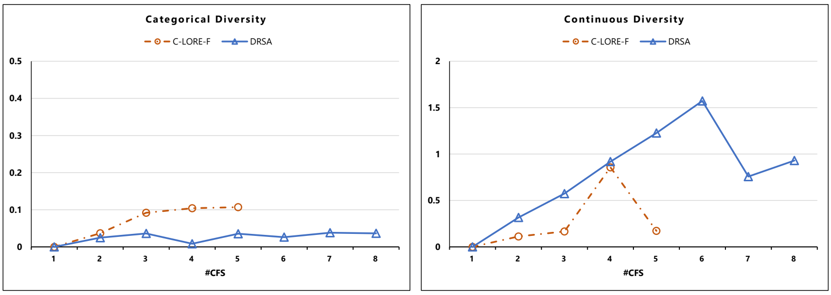


FIGURE 4.4: Diversity of counterfactual examples, in Categorical and Continuous attributes. The numbers in the y-axis represent the metrics values (the higher values the better performance), while the x-axis represents the number of generated counterfactual instances.

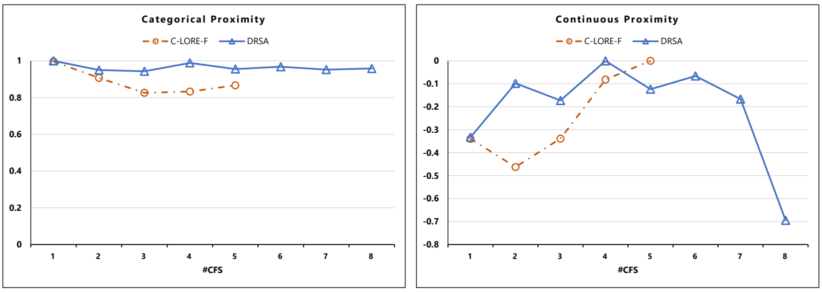


FIGURE 4.5: Proximity of counterfactual examples, in Categorical and Continuous attributes. The numbers in the y-axis represent the metrics values (the higher values, the better performance), while the x-axis represents the number of generated counterfactual instances.

between diversity and proximity metrics can be observed in the case of categorical features.

4.4.5 Analysis of Computational Complexity

Computational resources are crucial in any practical application. In this subsection, we analyse the time complexity of the system. The most costly parts of the system are the generation of neighbours, the construction of the decision tree and the construction of the rules in the DRSA method. So, we first present the theoretical analysis of the time complexity of each part, and then we show the experimental setup and the running time in seconds.

We solve the generation of neighbours as a search problem using the Uniform Cost Search algorithm. Hence, the total running time complexity of this part of the system is $O(b^{1+\lfloor C^*/\epsilon \rfloor})$, where b is the branching factor and C^* is the cost of the optimal solution, assuming that every action costs at least ϵ [1]. The time complexity for the decision tree algorithm is $O(n) + O(m \cdot n \cdot log_2 n) + O(n \cdot log_2 n)$ [73]. The time complexity of the DRSA method is $O(m^2 \cdot n^2)$. Here m is the number of examples, and n is the number of attributes.

The experiments were carried out on a 64- bit computer, with AMD Ryzen 7 3700U (4 Cores, 2.3 GHz) and 16 GB RAM, running Windows 11 operating system. The number of examples in the test set is 1,111. For each example, we generate 800 neighbours and use them to derive the explanation. Table 4.2 shows the elapsed time of generating the neighbours, building the decision trees and constructing the rules in seconds. As expected, the most expensive part of the system is the generation of the neighbours. The decision tree is slightly faster than the DRSA, which is expected as the DRSA complexity is quadratic, whereas the decision tree complexity is logarithmic. The estimated time to obtain an explanation is 9.051 seconds using the C-LORE-F method and 9.268 seconds using the DRSA method.

	Min	Max	Average
Neighbours Generation	5.453	19.252	8.882
C-LORE-F	0.093	1.149	0.169
DRSA	0.125	2.556	0.386

TABLE 4.2: Running time comparisons.

4.5 Summary

We have proposed a methodology to derive an explanation for the decision made by the RETIPROGRAM system, which was developed in our research group to estimate the personalised risk of developing diabetic retinopathy as early as possible. RETIPROGRAM is in use at a regional hospital in the city of Reus (Spain). The work presented in this chapter is focused on comparing two different explanation methods: one based on decision trees (C-LORE-F) and the other one based on decision rules (DRSA). As we have shown in the previous chapter, C-LORE-F with DT [74], was compared with other state-of-the-art methods. The current work shows that DRSA is also a valid method for generating explanatory rules. After comparing the obtained explanation from the C-LORE-F and DRSA methods using multiple evaluation metrics, we found that both generate an adequate explanation. C-LORE-F is slightly better in hit and fidelity indicators, but its sparsity is smaller than that of DRSA. We have also shown that the time needed for the generation of the explanation is of 9 seconds, which is a good time for real use for medical physicians when visiting a patient.

It is worth mentioning that the methods proposed for constructing explanations based on rules are general, even if they have been studied for the Diabetic Retinopathy problem. They could be applied to other fields and with any other black box classifier. Finally, one advantage of using the DRSA method is that, unlike a decision tree, DRSA generates more than just one rule as an explanation. However, the C-LORE-F method with DT is slightly better in most of the evaluation metrics. Hence, in the next chapter, we show that it is possible to overcome the issue of generating a single rule as an explanation output using DT as an interpretable model by replacing it with an FDT.

Chapter 5

Fuzzy-LORE: A Method for Extracting Local and Counterfactual Explanations using Fuzzy Decision Trees

5.1 Introduction

Despite the promising outcome obtained with Guided-LORE and C-LORE-F, they still have some shortcomings. First, the quality of the obtained counterfactual instances should be improved [75]. Second, the basic explanation in LORE (and its variants) is limited to a single rule derived from the activated path in a decision tree, which is not very informative. Third, the method is quite rigid and the explanation can't be adapted to different applications or user types.

In this chapter, we present a novel method called Fuzzy-LORE to address the shortcomings of standard LORE-based methods (i.e., LORE, Guided-LORE and C-LORE-F) and provide better explanations specifically for fuzzy-based ML systems. Fuzzy-LORE adapts our previous LORE-based methods, namely C-LORE-F, by using fuzzy decision trees as an alternative interpretable model to the classical decision trees.

Nowadays, there has been growing interest in using fuzzy logic to develop explanation methods [76]. The linguistic modeling and similarity to human reasoning make fuzzy logic [77] a useful formalism for providing explanations. Alonso *et al.* [78] explore the potential of fuzzy logic in advancing the field of XAI. They discuss how fuzzy logic can be applied to develop explainable models, such as fuzzy rule-based systems and fuzzy clustering, which provide interpretable and transparent representations of the decision-making process. They also highlight important research directions for applying fuzzy logic in XAI, such as developing techniques for combining fuzzy logic with other XAI approaches, exploring new applications of fuzzy logic in XAI, and developing methods for validating and testing fuzzy logic-based models. Mencar and Alonso [79] provide an overview of how fuzzy modeling can be used to facilitate the development of XAI systems. They highlight the advantages of fuzzy logic in effectively representing uncertain and imprecise information, making it a valuable tool for decision-making in complex systems. Mendel and Bonissone [80] argued that rule-based fuzzy inference systems are well-suited for XAI due to their inherent interpretability.

Several methods have been proposed to generate factual and counterfactual rules from fuzzy-based systems. Stepin *et al.* [81] proposed a method for generating factual and counterfactual rules from decision trees. Initially, they identify all the rules that resulted in a label different than the one produced by the system in the point to be explained and considered them as candidate counterfactual rules. Then, they rank these rules based on their distance to the rule with the same label as the instance being explained and the highest confidence. Zhang *et al.* [82] introduced CF-MABLAR, a rule generation framework for Mamdani fuzzy classification systems, which is an extension of the MARkov BLAnket Rules (MABLAR) framework. By approximating the causal links between inputs and outputs of fuzzy systems, CF-MABLAR can generate counterfactual rules based on these approximations. Guillermo *et al.* [83] presented a method similar to the proposed method in this work for generating factual and counterfactual explanations from fuzzy decision trees. They emphasized the benefits of using fuzzy decision trees, which can activate multiple branches during inference and allow for factual explanations incorporating multiple rules rather than just one. Moreover, they suggested a method for generating counterfactual instances based on the given instance and the obtained factual rules. In contrast, our approach involves generating counterfactual rules and using them along with the given instance to generate counterfactual instances, as detailed in section 5.3.

The rest of this chapter is structured as follows. Section 5.2 explains the proposed method. In Section 5.3, we describe the experimental setup and discuss the obtained results. Finally, in section 5.4, we summarize the chapter.

5.2 Proposed Method

As stated earlier, Fuzzy-LORE is an adaptation of our previous method C-LORE-F, in which the explanation for a given example is derived from a Fuzzy Decision Tree rather than a Decision Tree (DT). Hence, in this section we explain how a FDT can be constructed (subsection 5.2.1), and then we outline the inference procedure in FDT in subsection 5.2.2. Finally, in subsection 5.2.3, we describe the explanation extraction procedure.

5.2.1 Fuzzy Decision Tree Construction

The second step in the mcFuzzy-LORE method involves constructing a multi-class fuzzy decision tree (FDT) based on the neighbours of x obtained in the first step. This stage aims to create a concise and understandable model that emulates the behaviour of the black-box classifier and facilitates the derivation of explanations.

We consider that the black-box model b has a set of input attributes, where each attribute f_i is a linguistic variable with terms $\mathcal{T}_i = \{t_{i,1}, t_{i,2}, ..., t_{i,k}\}$, with associated fuzzy sets $\mu_{t_{i,j}}$ that define a strong fuzzy partition. These same terms and fuzzy sets will be used in the generation of the explanations in order to facilitate their interpretation. Figure 5.1 shows as an example the linguistic variable EVOL with its terms and membership functions defined on a numerical reference scale. EVOL is the evolution time of diabetes in the DR data set (the amount of time since the patient was first diagnosed with diabetes).

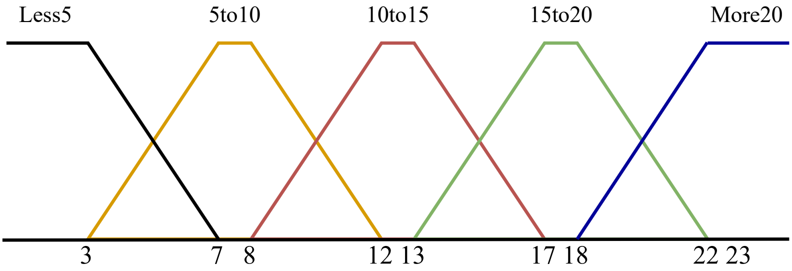


FIGURE 5.1: The membership functions of the EVOL variable.

The algorithm for constructing the FDT in mcFuzzy-LORE is based on the classic ID3 method of induction of trees. In particular, we have considered the fuzzified version proposed by Yuan and Shaw [58] for its simplicity and good performance [71]. The induction procedure has two key parameters: the significance level (α) and the truth level threshold (β).

These parameters guide the tree construction process, with α filtering out evidence that is not significant enough and β regulating the growth of the tree by setting a minimum threshold for ending a branch. Empirical results suggest that $\alpha = 0.1$ and $\beta = 0.9$ are appropriate values for these parameters [18].

The main steps to construct the FDT are the following:

1. Select the best attribute as the root of the tree, based on the ambiguity function [64].

2. For each linguistic term of the selected attribute, create a branch with examples that have support at least equal to α, and compute the truth level of classification for each class.

3. If the truth level of classification is above β for at least one class, the branch is terminated, and the label is set as the class with the highest truth level.

4. Otherwise, check if an additional attribute will further reduce the classification ambiguity. If that is the case, select the best one as a new decision node of the branch and repeat step 2 until no further growth is possible.

5. If no further growth is possible, the branch is terminated as a leaf with a label corresponding to the class with the highest truth level.

Once the tree has been constructed, each branch can be considered as a classification rule, with a degree of support equal to the truth level of its conclusion. These rules have the following structure:

$$r : \text{IF } (f_i \text{ IS } t_{i,a}) \text{ AND } (f_j \text{ IS } t_{j,b}) \text{ ... AND } (f_z \text{ IS } t_{z,c}) \text{ THEN } class \text{ IS } y_k \quad (5.1)$$

5.2.2 Inference in Fuzzy Decision Trees

When a given individual x has to be classified with the FDT, the Mamdani inference procedure is used in the following way:

1. Calculate the activation level of the conditions of each rule using the t-norm minimum, and determine the membership of x to the conclusion class y by multiplying the satisfaction level of the premises and the degree of support, i.e., DoS, of the rule. This value $\mu_r(x, y)$ will also be called the *confidence* of the rule.

$$\mu_r(x, y) = \min(\mu_{t_{i,a}}(x), \mu_{t_{j,b}}(x), ..., \mu_{t_{z,c}}(x)) \cdot \text{DoS}(r) \tag{5.2}$$

2. Combine the memberships for the same class given by different rules using the t-conorm maximum, obtaining the degree of membership of x to each class y. R_y is the set of rules of the FDT with conclusion y.

$$\mu(x, y) = \max_{r \in R_y} \mu_r(x, y) \tag{5.3}$$

3. Select the class with the highest value as the final decision class.

$$y^* = \operatorname*{argmax}_{y \in Y} \mu(x, y) \tag{5.4}$$

5.2.3 Explanation Extraction from FDT

The main change in Fuzzy-LORE is the explanation extraction process. Fuzzy-LORE derives an explanation from the constructed FDT, slightly different from the one derived by the LORE-based methods. Keep in mind that, similarly to LORE [52], Fuzzy-LORE supports only binary classification problems. Let us consider an instance x so that $b(x) = y$, and let $R = R^+ \cup R^-$ be the set of all fuzzy rules of the constructed

FDT given the instance x. Here, R^+ refers to the set of rules that have the conclusion y, and R^- is the set of fuzzy rules that have the *opposite* conclusion. Each rule $r \in R$ has the format defined in section 5.1. Then, we can derive the explanation components $(\mathcal{R}, \Delta, \mathcal{C})$ as follows.

Decision rules

The decision rules $\mathcal{R}$ are the rules used to explain the classification of an instance, x, into a specific label, y. The rules in R^+ provide the foundation for the decision rules of the explanation. Unlike the classical LORE method [52], that only has a single activated crisp rule, Fuzzy-LORE allows for a more flexible explanation by including a subset of R^+ in $\mathcal{R}$. First, we sort the rules in R^+ in descending order according to their confidence scores. Then, depending on the application and user preference, $\mathcal{R}$ can consist of all rules in R^+, the top k rules with the highest confidence scores, or the set of rules with confidence above a certain threshold. In the experiments described in the next section, the top 3 rules with the highest confidence were included in $\mathcal{R}$.

Counterfactual Rules and Instances

This step aims to identify alternative decision rules that are similar to those in $\mathcal{R}$, but lead to different outcomes. The final purpose is to generate counterfactual instances, which are individuals similar to x but that belong to a different class.

Algorithm 6 outlines the process for generating counterfactual explanations, which involves four inputs: x, the instance whose classification has to be explained; $\mathcal{R}$, the set of decision rules; R^-, the set of candidate counterfactual rules; and y, the black-box decision towards x. The algorithm returns a tuple consisting of the set of counterfactual rules, Δ, and the set of counterfactual instances, $\mathcal{C}$.

Algorithm 6: Extraction of counterfactual rules and instances

```
Input  : $x$, the instance; $\mathcal{R}$, the set of decision rules; $R^-$, the
         candidate counterfactual rules; $y$, the black-box
         decision.
Output: $\Delta$, the set of counterfactual rules; $\mathcal{C}$, the set of
         counterfactual instances.
$y_c \leftarrow 1 - y$
$r, v \leftarrow MaxConf(\mathcal{R})$
$\Delta \leftarrow \phi$
$\mathcal{C} \leftarrow \phi$
foreach $r^- \in R^-$ do
    $\eta \leftarrow CondsLowActiv(r^-, x)$
    $x_c \leftarrow Modify(x, \eta)$
    $v_c \leftarrow \mu_{r^-}(x_c, y_c)$
    if $v_c > v$ then
        $\Delta \leftarrow \Delta \bigcup \{r^-\}$
        $\mathcal{C} \leftarrow \mathcal{C} \bigcup \{x_c\}$
    end if
end foreach
return $\Delta, \mathcal{C}$ ;
```

The algorithm first identifies the opposite class and then gets the rule r with the highest confidence score v from the set of rules in $\mathcal{R}$ using the function $MaxConf$. The sets Δ and $\mathcal{C}$ are then initialized as empty sets.

For each rule r^- in the set of candidate counterfactual rules, the function $CondsLowActiv$ returns the set of conditions η in r^- that have an activation of less than 0.5 for x. These are the conditions which could have a higher activation if the values of the corresponding attributes in x were different. This higher activation would then lead to a different classification of x.

Next, the function $Modify$ takes as input x and η to generate a candidate counterfactual instance x_c. To generate x_c, a clone of x is first made. Afterwards, for each condition $c_i = (f_i \text{ IS } t_{i,a})$ in η, the value of f_i in the cloned version of x is set to the center of the term $t_{i,a}$. The center of the term is taken to obtain the maximum activation of the condition.

Finally, the confidence score v_c of the rule r^- is then recalculated using Equation 5.2 for the instance x_c and the label y_c. If $v_c > v$, the rule r^- and the corresponding instance x_c are added to the sets Δ and $\mathcal{C}$, respectively. If this is the case, we have found a counterfactual rule that would be activated with high confidence if some of the values of x were different. These values are precisely those shown in the associated counterfactual instance.

5.3 Experiments and Results

In this section we present and analyse the experimental results obtained by assessing the performance of Fuzzy-LORE alongside other methods and evaluating the generated counterfactual examples. The experiments were conducted using the diabetic retinopathy private dataset.

5.3.1 Evaluation of the Explanation Results

As described in the previous section, a Fuzzy-LORE explanation contains the explanation decision rules $\mathcal{R}$ and a set of counterfactual rules Δ, from which the counterfactual examples, $\mathcal{C}$, are derived. These components are obtained from a fuzzy decision tree (a set of fuzzy decision rules), that we call the *explanation model*. In this section we evaluate the quality of the rules generated by the proposed method and compare it to the LORE-based methods.

Table 5.1 shows the means and standard deviations of the metrics for Fuzzy-LORE and the previous LORE-based methods on the test set. It may be seen that Fuzzy-LORE and C-LORE-F show almost the same performance in the Hit and Fidelity measures. C-LORE-F is slightly better than Fuzzy-LORE in terms of l-Fidelity. However, Fuzzy-LORE outperforms clearly all the other methods in terms of c-Hit. We can attribute such improvement in the c-Hit measure to the quality of the generated counterfactual examples (which are evaluated in more depth in the next subsection 5.3.2).

Methods	Hit	Fidelity	l-Fidelity	c-Hit
LORE	0.95±0.13	0.96±0.05	0.95±0.09	0.79±0.32
Guided-LORE	0.99±0.02	0.98±0.06	0.99±0.03	0.83±0.28
C-LORE-F	**1.00±0.00**	**0.99 ± 0.002**	**0.99 ± 0.002**	0.89 ± 0.29
Fuzzy-LORE	**1.00±0.00**	**0.99±0.03**	0.98±0.04	**0.96 ± 0.17**

TABLE 5.1: Evaluation of the explanation results for Fuzzy-LORE vs other LORE-based methods.

5.3.2 Evaluation of the counterfactual examples

In this subsection, we evaluate the generated counterfactual examples for C-LORE-F and Fuzzy-LORE (as they showed almost the same performance).

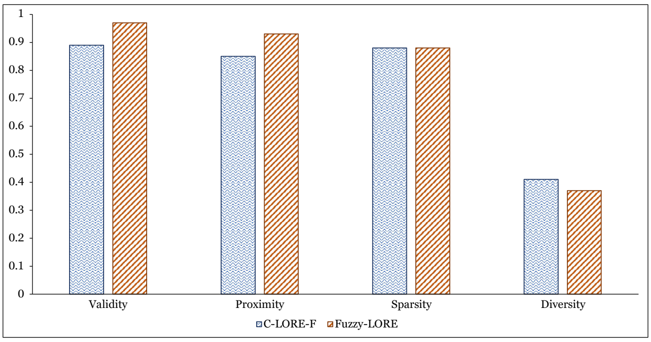


FIGURE 5.2: Evaluation of the counterfactual examples for C-LORE-F and Fuzzy-LORE.

Figure 5.2 shows the comparative results of Fuzzy-LORE vs C-LORE-F with respect to these evaluation metrics. In general, Fuzzy-LORE showed better performance than C-LORE-F, mainly in terms of validity and proximity. Both Fuzzy-LORE and C-LORE-F have similar performance in terms of sparsity. Looking at the diversity results, we can find that C-LORE-F generates slightly more diverse counterfactual examples than the proposed method. However, both of them showed a low performance in this metric. This issue will be studied in future work.

5.4 Summary

Fuzzy-LORE is a novel post-hoc explanation technique for fuzzy binary classifiers, which learns a local fuzzy decision tree on a synthetic neighbourhood of an instance. It then extracts a meaningful explanation consisting of three components: (1) A set of decision rules that explain the reasons behind the classification decision. (2) A set of counterfactual rules that suggest minimal changes to the instance features to obtain a different outcome. (3) A set of counterfactual examples. The method was evaluated on a dataset used to assess the risk of developing diabetic retinopathy. Results showed that using the fuzzy decision tree as an explanation model provides better explanations than the decision tree, particularly in the counterfactual rules and instances. However, both Fuzzy-LORE and other classical LORE-based methods only support binary classifiers, which is not representative of most real-world scenarios. For example, the first version of RETIPROGRAM developed by our research group was a binary classifier, but recently we have developed a new version that supports the classification of diabetic retinopathy into one of five grades using a multi-class fuzzy random forest classifier. In the next chapter, we describe how we adapted the Fuzzy-LORE method and developed a new method for explaining the decisions produced by multi-class fuzzy-based classifiers.

Chapter 6

Multi-class Fuzzy-LORE

6.1 Introduction

Multi-class classification is a fundamental task in Machine Learning, with applications in various domains such as healthcare, finance, and transportation [84]. With the increasing availability of resources and the growing amount of data being produced, successful models and algorithms have been developed for multi-class classification. One example of a fuzzy-based algorithm that can address the multi-class classification problem is the FRF, which is an ensemble of many FDTs [85, 64]. This algorithm combines the advantages of fuzzy logic and random forests to provide a powerful and accurate tool for classification tasks.

However, as discussed in chapter 1, as these models become more accurate, they also become more complex and harder to understand. They can be viewed as black boxes, which can be a major issue when it comes to gaining insight into how the model makes its predictions and building trust in its decision-making process [28, 86].

In the previous chapters, we explained C-LORE-F and Fuzzy-LORE, two LORE-inspired methods for explaining the decisions made by fuzzy-based systems. C-LORE-F incorporates additional information about the fuzzy sets that define the meaning of the linguistic values of the fuzzy attributes and uses this information in the generation of neighbours of the studied instance. Fuzzy-LORE employs fuzzy decision trees as an alternative to classical decision trees to address the limitations of standard LORE-based methods. However, these methods can only explain binary-class classifiers.

In this chapter, we propose a novel method called multi-class Fuzzy-LORE (mcFuzzy-LORE), which extends Fuzzy-LORE to explain multi-class fuzzy-based classifiers such as fuzzy random forests. It can also be applied to explain binary fuzzy-based classifiers as they are a special case of multi-class classifiers.

We evaluated the proposed method on a private dataset used to train an FRF-based multi-class classifier that assesses the risk of developing diabetic retinopathy in diabetic patients. The experimental results show that mcFuzzy-LORE outperforms the prior classical LORE-based methods, especially in generating counterfactual instances.

The rest of this chapter is structured as follows. First, section 6.2 defines and formulates the problem. Then, in Section 6.3, we explain the proposed method. Next, section 6.4 describes the experimental setup and discusses the obtained results. Finally, in Section 6.5, we conclude the chapter.

6.2 Problem Formulation

In this section,we first recall the basic notations used in the classification of tabular data. Subsequently, we outline the problem of explaining the outcomes produced by black box models and introduce the concept of explanation, for which we offer a proposed solution.

Consider a tabular data classification problem where the input data is represented by a set of instances, X, and each instance $x \in X$ has a set of features $x_{f_1}, x_{f_2}, ..., x_{f_{|\mathcal{F}|}}$. Y represents the output labels, and each instance is associated with a label $y \in Y$. The goal of a classifier is to learn a mapping from instances to labels, represented by a function $f : X \rightarrow Y$. In this context, Y can either be nominal or ordinal. Nominal labels are categorical values with no inherent order or ranking. On the other hand, ordinal labels are categorical values with a defined order.

The problem of providing an explanation for a black box multi-class classifier is defined as follows.

A black box classifier, b, is a type of fuzzy machine learning model that utilizes a specific method to fuzzify the input attributes. It takes in a set of input features, x, and produces an output, y, while keeping its internal workings or decision-making process undisclosed. Therefore, we have access to the input features and the classifier's outputs. The goal is to provide an explanation, e, for the decision $b(x) = y$. The explanation e is form of a triplet $(\mathcal{R}, \Delta, \mathcal{C})$ as defined in 3.2.

6.3 Proposed Method

In mcFuzzy-LORE, we propose a modification to Fuzzy-LORE's neighbour generation process, as described in Section 6.3.1. Unlike Fuzzy-LORE, which generates synthetic neighbours from only two classes, the predicted class and its opposite, mcFuzzy-LORE generates a set of synthetic neighbours from a user-defined set of labels of interest ($Y^* \subseteq Y$), where Y is the set of all possible labels. The generation process is based on the approach proposed in C-LORE-F in section 3.3.

Additionally, since Y^* can contain more than one label, we make some adaptations to the procedure for extracting counterfactual rules and instances, as explained in section 6.3.2.

6.3.1 Neighbours Generation

Algorithm 7 presents the procedure for generating neighbours of a given instance. The algorithm takes as input the instance to be explained (x), the black box system (b), the class (y) assigned to x by the black box, the set of labels (or classes) of interest ($Y^* \subseteq Y$), an auxiliary set (T), and the maximum depth of analysis (L).

The selection of the labels of interest, Y^*, for generating the explanation depends on the nature of Y. While it may be determined by the user, if Y is nominal, Y^* may contain all labels in Y. However, if Y is ordinal, Y^* should include y and its previous and next labels. If y is the k-th label in Y, then $Y^* = \{y_{k-1}, y_k, y_{k+1}\}$. This set will only contain two elements if y is the first or last label in Y. The reason for this definition is that in the case of having a set of ordered labels, individuals close to x will likely belong to y or similar classes, not distant ones. Therefore, looking for close neighbours of x that belong to all classes does not make sense. The auxiliary set T generates examples that are classified with labels in $Y^* \backslash \{y\}$. It can be a subset of the original training set for b or another dataset with a similar distribution. The maximum depth parameter d restricts the exploration of the space around x during the generation process.

The output of the algorithm is a set D that contains synthetic neighbours of x, which are classified by b with the classes of interest. The *Generate* function is the core of the generation process, which is based on the C-LORE-F method in algorithm 4 from section 3.2.

Algorithm 7: Neighbours Generation

```
Input  : $x$: the instance whose classification has to be explained,
         $y$: the decision, $b$: the black box classifier, $Y^*$: the set of
   labels of interest, $T$: the auxiliary test, $L$: the maximum level of
   exploration, and the set of attributes $\mathcal{F}$
Output: $D$: the set of generated neighbours of x.
$D \leftarrow \phi$
$D_y \leftarrow Generate(x, y, b, L, \mathcal{F})$
$D \leftarrow D \bigcup D_y$
foreach $\hat{y} \in Y^* \backslash \{y\}$ do
    $\hat{x} \leftarrow GetClosest(T, x, \hat{y})$
    $D_{\hat{y}} \leftarrow Generate(\hat{x}, \hat{y}, b, L, \mathcal{F})$
    $D \leftarrow D \bigcup D_{\hat{y}}$
end foreach
return $D$ ;
```

The algorithm begins by initialising an empty set D to store the neighbours of the input instance x with labels of interest. The *Generate* function is then called to generate a set of neighbours D_y that are assigned to the same label as x by the black box system.

For each label $\hat{y}$ in the set of labels of interest Y^* that is different from y, the algorithm finds the closest example to x in the auxiliary set T that has the label $\hat{y}$. This example is denoted as $\hat{x}$. The *Generate* function is then applied to $\hat{x}$ to generate neighbours that belong to the class $\hat{y}$ as assigned by the black box system. The resulting neighbours are added to the set D. In the end, D contains individuals that are similar to x and belong to all the classes of interest. This synthetic and multi-class dataset D can be used to build a fuzzy decision tree, as explained in section 5.2.1.

6.3.2 Explanation Extraction from a FDT

Given the problem definition, the aim of mcFuzzy-LORE is to generate an explanation e in the form of $(\mathcal{R}, \Delta, \mathcal{C})$. In order to achieve this, the rules obtained through the FDT induction algorithm are divided into two groups: R^+ and R^-. The former group includes rules with the conclusion y, the label assigned to the input x by the black box system, while the latter consists of rules with a label y^- from the set of labels of interest, but different from y, i.e. $y^- \in L^* \backslash \{y\}$. R^+ is referred to as the candidate decision rules, while R^- is referred to as the candidate counterfactual rules. Using this categorization, the explanation components can be derived as follows. As the group of labels only changes for R^-, the extraction procedure for decision rules, $\mathcal{R}$, remains the same as explained in Fuzzy-LORE in section 5.2.3.

Algorithm 8 describes the adapted process for generating counterfactual explanations. For each rule r^- in the set of candidate counterfactual rules, the algorithm determines the label of this rule using the function *Conclusion*, i.e., $y_c \in Y^* \backslash \{y\}$. Then, we follow the same steps explained in section 5.2.3 to decide whether the rule r^- and the corresponding

Algorithm 8: Extraction of counterfactual rules and instances

```
Input  : $x$, the instance; $\mathcal{R}$, the set of decision rules; $R^-$, the
         candidate counterfactual rules.
Output : $\Delta$, the set of counterfactual rules.; $\mathcal{C}$, the set of
         counterfactual instances.
$r, v \leftarrow MaxConf(\mathcal{R})$
$\Delta \leftarrow \phi$
$\mathcal{C} \leftarrow \phi$
foreach $r^- \in R^-$ do
    $y_c \leftarrow Conclusion(r^-)$
    $\eta \leftarrow CondsLowActiv(r^-, x)$
    $x_c \leftarrow Modify(x, \eta)$
    $v_c \leftarrow \mu_{r^-}(x_c, y_c)$
    if $v_c > v$ then
        $\Delta \leftarrow \Delta \bigcup \{r^-\}$
        $\mathcal{C} \leftarrow \mathcal{C} \bigcup \{x_c\}$
    end if
end foreach
return $\Delta, \mathcal{C}$ ;
```

instance x_c can be added to the sets Δ and $\mathcal{C}$, respectively. It is important to highlight that mcFuzzy-LORE adopts a distinct approach in the multi-class setting, even though it employs the same underlying algorithm as the binary setting. The key distinction lies in the consideration and analysis of a specific subset of output labels, enabling the generation of counterfactual rules and instances for each label of interest. This allows mcFuzzy-LORE to provide comprehensive explanations by encompassing the multiple classes present in the problem domain.

6.3.3 Illustrative Example

The procedure for generating decision rules, counterfactual rules, and counterfactual instances can be illustrated with an example involving the diagnosis of diabetic retinopathy, which is the application that will be described in the experimental section. A random instance from the dataset is selected, denoted by $x = (Age : 65, Sex : 0, EVOL : 10, TTM : 1, HbA1c : 10.5, CKDEPI : 85.05, MA : 160.0, BMI : 30.3, HTAR : 0.0)$. This instance belongs to Class1. The membership degrees for each fuzzy variable in the form of $f_i : \{t_{i,a} : \mu_{t_{i,a}}(x)\}$, for all linguistic terms of f_i, are the following:

- **Age**: Twenties: 0, Thirties: 0, Forties: 0, Fifties: 0, Sixties: 1, Seventies: 0, Old: 0
- **Sex**: Man: 1, Woman: 0
- **EVOL**: Less5: 0, 5to10: 0.5, 10to15: 0.5, 15to20: 0, More20: 0
- **TTM**: Diet: 0, OralAntidiab: 1, Insuline: 0
- **HbA1c**: Less6: 0, 6to7: 0, 7to8: 0, 8to9: 0, More9: 1
- **CKDEPI**: VeryLow: 0, Low: 0, Normal: 0, High: 0.9, VeryHigh: 0.1
- **MA**: Correct: 0, High: 1
- **BMI**: Underweight: 0, NormalLow: 0, NormalHigh: 0, Overweight-Low: 0, OverweightHigh: 0.5, ObeseLow: 0.5, ObeseHigh: 0
- **HTAR**: GoodControl: 1, BadControl: 0

The meaning of these attributes is described in the experimental section. Just to follow the example, please note that TTM (treatment) is a discrete attribute, and its values are encoded as 0: Diet, 1: OralAntidiab, and 2: Insuline. Therefore, the condition TTM IS Insuline, for example, would be activated with a degree of membership of 1 if the value of TTM in x were 2.

In this example, the top rule from the set of rules that concludes with "Class1" is selected as the only decision rule, denoted by the set $\mathcal{R}$. The rule is the following:

$$\begin{aligned}&r^{+} : \text{ IF}(\mathit{Evol} \text{ IS } 10\mathit{to}15) \text{ AND } (\mathit{HbA1c} \text{ IS } \mathit{More}9)\\ &\text{AND } (\mathit{TTM} \text{ IS } \mathit{OralAntidiab})\\ &\text{THEN}\\ &\mathit{class} \text{ IS } \mathit{Class}1 \ (\mathit{DoS} = 0.73,\ \mathit{confidence} = 0.36)\end{aligned}$$

Thus, $MaxConf(\mathcal{R})$ will return $r = r^{+}$ and $v = 0.36$.

The following two rules are considered from the set R^{-} of candidate counterfactual rules:

$$\begin{aligned}&r_1^{-} : \text{ IF}(\mathit{Evol} \text{ IS } 5\mathit{to}10) \text{ AND } (\mathit{TTM} \text{ IS } \mathit{Diet})\\ &\text{THEN}\\ &\mathit{class} \text{ IS } \mathit{Class}0 \ (\mathit{DoS} = 1.0,\ \mathit{confidence} = 0.0)\\ &r_2^{-} : \text{ IF}(\mathit{Evol} \text{ IS } \mathit{More}20)\\ &\text{AND } (\mathit{TTM} \text{ IS } \mathit{Insuline})\\ &\text{AND } (\mathit{HTAR} \text{ IS } \mathit{GoodControl})\\ &\text{THEN}\\ &\mathit{class} \text{ IS } \mathit{Class}2 \ (\mathit{DoS} = 52,\ \mathit{confidence} = 0.0)\end{aligned}$$

In the case of r_1^{-}, by invoking the $CondsLowActiv(r_1^{-}, x)$ function, the value of η would be $\{(TTM\ IS\ Diet)\}$. Applying the $Modify$ function to r_1^{-} would result in a candidate counterfactual instance,

$$\begin{gathered}x_c = (Age : 65,\ Sex : 0,\ EVOL : 10, \textbf{TTM: 0},\ HbA1c : 10.5,\\ CKDEPI : 85.05,\ MA : 160.0,\ BMI : 30.3,\ HTAR : 0.0)\end{gathered}$$

Consequently, the new confidence score for the the rule r_1^{-} would be 0.5 instead of 0.0. As it is greater than $v = 0.36$, r_1^{-} and its associated counterfactual instance would be added to the set of counterfactual rules and instances, respectively. Note that the only difference between the counterfactual instance and x is the value of the treatment attribute TTM.

For r_2^{-}, η would take the value

$$\eta = \{(EVOL\ IS\ More20), (TTM\ IS\ Insuline)\}$$

Applying the $Modify$ function to this rule would return

$$x_c = (Age : 65,\ Sex : 0, \textbf{EVOL: 22.5}, \textbf{TTM: 2},\ HbA1c : 10.5,$$
$$CKDEPI : 85.05,\ MA : 160.0,\ BMI : 30.3,\ HTAR : 0.0)$$

where the values in bold represent the modified attributes. The resulting confidence score for r_2^- would now be 0.5. Since this score is higher than the value of v, i.e., 0.36, we would add r_2^- and its corresponding counterfactual instance to their respective sets.

6.4 Experiments and Results

We have considered two separate evaluations. In the first one, explained in subsection 6.4.2, we compare the performance of mc-FuzzyLORE with LORE [52] and C-LORE-F [75]. In the second evaluation, described in subsection 6.4.3, we evaluate the quality of the generated counterfactual instances.

6.4.1 Dataset

We conducted experiments using a new version from the Diabetic Retinopathy dataset. These data were used to construct a multi-class fuzzy random forest classifier for DR identification, with four distinct levels: NoDR, Mild, Moderate, and Severe. Table 6.1 provides information about the dataset. It was split into training (80%) and testing (20%) subsets. In this study we have used the testing subset to perform the experiments.

6.4.2 Evaluation of the Explanation Results

In this subsection, we compare the performance of the proposed method with the LORE and C-LORE-F methods. Since these two methods were

TABLE 6.1: Diabetic retinopathy data distribution.

	Training	**Testing**	**Total**
NoDR	1394 (83.6%)	349 (83.7%)	1743
Mild	191 (11.5%)	48 (11.5%)	239
Moderate	58 (3.5%)	14 (3.4%)	72
Severe	24 (1.4%)	6 (1.4%)	30
Total	1667	417	2084

initially designed for binary classifiers, we adapted them to work with multi-class classifiers using the one-vs-all multi-classification method. This adaptation involved treating each class as the positive class and all the other classes as the negative class. The results of these two methods are the average of the results of the four cases.

Method	Hit	Fidelity	l-Fidelity	c-Hit
LORE	0.84	0.98	0.98	0.34
C-LORE-F	**0.99**	**0.99**	**0.99**	0.27
mcFuzzy-LORE	0.96	0.98	0.98	**0.63**

TABLE 6.2: Evaluation of the explanation results for mcFuzzy-LORE vs other LORE-based methods.

Table 6.2 presents the results of the evaluation of the explanation model, considering the case in which the class labels are nominal. The results indicate that C-LORE-F and mcFuzzy-LORE outperform LORE in terms of the Hit metric. This finding suggests that C-LORE-F is a more advantageous neighbour generation method than LORE. In terms of the Fidelity and l-Fidelity metrics, the performance is very similar, with C-LORE-F showing a slightly better performance. Regarding the c-Hit metric, mcFuzzy-LORE offers more significant counterfactual explanations than LORE and C-LORE-F.

Furthermore, we investigated the performance of mcFuzzy-LORE under two different scenarios, considering that the labels in the DR dataset

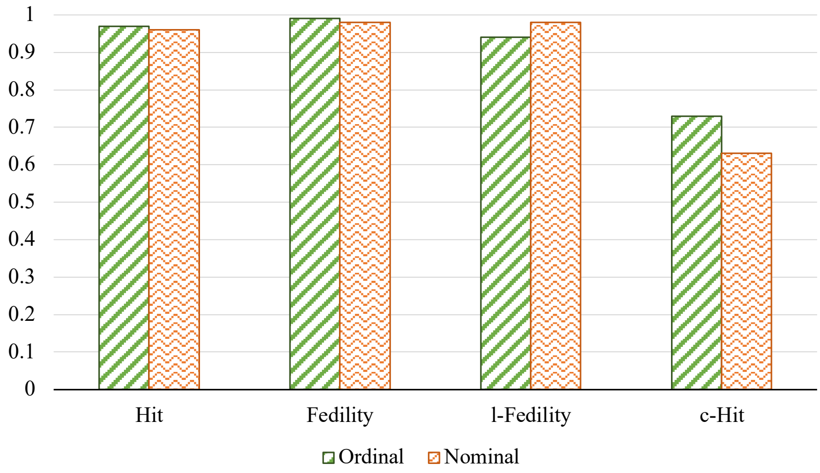


FIGURE 6.1: Evaluation of the explanation results for the Ordinal and Nominal cases.

are nominal or ordinal, as explained in subsection 6.3.1. For the nominal case, we generated neighbours from all labels in Y, while for the ordinal case, we generated neighbours for the classes y_l, y, y_r, where $y_l < y < y_r$. The results of this analysis are presented in Figure 6.1. We did not observe any significant differences in the hit, fidelity, and l-fidelity metrics. However, the c-Hit metric improves in the ordinal case, indicating that the proposed method generated more focused and precise counterfactual explanations. In conclusion, the results suggest that mcFuzzy-LORE performs consistently and robustly under both nominal and ordinal label assumptions.

6.4.3 Evaluation of the counterfactual instances

As we stated earlier, counterfactual instances are important for understanding the modifications required to obtain a different outcome, especially in healthcare applications. Therefore, in this subsection, we assess the effectiveness of mcFuzzy-LORE in the generation of actionable and diverse counterfactual instances in the nominal and ordinal label scenarios.

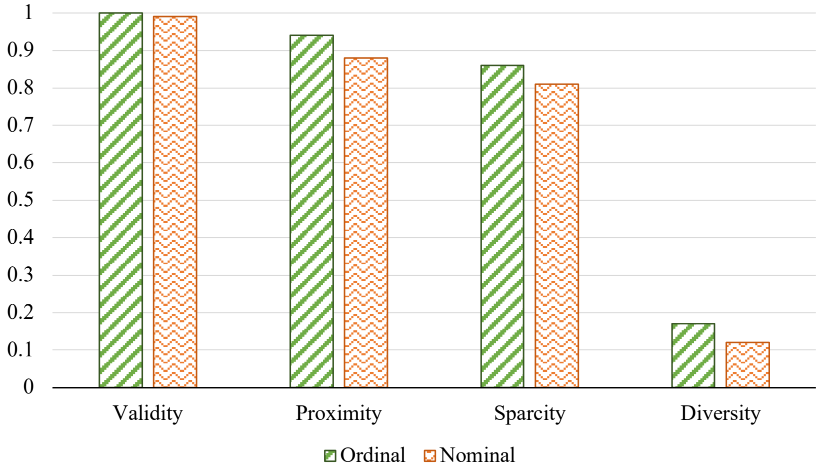
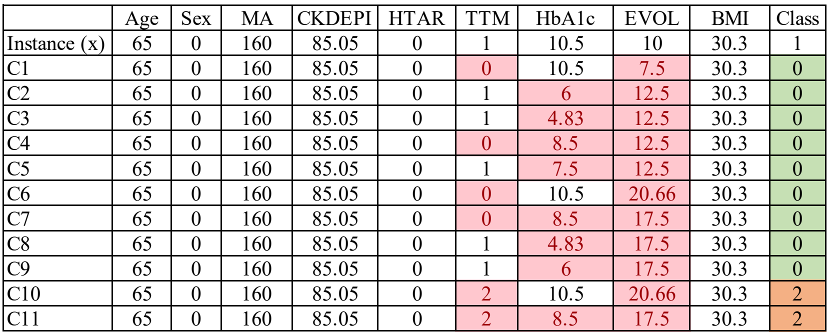


FIGURE 6.2: Evaluation of the counterfactual instances for Ordinal and Nominal cases.

The results displayed in Figure 6.2 reveal that the proposed method consistently constructs valid counterfactual instances regardless of whether the output variable is considered ordinal or nominal. In both cases, nearly all of the generated counterfactual instances have labels different from the one assigned to the input x. In addition, the method's performance is better in the ordinal case than in the nominal case in terms of the Proximity, Sparsity, and Diversity metrics. The obtained results for the Proximity and Sparsity metrics are significant for both nominal and ordinal cases, with scores surpassing 80% for both metrics, which indicates that the counterfactual instances produced by the proposed method show a high level of similarity to the original input (Proximity), and require minimal modifications to attribute values (Sparsity) to reach a counterfactual decision. The observed performance of both metrics suggests that the proposed method effectively generates meaningful and relevant counterfactual explanations that closely resemble the original input while minimizing the number of changes required to achieve a different decision. Overall, these findings highlight the promising potential of the proposed method in generating accurate and close counterfactual instances with minimal sparsity, which could be valuable for explaining and interpreting model predictions. Such findings are confirmed in the example shown in Figure 6.3.

Considering the diversity scores, mcFuzzy-LORE shows a low average diversity score of less than 20%, which can be attributed to the proposed method's goal of making minimal changes to alter the black box's decision, resulting in dense and very similar counterfactual instances.



	Age	Sex	MA	CKDEPI	HTAR	TTM	HbA1c	EVOL	BMI	Class
Instance (x)	65	0	160	85.05	0	1	10.5	10	30.3	1
C1	65	0	160	85.05	0	0	10.5	7.5	30.3	0
C2	65	0	160	85.05	0	1	6	12.5	30.3	0
C3	65	0	160	85.05	0	1	4.83	12.5	30.3	0
C4	65	0	160	85.05	0	0	8.5	12.5	30.3	0
C5	65	0	160	85.05	0	1	7.5	12.5	30.3	0
C6	65	0	160	85.05	0	0	10.5	20.66	30.3	0
C7	65	0	160	85.05	0	0	8.5	17.5	30.3	0
C8	65	0	160	85.05	0	1	4.83	17.5	30.3	0
C9	65	0	160	85.05	0	1	6	17.5	30.3	0
C10	65	0	160	85.05	0	2	10.5	20.66	30.3	2
C11	65	0	160	85.05	0	2	8.5	17.5	30.3	2

FIGURE 6.3: The values of an instance with its generated counterfactual instances.

The table in Figure 6.3 shows as an example an instance (x), in the first row, and its generated counterfactual instances C1 - C11. Highlighted cells refer to the features that have been changed in each counterfactual instance with respect to the original instance. We can see that all counterfactual instances, except for C1, are actionable. In the case of C1, the EVOL value decreased instead of increasing, which contradicts its expected behavior. This issue may arise from selecting close examples

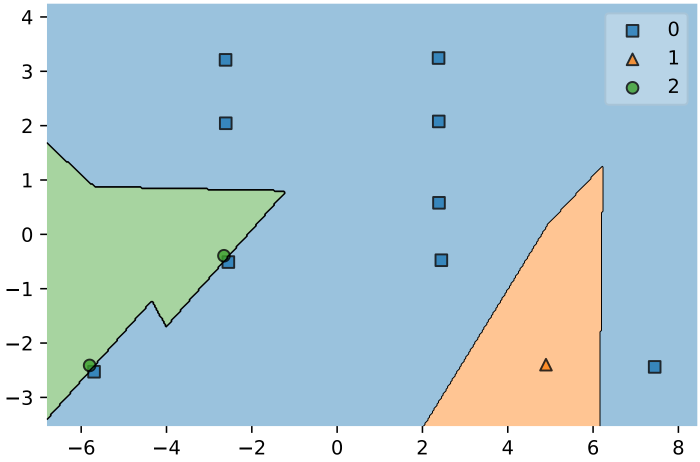


FIGURE 6.4: The scatter plot of the instance with its generated counterfactual instances. Triangle refers to the original instance, squares and circles are the counterfactual instances from two different classes.

that have different labels from the auxiliary set T, which can be resolved by discarding such examples during the generation process. Figure 6.4 shows the scatter plot of the instance (x) and its generated counterfactual instances. We utilized Principal Component Analysis (PCA) and k-Nearest Neighbors (k-NN) to plot a set of counterfactual instances in addition to the instance to explain, in a 2D plane, and draw the class boundaries. Initially, we performed PCA to reduce the dimensionality of the data to 2 dimensions. Next, we plotted the transformed data on a 2D scatterplot, with the first principal component on the x-axis and the second principal component on the y-axis. Subsequently, we employed k-NN as a classifier to segregate the classes and delineate the decision boundaries on the scatterplot. Improving the diversity of the generated counterfactual instances while retaining the main goal of making minimal changes is an area for future work.

6.5 Summary

This chapter presented a novel post-hoc explanation method called multi-class Fuzzy-LORE (mcFuzzy-LORE) to explain the decisions made by multiclass fuzzy-based classifiers such as Fuzzy Random Forests. The distinctive feature of mcFuzzy-LORE is the use of fuzzy decision trees to provide human-readable rules that describe the reasoning behind the classifier output given certain specific inputs. The explanation produced by mcFuzzy-LORE consists of (1) A set of linguistic decision rules that explain the reasons behind the classification decision given by the black box system. (2) A set of linguistic counterfactual rules similar to input instance features but whose output is a different but close class. (3) A set of counterfactual instances that suggest minimal changes in the instance features to get a different but close output class.

Chapter 7

Conclusion

In this chapter, we conclude the Ph.D. dissertation work by summarising the main contributions and suggesting some lines of future work

7.1 Summary of Contributions

Fuzzy-based systems are widely used in many domains due to their ability to handle complex and uncertain systems. However, their opaque nature can be a significant challenge in understanding their decision-making process. This thesis aimed to address this challenge by proposing several methods for generating local and contextualized explanations for fuzzy-based systems.

In Chapter 3, we proposed two novel methods to enhance local surrogate models through neighborhood generation techniques. These methods, known as "Guided LORE" and "Contextualized LORE for Fuzzy Attributes," are specifically designed to generate neighborhoods around specific data points and construct local surrogate models that are more accurate and interpretable for complex systems, such as fuzzy-based models.

In Chapter 4, we studied two different methods for generating rules, namely the C-LORE-F method introduced in Chapter 3 and the Dominance-Based Rough Set Approach (DRSA). Specifically, the study compared classic crisp decision trees and preferential decision rules based on rough sets. The analysis aimed to generate explanations for the results of the RETIPROGRAM classifier, a tool used to assess the risk of developing diabetic retinopathy.

In Chapter 5, we proposed a novel method called Fuzzy-LORE to address the shortcomings of standard LORE-based methods and provide better explanations in the case of fuzzy-based binary ML systems. Fuzzy-LORE adapts our previous LORE-based methods by using fuzzy decision trees as an alternative to the classical decision trees.

In Chapter 6, we proposed a new method called multi-class Fuzzy-LORE (mcFuzzy-LORE) as an adaptation of Fuzzy-LORE to explain the decisions made by multi-class fuzzy-based classifiers such as Fuzzy Random Forests.

Overall, the contributions of this thesis have focused on developing methods for explaining fuzzy-based systems. Our proposed methods for generating local explanations for fuzzy systems are more informative and accurate due to the use of neighborhood generation techniques, contextualized explanations that incorporate the influence of fuzzy attributes, and the use of fuzzy decision trees to generate local and counterfactual explanations. The development of mcFuzzy-LORE provides a novel solution for explaining the decisions made by multiclass fuzzy-based classifiers.

In conclusion, our research has demonstrated that it is possible to improve the interpretability of fuzzy-based systems through the development of methods for generating more accurate and informative explanations. Our methods provide a transparent view of the decision-making process of fuzzy-based models, which can help to build trust in these models and facilitate their adoption in real-world applications.

7.2 Future work

While our proposed methods have shown promising results, there are still several challenges and limitations that need to be addressed in future research.

One promising area for future research is the evaluation of the actions used in the generation of neighbours for Fuzzy-LORE. By defining these actions and utilizing contextual knowledge of the attributes, the Uniform Cost Search algorithm was able to generate relevant neighbours. However, there is still a need to evaluate the effectiveness of these actions and define new actions that can improve the quality of the generated neighbours which will be reflected in the quality of the explanation.

Another area for exploration is the scenario in which there is no access to attribute descriptions. In this case, researchers may need to rely on alternative methods for defining actions in order to generate relevant neighbours.

Another important area for future research is improving the diversity of the counterfactual examples generated by our methods. Although our fuzzy decision tree explanation model outperforms the decision tree in generating counterfactual rules and instances, the diversity scores were relatively low, indicating a need for further research to enhance the diversity of generated counterfactual instances.

Furthermore, additional research is needed to investigate the effectiveness of these counterfactual explanations across different application domains. For example, extending these methods to computer vision or natural language processing tasks could provide valuable insights into the decision-making processes of these models and improve their interpretability.

On the practical side, integrating our proposed methods with RETIPROGRAM, could enhance the accuracy and reliability of the risk assessment by identifying and correcting potential biases or inconsistencies in the underlying FDTs, leading to better patient outcomes and reducing the burden of DR on healthcare systems. Collecting feedback from the doctors at the Sant Joan University Hospital on the real-world use of the Retiprogram system with our proposed methods for explaining the decision allows us to analyze the system's performance and the comments provided by the doctors in cases where the model produces an unreasonable answer. Addressing these practical topics could lead to successful deployment of our methods in real-world systems.

Finally, another potential avenue for future research is the use of language generation models, such as the GPT, to generate human-readable explanations for non-technical users. This approach can provide a more natural language explanation that can be easily understood by the end-users, allowing them to better understand the decisions made by the model. The generated explanations can also serve as a tool to build trust and increase transparency in the decision-making process of the model.

To use this approach, the generated counterfactual instances and explanations can be used as input to the language generation model. The model can then generate a more coherent and human-readable explanation by converting the technical terms used in the counterfactual explanation into simpler language that non-technical users can easily understand. However, this approach also requires further research to evaluate the effectiveness of the generated explanations and ensure that they are still accurate and informative. Moreover, the generated explanations should also be evaluated to ensure that they do not introduce any biases or reinforce any existing biases that may be present in the original model.

Bibliography

[1] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. 2010. URL: http://aima.cs.berkeley.edu.

[2] TM Mitchell. "Machine Learning McGraw-Hill International". In: (1997).

[3] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". In: *nature* 521.7553 (2015), pp. 436–444.

[4] Alejandro Barredo Arrieta et al. In: *Information fusion* 58 (2020), pp. 82–115.

[5] Amina Adadi and Mohammed Berrada. "Peeking inside the black-box: a survey on explainable artificial intelligence (XAI)". In: *IEEE access* 6 (2018), pp. 52138–52160.

[6] Andreas Holzinger et al. "What do we need to build explainable AI systems for the medical domain?" In: *arXiv preprint arXiv:1712.09923* (2017).

[7] Brian S Caffo et al. "Explainable artificial intelligence models and methods in finance and healthcare". In: *Frontiers in Artificial Intelligence* 5 (2022).

[8] Francesco Sovrano and Fabio Vitali. "An objective metric for explainable AI: how and why to estimate the degree of explainability". In: *arXiv preprint arXiv:2109.05327* (2021).

[9] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. "Deep inside convolutional networks: Visualising image classification models and saliency maps". In: *arXiv preprint arXiv:1312.6034* (2013).

[10] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why Should I Trust You?: Explaining the Predictions of Any Classifier". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM. 2016, pp. 1135–1144.

[11] Scott M Lundberg and Su-In Lee. "A unified approach to interpreting model predictions". In: *Advances in neural information processing systems* 30 (2017).

[12] Tim Miller. "Explanation in artificial intelligence: Insights from the social sciences". In: *Artificial intelligence* 267 (2019), pp. 1–38.

[13] Daniel Vale, Ali El-Sharif, and Muhammed Ali. "Explainable artificial intelligence (XAI) post-hoc explainability methods: Risks and limitations in non-discrimination law". In: *AI and Ethics* (2022), pp. 1–12.

[14] Nisha Gupta, Harjeet Singh, and Jimmy Singla. "Fuzzy Logic-based Systems for Medical Diagnosis–A Review". In: *2022 3rd International Conference on Electronics and Sustainable Communication Systems (ICESC)*. IEEE. 2022, pp. 1058–1062.

[15] Yan Wang et al. "A systematic review of fuzzing based on machine learning techniques". In: *PLOS ONE* 15.8 (Aug. 2020), pp. 1–37. DOI: 10.1371/journal.pone.0237749. URL: https://doi.org/10.1371/journal.pone.0237749.

[16] Krzysztof Cpałka. *Design of interpretable fuzzy systems*. Springer, 2017.

[17] Alonso Moral et al. *Explainable fuzzy systems*. Springer, 2021.

[18] Emran Saleh et al. "Learning ensemble classifiers for diabetic retinopathy assessment". In: *Artificial intelligence in medicine* 85 (2018), pp. 50–63.

[19] Salvatore Greco, Benedetto Matarazzo, and Roman Slowinski. "Rough sets theory for multicriteria decision analysis". In: *European journal of operational research* 129.1 (2001), pp. 1–47.

[20] Najlaa Maaroof et al. "Fuzzy-LORE: A Method for Extracting Local and Counterfactual Explanations Using Fuzzy Decision Trees". In: *Artificial Intelligence Research and Development - Proceedings of CCIA 2022*. Ed. by Atia Cortés, Francisco Grimaldo, and Tommaso Flaminio. Vol. 356. Frontiers in Artificial Intelligence and Applications. IOS Press, 2022, pp. 345–354. DOI: 10.3233/FAIA220357. URL: https://doi.org/10.3233/FAIA220357.

[21] Michael Van Lent, William Fisher, and Michael Mancuso. "An explainable artificial intelligence system for small-unit tactical behavior". In: *Proceedings of the national conference on artificial intelligence*. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999. 2004, pp. 900–907.

[22] Gesina Schwalbe and Bettina Finzel. "A comprehensive taxonomy for explainable artificial intelligence: a systematic survey of surveys on methods and concepts". In: *Data Mining and Knowledge Discovery* (2023), pp. 1–59.

[23] Johanna D Moore and William R Swartout. *Explanation in expert systemss: A survey*. Tech. rep. University of Southern California Marina del Rey Information Sciences Inst, 1988.

[24] David Gunning. *Broad agency announcement explainable artificial intelligence (XAI)*. Tech. rep. Technical report, 2016.

[25] David Gunning. "Explainable artificial intelligence (xai)". In: *Defense advanced research projects agency (DARPA), nd Web* 2.2 (2017), p. 1.

[26] Tania Lombrozo. "The structure and function of explanations". In: *Trends in Cognitive Sciences* 10.10 (2006), pp. 464–470. ISSN: 1364-6613. DOI: https://doi.org/10.1016/j.tics.2006.08.004.

[27] S Barocas et al. "The FAT-ML workshop series on fairness, accountability, and transparency in machine learning". In: *cit. on* (2018), p. 7.

[28] Finale Doshi-Velez and Been Kim. “Towards a rigorous science of interpretable machine learning”. In: *arXiv preprint arXiv:1702.08608* (2017).

[29] Cynthia Rudin. “Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead”. In: *Nature machine intelligence* 1.5 (2019), pp. 206–215.

[30] Bryce Goodman and Seth Flaxman. “European Union regulations on algorithmic decision-making and a “right to explanation””. In: *AI magazine* 38.3 (2017), pp. 50–57.

[31] Wojciech Samek and Klaus-Robert Müller. “Towards explainable artificial intelligence”. In: *Explainable AI: interpreting, explaining and visualizing deep learning* (2019), pp. 5–22.

[32] Roberto Confalonieri et al. “A historical perspective of explainable Artificial Intelligence”. In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 11.1 (2021), e1391.

[33] Wojciech Samek, Thomas Wiegand, and Klaus-Robert Müller. “Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models”. In: *arXiv preprint arXiv:1708.08296* (2017).

[34] Leila Arras et al. “" What is relevant in a text document?": An interpretable machine learning approach”. In: *PloS one* 12.8 (2017), e0181142.

[35] Christoph Molnar. *Interpretable machine learning*. Lulu. com, 2020.

[36] Riccardo Guidotti et al. “A survey of methods for explaining black box models”. In: *ACM computing surveys (CSUR)* 51.5 (2018), pp. 1–42.

[37] Nadia Burkart and Marco F Huber. “A survey on the explainability of supervised machine learning”. In: *Journal of Artificial Intelligence Research* 70 (2021), pp. 245–317.

[38] C Lipton Zachary. “The mythos of model interpretability”. In: *Queue* 16.3 (2018), pp. 31–57.

[39] Christoph Molnar, Giuseppe Casalicchio, and Bernd Bischl. “Interpretable machine learning–a brief history, state-of-the-art and challenges”. In: *ECML PKDD 2020 Workshops: Workshops of the European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD 2020): SoGood 2020, PDFL 2020, MLCS 2020, NFMCP 2020, DINA 2020, EDML 2020, XKDD 2020 and INRA 2020, Ghent, Belgium, September 14–18, 2020, Proceedings*. Springer. 2021, pp. 417–431.

[40] Luca Longo et al. “Explainable artificial intelligence: Concepts, applications, research challenges and visions”. In: *Machine Learning and Knowledge Extraction: 4th IFIP TC 5, TC 12, WG 8.4, WG 8.9, WG 12.9 International Cross-Domain Conference, CD-MAKE 2020, Dublin, Ireland, August 25–28, 2020, Proceedings*. Springer. 2020, pp. 1–16.

[41] Seyedeh Neelufar Payrovnaziri et al. “Explainable artificial intelligence models using real-world electronic health record data: a systematic scoping review”. In: *Journal of the American Medical Informatics Association* 27.7 (2020), pp. 1173–1185.

[42] Gabrielle Ras et al. “Explainable deep learning: A field guide for the uninitiated”. In: *Journal of Artificial Intelligence Research* 73 (2022), pp. 329–397.

[43] Diogo V Carvalho, Eduardo M Pereira, and Jaime S Cardoso. “Machine learning interpretability: A survey on methods and metrics”. In: *Electronics* 8.8 (2019), p. 832.

[44] Waddah Saeed and Christian Omlin. “Explainable ai (xai): A systematic meta-survey of current challenges and future opportunities”. In: *Knowledge-Based Systems* (2023), p. 110273.

[45] Weiping Ding et al. “Explainability of artificial intelligence methods, applications and challenges: A comprehensive survey”. In: *Information Sciences* (2022).

[46] Leilani H Gilpin et al. “Explaining explanations: An overview of interpretability of machine learning”. In: *2018 IEEE 5th International Conference on data science and advanced analytics (DSAA)*. IEEE. 2018, pp. 80–89.

[47] Wojciech Samek et al. *Explainable AI: interpreting, explaining and visualizing deep learning*. Vol. 11700. Springer Nature, 2019.

[48] Benjamin Letham et al. “Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model”. In: *The Annals of Applied Statistics* 9.3 (2015), pp. 1350–1371.

[49] Fulton Wang and Cynthia Rudin. “Falling rule lists”. In: *Artificial Intelligence and Statistics*. 2015, pp. 1013–1022.

[50] Rich Caruana et al. “Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission”. In: *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. 2015, pp. 1721–1730.

[51] Berk Ustun and Cynthia Rudin. “Supersparse linear integer models for optimized medical scoring systems”. In: *Machine Learning* 102.3 (2016), pp. 349–391.

[52] Riccardo Guidotti et al. “Local rule-based explanations of black box decision systems”. In: *arXiv preprint arXiv:1805.10820* (2018).

[53] David Alvarez-Melis and Tommi S Jaakkola. “On the robustness of interpretability methods”. In: *arXiv preprint arXiv:1806.08049* (2018).

[54] Giorgio Visani, Enrico Bagli, and Federico Chesani. “OptiLIME: Optimized LIME Explanations for Diagnostic Computer Algorithms”. In: *Proceedings of the CIKM 2020 Workshops co-located with 29th ACM International Conference on Information and Knowledge Management*

(CIKM 2020), Galway, Ireland, October 19-23, 2020. Ed. by Stefan Conrad and Ilaria Tiddi. Vol. 2699. CEUR Workshop Proceedings. CEUR-WS.org, 2020.

[55] Giorgio Visani et al. "Statistical stability indices for LIME: Obtaining reliable explanations for machine learning models". In: *J. Oper. Res. Soc.* 73.1 (2022), pp. 91–101. DOI: 10.1080/01605682.2020.1865846.

[56] D Randall Wilson and Tony R Martinez. "Improved heterogeneous distance functions". In: *Journal of artificial intelligence research* 6 (1997), pp. 1–34.

[57] Emran Saleh, Najlaa Maaroof, and Mohammed Jabreel. "The deployment of a decision support system for the diagnosis of Diabetic Retinopathy into a Catalan medical center". In: *6th URV Doctoral Workshop in Computer Science and Mathematics*. 2020, p. 45.

[58] Yufei Yuan and Michael J Shaw. "Induction of fuzzy decision trees". In: *Fuzzy Sets and systems* 69.2 (1995), pp. 125–139.

[59] Florencia Aguiree et al. "IDF diabetes atlas". In: (2013).

[60] Jonathan E Shaw, Richard A Sicree, and Paul Z Zimmet. "Global estimates of the prevalence of diabetes for 2010 and 2030". In: *Diabetes research and clinical practice* 87.1 (), pp. 4–14.

[61] Arun T Nair, K Muthuvel, and KS Haritha. "Effectual Evaluation on Diabetic Retinopathy". In: *Information and Communication Technology for Competitive Strategies (ICTCS 2020)*. Springer, 2022, pp. 559–567.

[62] Maribel López et al. "Prevalence of diabetic retinopathy and its relationship with glomerular filtration rate and other risk factors in patients with type 2 diabetes mellitus in Spain. DM2 HOPE study". In: *Journal of clinical & translational endocrinology* 9 (2017), pp. 61–65.

[63] Pedro Romero-Aroca et al. "A clinical decision support system for diabetic retinopathy screening: creating a clinical support application". In: *Telemedicine and e-Health* 25.1 (2019), pp. 31–40.

[64] Emran Saleh et al. "Learning fuzzy measures for aggregation in fuzzy rule-based models". In: *International Conference on Modeling Decisions for Artificial Intelligence*. Springer. 2018, pp. 114–127.

[65] Pedro Romero-Aroca et al. "Validation of a diagnostic support system for diabetic retinopathy based on clinical parameters". In: *Translational Vision Science & Technology* 10.3 (2021), pp. 17–17.

[66] Zdzisław Pawlak. *Rough sets: Theoretical aspects of reasoning about data*. Vol. 9. Springer Science & Business Media, 1991.

[67] Roman Słowiński, Salvatore Greco, and Benedetto Matarazzo. "Rough set methodology for decision aiding". In: *Springer handbook of computational intelligence*. Springer, 2015, pp. 349–370.

[68] Jerzy Błaszczyński, Roman Słowiński, and Marcin Szeląg. "Sequential covering rule induction algorithm for variable consistency rough set approaches". In: *Information Sciences* 181.5 (2011), pp. 987–1002.

[69] Jerzy Błaszczyński, Roman Słowiński, and Marcin Szeląg. "Induction of Ordinal Classification Rules from Incomplete Data". In: *Rough Sets and Current Trends in Computing 2012*. Ed. by J.T. Yao et al. Vol. 7413. LNAI. Springer, 2012, pp. 56–65.

[70] Maria Esther Santos Blanco et al. "A Clinical Decision Support System (CDSS) for diabetic retinopathy screening. Creating a clinical support application." In: *Investigative Ophthalmology & Visual Science* 61.7 (2020), pp. 3308–3308.

[71] Emran Saleh et al. "A Fuzzy Random Forest Approach for the Detection of Diabetic Retinopathy on Electronic Health Record Data." In: *CCIA*. 2016, pp. 169–174.

[72] Sandra Wachter, Brent Mittelstadt, and Chris Russell. “Counterfactual explanations without opening the black box: Automated decisions and the GDPR”. In: *Harv. JL & Tech.* 31 (2017), p. 841.

[73] Habiba Muhammad Sani, Ci Lei, and Daniel Neagu. “Computational complexity analysis of decision tree algorithms”. In: *International Conference on Innovative Techniques and Applications of Artificial Intelligence*. Springer. 2018, pp. 191–197.

[74] Najlaa Maaroof et al. “Contextualized LORE for Fuzzy Attributes”. In: *Artificial Intelligence Research and Development* (2021), p. 435.

[75] Najlaa Maaroof et al. “Guided-LORE: Improving LORE with a Focused Search of Neighbours”. In: *Trustworthy AI - Integrating Learning, Optimization and Reasoning*. Ed. by Fredrik Heintz, Michela Milano, and Barry O’Sullivan. Springer, 2021, pp. 49–62. ISBN: 978-3-030-73959-1.

[76] Jose Maria Alonso Moral et al. “Toward Explainable Artificial Intelligence Through Fuzzy Systems”. In: *Explainable Fuzzy Systems: Paving the Way from Interpretable Fuzzy Systems to Explainable AI Systems*. Cham: Springer International Publishing, 2021, pp. 1–23. ISBN: 978-3-030-71098-9. DOI: 10.1007/978-3-030-71098-9_1. URL: https://doi.org/10.1007/978-3-030-71098-9_1.

[77] L.A. Zadeh. “Fuzzy sets”. In: *Information and Control* 8.3 (1965), pp. 338–353. ISSN: 0019-9958. DOI: https://doi.org/10.1016/S0019-9958(65)90241-X. URL: https://www.sciencedirect.com/science/article/pii/S001999586590241X.

[78] Jose Maria Alonso Moral et al. “Designing interpretable fuzzy systems”. In: *Explainable Fuzzy Systems: Paving the Way from Interpretable Fuzzy Systems to Explainable AI Systems* (2021), pp. 119–168.

[79] Corrado Mencar and José M Alonso. "Paving the way to explainable artificial intelligence with fuzzy modeling: tutorial". In: *Fuzzy Logic and Applications: 12th International Workshop, WILF 2018, Genoa, Italy, September 6–7, 2018, Revised Selected Papers*. Springer. 2019, pp. 215–227.

[80] Jerry M Mendel and Piero P Bonissone. "Critical thinking about explainable AI (XAI) for rule-based fuzzy systems". In: *IEEE Transactions on Fuzzy Systems* 29.12 (2021), pp. 3579–3593.

[81] Ilia Stepin et al. "Generation and evaluation of factual and counterfactual explanations for decision trees and fuzzy rule-based classifiers". In: *2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. 2020, pp. 1–8. DOI: 10.1109/FUZZ48607.2020.9177629.

[82] Te Zhang, Christian Wagner, and Jonathan. M. Garibaldi. "Counterfactual rule generation for fuzzy rule-based classification systems". In: *2022 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. 2022, pp. 1–8. DOI: 10.1109/FUZZ-IEEE55066.2022.9882705.

[83] Guillermo Fernández et al. "Factual and Counterfactual Explanations in Fuzzy Classification Trees". In: *IEEE Transactions on Fuzzy Systems* 30.12 (2022), pp. 5484–5495. DOI: 10.1109/TFUZZ.2022.3179582.

[84] Ethem Alpaydin. *Introduction to machine learning 2nd ed*. 2010.

[85] Jordi Pascual-Fontanilles et al. "Adapting a Fuzzy Random Forest for Ordinal Multi-Class Classification". In: *Artificial Intelligence Research and Development*. Vol. 356. IOS Press. Oct. 2022, pp. 181–190. ISBN: 9781643683263. DOI: 10.3233/FAIA220336.

[86] Scott M Lundberg and Su-In Lee. "A Unified Approach to Interpreting Model Predictions". In: *Advances in Neural Information Processing Systems 30*. Ed. by I. Guyon et al. Curran Associates, Inc., 2017,

pp. 4765–4774. URL: http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf.