UNIVERSIDADE FEDERAL DE MINAS GERAIS ESCOLA DE ENGENHARIA
Programa de Pós Graduação em Engenharia Elétrica

Turíbio Tanus Salis

EXPLAINING CLASSIFICATION OUTCOMES THROUGH GEOMETRIC GENERATION OF COUNTERFACTUAL EXAMPLES WITH DATA STRUCTURES AND GRAPHS

Belo Horizonte
2024

Turíbio Tanus Salis

Explaining Classification Outcomes through Geometric Generation of Counterfactual Examples with Data Structures and Graphs

Doctoral thesis submitted to the Examination Committee appointed by the Graduate Program in Electrical Engineering at the School of Engineering of the Federal University of Minas Gerais, as one of the requirements for obtaining the title of Doctor in Electrical Engineering.

Concentration area: Computer and Telecommunications Systems.

Research line: Computational Intelligence.

Advisor: Prof. Antônio de Pádua Braga, PhD.

Belo Horizonte
2024

S166e

Salis, Turíbio Tanus.
Explaining classification outcomes through geometric generation of counterfactual examples with data structures and graphs [recurso eletrônico] / Turíbio Tanus Salis. – 2024.
1 recurso online (172 f. : il., color.) : pdf.

Orientador: Antônio de Pádua Braga.

Tese (doutorado) – Universidade Federal de Minas Gerais, Escola de Engenharia.

Apêndices: f. 149-172.

Bibliografia: f. 124-148.

1. Engenharia elétrica – Teses. 2. Geometria – Teses. 3. Aprendizado de máquina – Teses. 4. Geometria computacional – Teses. 5. Simulação por computador – Teses. 6. Grafo (Sistema de Computador) – Teses. I. Braga, Antônio de Paula. II. Universidade Federal de Minas Gerais. Escola de Engenharia. III. Título.

CDU: 621.3(043)

Ficha catalográfica elaborada pelo bibliotecário Marcio A. A. Gomes CRB/6 2812
Biblioteca Prof. Mário Werneck, Escola de Engenharia da UFMG

UNIVERSIDADE FEDERAL DE MINAS GERAIS

ESCOLA DE ENGENHARIA

COLEGIADO DO CURSO DE GRADUAÇÃO / PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

FOLHA DE APROVAÇÃO

"Explaining Classification Outcomes Through Geometric Generation Of Counterfactual Examples With Data Structures And Graphs"

TURÍBIO TANUS SALIS

Tese de Doutorado submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, como requisito para obtenção do grau de Doutor em Engenharia Elétrica.

Aprovada em 04 de dezembro de 2024.

Por:

Prof. Dr. Antônio de Pádua Braga
DELT (UFMG) - Orientador

Prof. Dr. Frederico Gualberto Ferreira Coelho
DELT (UFMG)

Prof. Dr. Luis Enrique Zárate Gálvez
DECOM (Pontifícia Universidade Católica de Minas Gerais (PUC-MG))

Profa. Dra. Cristiane Neri Nobre
DECOM (Pontifícia Universidade Católica de Minas Gerais (PUC-MG))

Prof. Dr. Luiz Carlos Bambirra Torres
Departamento de Computação e Sistemas (UFOP)

Documento assinado eletronicamente por **Antonio de Padua Braga**, **Membro**, em 04/12/2024, às 17:56, conforme horário oficial de Brasília, com fundamento no art. 5º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Frederico Gualberto Ferreira Coelho**, **Membro**, em 05/12/2024, às 09:57, conforme horário oficial de Brasília, com fundamento no art. 5º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Luis Enrique Zarate Galvez**, **Usuário Externo**, em 03/01/2025, às 13:49, conforme horário oficial de Brasília, com fundamento no art. 5º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Luiz Carlos Bambirra Torres**, **Usuário Externo**, em 09/01/2025, às 17:33, conforme horário oficial de Brasília, com fundamento no art. 5º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Cristiane Neri Nobre**, **Usuário Externo**, em 30/01/2025, às 17:36, conforme horário oficial de Brasília, com fundamento no art. 5º do Decreto nº 10.543, de 13 de novembro de 2020.

A autenticidade deste documento pode ser conferida no site https://sei.ufmg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **3768578** e o código CRC **0F61B174**.

To my parents.
Aos meus pais.

Acknowledgments

I thank God for the opportunity to complete this important stage of my life, where I could meet wonderful people like my teachers and colleagues.

I would like to express my heartfelt gratitude to The Boss, my project advisor, Professor PhD Antônio de Pádua Braga, for his invaluable teachings, unwavering support, encouragement, and guidance throughout the entire journey. His powerful embedded neuronal classifiers, predictors, and constructive instructions have been primordial in determining all my journey. I would like to express my deepest gratitude to him for receiving me in the Lab, believing in the viability of the work and being my mentor. My Teacher, I really appreciate that. Thank you very much!

I am immensely grateful to the deputy Lab Boss, Professor Fred Coelho, for all the support and teachings. Thanks to all the teachers for the zealous preparation and exposition of the classes.I would also like to thank Marcelo do Vale Menezes, Júlio César Ferreira (Chicó) and José André Gomes Junior (Zezicha) for the opportunity to study a PhD and for encouraging the personal and professional development of the entire team.

Thanks to all dear LITC colleagues for all the companionship, shared work, teaching and relaxed moments. I really appreciate all the knowledge shared. I especially thank Professor and colleague Luiz Carlos Bambirra Torres , whose doctoral work inspired a lot in my projects.

I would like to thank dominae meae, Luana Tanus, for her love, friendship, and patience. I thank the boy Gibran, "The Cricket's Chassi", Tanus for his affection, sparrings, games, and complicity. Thanks to Mrs. Lúcia Morais, my mother-in-law, for her powerful motherly prayers on my behalf, and to my mother and sisters for their eternal love, affection and logistical support.

I would like to express my sincere gratitude to the examination board members for your time and dedication to improve my research. Your suggestions will made a significant improvement in the quality of my work.

Thanks to Mr. João Silva, Mrs. Maria Santos and the glorious Brazilian people for the financial contribution destined to our university.

"For God so loved the world, that he gave his only begotten Son, that whosoever believeth in him should not perish, but have everlasting life."

– John 3:16

"Porque Deus amou o mundo de tal maneira que deu o seu Filho unigênito, para que todo aquele que nele crê não pereça, mas tenha a vida eterna."

– João 3:16

RESUMO

A característica que mais diferencia os seres humanos de quaisquer outras entidades conhecidas reside no arsenal cognitivo proporcionado pela mente que é instanciada em seu complexo cérebro. Esse conjunto de ferramentas cognitivas dos seres humanos compreende o que se denomina de inteligência. Um aspecto fundamental da inteligência humana é a capacidade de conduzir investigações e descobertas científicas, trazendo inúmeros benefícios à humanidade, desde tratamentos médicos até tecnologias de exploração espacial. Para facilitar, disseminar e amplificar o uso das capacidades cognitivas humanas, pesquisadores têm buscado automatizar esses processos por meio da Inteligência Artificial (IA). Esses esforços, amparados pela tecnologia atual, estão revolucionando a pesquisa científica, tornando-a mais rápida e descentralizada, sobretudo com o uso das IA Generativas (Gen-AI). No entanto, para que modelos de IA sejam amplamente adotados a processos de descoberta científica e em aplicações de missão-crítica, é imprescindível torná-la compreensível aos seres humanos, o que é o foco da Inteligência Artificial Explicável (XAI). Diversas técnicas foram desenvolvidas para prover interpretabilidade a modelos de IA, como a geração de regras lógicas simbólicas, interpretações por meio de gráficos e imagens, descrição da importância das variáveis e análises por meio do uso de amostras (exemplos), destacando-se a Análise Contrafactual. A análise contrafactual beneficia modelos de IA em aspectos como análise do comportamento do modelo, verificação de erros, avaliação de equidade e vieses, incremento de dados, extração de regras simbólicas, estudos de otimização, além de fornecer subsídios para ajustar dados de amostras de modo que se obtenha uma saída (predição) distinta do modelo. Este trabalho investiga o uso de estruturas geométricas, especialmente o grafo de Gabriel, para balizar a realização de análises contrafactuais. O grafo de Gabriel permite representar a similaridade entre as amostras de um conjunto de dados e é adotado para direcionar a criação de exemplos contrafactuais sob diferentes restrições e comportamentos das dimensões espaciais. A ideia central é selecionar estrategicamente pontos reais de referência baseando-se na estrutura geométrica do grafo associada às métricas estatísticas das amostras. Uma vez selecionada a amostra de referência, pode-se escolher diferentes estratégias de criação das novas instâncias contrafactuais. A avaliação da abordagem com conjuntos de dados reais evidenciou a eficácia do uso de estruturas de dados baseadas no grafo de Gabriel na geração de exemplos contrafactuais e ressaltou o potencial de estruturas geométricas em iniciativas de explicabilidade. Contudo, a área ainda demanda esforços adicionais, especialmente no desenvolvimento de algoritmos para lidar com dados complexos e de elevada dimensionalidade.

Palavras-chave:inteligência artificial (ia); inteligência artificial explicável (xai); análise contrafactual; geomatria computacional; grafo de Gabriel; estruturas geométricas; modelos de ia; interpretação de modelos; medidas de similaridade; otimização.

Abstract

The present thesis investigates the application of geometric structures, particularly the Gabriel Graph, for the generation of counterfactual examples to enhance explainability in Artificial Intelligence (AI) models. The research explores the potential of these structures to improve model interpretability by generating counterfactual samples that are realistic, diverse, and feasible, enabling better insights into decision boundaries and influencing features. The methodology involves the use of geometric data structures and statistical metrics to guide counterfactual generation, employing real-world datasets for validation. Results indicate the efficacy of the proposed methods in producing high-quality counterfactuals and emphasize the computational efficiency and applicability of geometric approaches in handling complex datasets. The findings highlight the role of geometric techniques in advancing Explainable AI (XAI), particularly in critical domains like healthcare, finance, and industrial systems, while identifying future research directions, such as developing efficient algorithms for high-dimensional data.

Keywords: artificial intelligence (AI); explainable artificial intelligence (XAI); counterfactual analysis; computational geometry; Gabriel graph; geometric structures; AI models; model interpretation; similarity measures; optimization.

List of Figures

List of Tables

List of Acronyms

AI: Artificial Intelligence

ANN: Artificial Neural Network

AUC: Area Under the Curve

CI: Computational Intelligence

CO: Convex Optimization

CSS: Computational Social Science

DE: Differential Evolution

DS: Data Science

DS: Dominating Set

Feas.: Feasibility

GDPR: General Data Protection Regulation

GG: Gabriel Graph

GGClassification: Gabriel Graph Classification

HCI: Human-Computer Interaction

HE: Heuristic

HMI: Human-Machine Interface

KDD: Knowledge Discovery in Databases

LIME: Local Interpretable Model-Agnostic Explanations

ML: Machine Learning

NRE: Not Required

OHE: One-Hot Encoding

OPT: Optional

PDF: Probability Density Function

Prox.: Proximity

RC: Reinforced Concrete

REQ: Required

R: Range

RNN: Recurrent Neural Network

SE: Support Edge

SHAP: SHapley Additive exPlanations

Spar.: Sparsity

SS: Satisfiability Solver

SVM: Support Vector Machine

TC: Threshold Criteria

TC+: Positive Class Threshold Criteria

TC-: Negative Class Threshold Criteria

Vor: Voronoi Diagram

W: Weights

XAI: Explainable Artificial Intelligence

Contents

Chapter 1

Introduction

1.1 Context Analysis and Guiding Questions

The widespread and growing dissemination of Artificial Intelligence (AI) has significantly increased the demand for explainability and interpretability. As AI models become more complex, understanding the rationale behind their decisions has become critical for fostering trust and ensuring ethical deployment. Different strategies for understanding the behavior of AI models have been developed over time, among them has emerged Counter Factual Examples Investigation. Counterfactual investigation has emerged as a prominent methodology for providing interpretability and explainability in AI systems, enabling the exploration of hypothetical scenarios to clarify decision-making processes.
While numerous methods exist for conducting counterfactual analyses, approaches leveraging geometric methods combined with statistical metrics remain rare. The present study aims to address this gap by exploring the utility of geometric data structures for generating counterfactual examples. This research is motivated by the potential of geometric methods to not only produce high-quality counterfactual samples but also to provide insights into the decision process of machine learning models. In order to highlight the work in right place, one may consider employing the typology of of XAI systems proposed by Cabitza et all [1]. Such topology has a practical approach, instead using theoretical and speculative concepts, the authors have considered a naively constructed definition of explanation for the aims of explainable AI and designed a software architecture point of view. The present thesis research new branch is highlighted on the Figure 1.1.

1.2 Hypotheses and Objectives

The primary objective of this research is to investigate whether geometric data structures, when associated with statistical metrics, can facilitate the generation of counterfactual samples that meet specific quality requirements. The study hypothesizes that geometric approaches can:

- Generate counterfactual samples that are realistic, diverse, and feasible within the data domain.
- Enhance the interpretability of classifier decision-making processes by visualizing decision boundaries and influential features.
- Identify the most suitable geometric data structures for different dataset characteristics.

- Demonstrate computational efficiency suitable for real-time systems subject to temporal constraints.

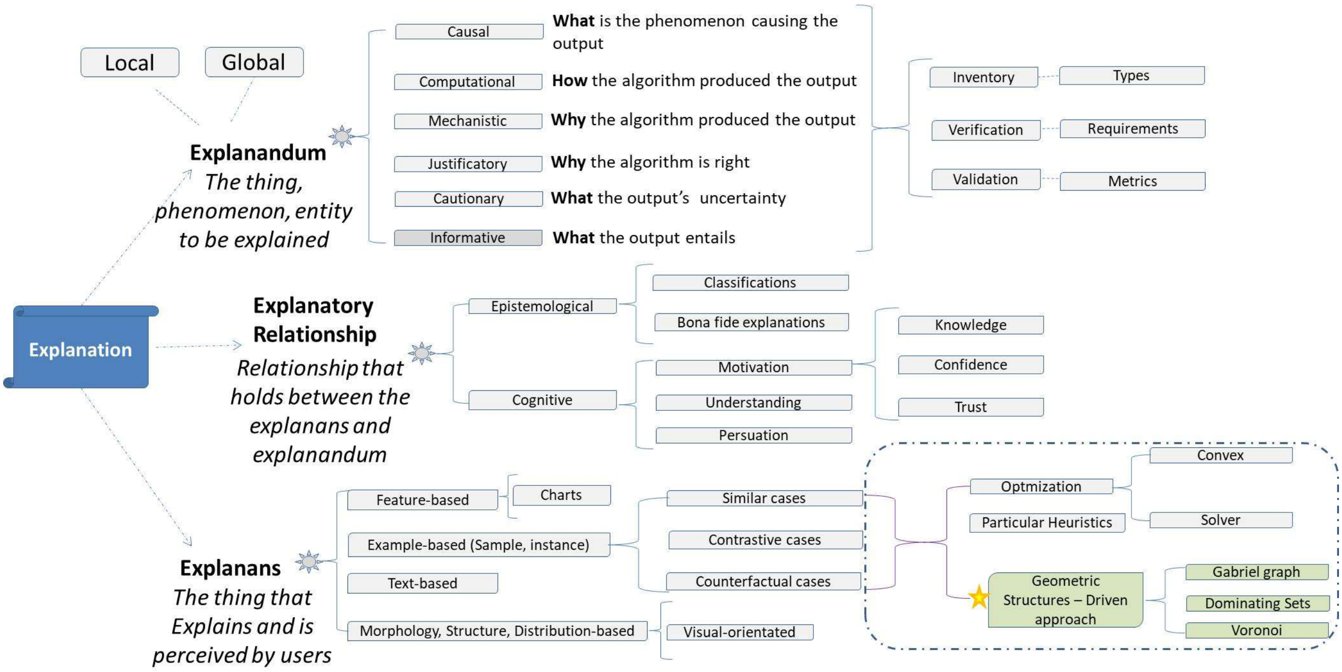


Figure 1.1: Thesis approach placed within *Cabitza et all Topology* [1] highlighted in green.

1.2.1 General Objectives

In general terms this work aims to validate the potential of geometric data structures for generating counterfactual samples that enhance the interpretability and explainability of AI models.

1.2.2 Specific Objectives

On can subdivide the objectives into the following detailed view:

1. To design and implement geometric methods for generating counterfactual examples.
2. To evaluate the performance of these methods across diverse datasets.
3. To identify the most effective geometric structures for counterfactual generation under various dataset characteristics.
4. To quantify the quality of counterfactual samples using predefined metrics, such as feasibility, diversity, and sparsity.
5. To assess the computational efficiency of the proposed methods in real-time settings.

1.3 Work Methodology

This research employs the following systematic methodology to achieve its objectives:

- **Theoretical Framework:** A comprehensive review of literature on counterfactual analysis, geometric methods, and statistical metrics to establish a strong theoretical foundation.
- **Observations of theoretical concepts:** Computational experiments for observations of theoretical concepts.
- **Methods Development and Implementation:** Design and implementation of geometric methods for counterfactual generation, leveraging data structures such as Gabriel graphs, Dominating sets, Voronoi diagrams, and convex hulls.
- **Experimental Validation:** Execution of computational experiments using benchmark datasets and a real-world dataset from an industrial steelmaking process.
- **Evaluation Metrics:** Validation of counterfactual samples based on feasibility, diversity, proximity, and interpretability.

1.4 Relevance of the Work and Contribution

This work is relevant to advancing the field of Explainable AI (XAI) by introducing novel geometric methods for counterfactual analysis. These methods hold potential for practical applications in critical domains such as healthcare, finance, and industrial systems, where interpretability and trustworthiness are paramount. Additionally, the study bridges a gap in the literature by systematically combining geometric structures with statistical metrics, providing a new perspective for counterfactual generation.

1.4.1 Areas of Applicability: Potential Applications and Stakeholders

The methods and results of this study have the potential to serve a wide array of applications. Several distinct research areas that could benefit from this work include:

- Explainable AI (XAI).
- Data science (DS) ans Machine Learning (ML).
- Safe and ethical AI systems.
- Computational social science (CSS).
- Automatic scientific discovery and knowledge discovery in databases (KDD).
- Healthcare.
- Industrial applications (aerospace, automotive, and manufacturing process).
- Enhanced human-computer interaction (HCI).
- Recommendation Systems

By addressing the challenges of counterfactual generation through geometric methods, this research open new roads for advancing the interpretability and trustworthiness of AI systems.

1.4.2 Additional Contributions

Throughout the research activities, additional contributions were made through publications and review efforts. Some of these contributions can be found in the following list:

- Carneiro, M. V., Salis, T. T., Almeida, G. M., & Braga, A. P. (2021). Prediction of Mechanical Properties of Steel Tubes Using a Machine Learning Approach. Journal of Materials Engineering and Performance, 30(1), 434-443.

- Alvarenga, W. J., Campos, F. V., Costa, A. C., Salis, T. T., Magalhães, E., Torres, L. C., & Braga, A. P. (2022). Time domain graph-based anomaly detection approach applied to a real industrial problem. Computers in Industry, 142, 103714.

- Silva, D. A. C., Salis, T. T., & Braga, A. P. (2021, November). Industrial case study of causal modeling of continuous casting and lamination of steel tubes. In 2021 IEEE Latin American Conference on Computational Intelligence (LA-CCI) (pp. 1-6). IEEE.

- Durãaes, R. G., Salis, T. T., Coelho, F. G. F., & de Pádua Braga, A. (2022, July). Explainability Analysis of a Machine Learning Model for Industrial Applications. In 2022 International Conference on Electrical, Computer and Energy Technologies (ICECET) (pp. 1-6). IEEE.

- Silva, D. A. C., Salis, T. T., & Braga, A. P. Structural Causal Model as Alternative to Black-Box Approaches Applied to an Industrial Problem" for possible publication in Neural Computing and Applications. Neural Computing and Applications (April, 2022)

Not published. Finished, Under revision articles:

- *Vitor M. Hanriot, Turibio T. Salis, Luiz C. B. Torres, Frederico Coelho, Antonio P. Braga. Large Margin Classifier with Graph-based Adaptive Regularization Threshold for Autonomous Machine Learning. Patter Recognition Letters. (Under revision by the journal)*

- *Salis, T. T., Braga, A. P. Explainable Artificial Intelligence in Industry: practical case studies and survey. (Originally submitted to Engineering Applications of Artificial Intelligence, transferred to Elsevier Expert Systems with Applications. Submission In progress)*

Paper in preparation (following the evolution of current experiments)

- Salis T.T., Balbino M., Torres L. C. B., Nobre C., Braga, A.P. Formulation of counterfactual explanations from characteristics' space geometric representations.

Papers Review:

- TNNLS-2022-P-20342.R2 entitled "HGBER: Heterogeneous Graph Neural Network with Bidirectional Encoding Representation" for the IEEE Transactions on Neural Networks and Learning Systems.

- TNNLS-2020-P-14486.R2 entitled "An Intelligent Method for Predicting the Pressure Coefficient Curve of Airfoils based Conditional Generative Adversarial Networks" for the IEEE Transactions on Neural Networks and Learning Systems.

- EAAI-22-4828 "A Review of Wearable Sensors Based Fall-Related Recognition Systems", for Engineering Applications of Artificial Intelligence
- Springer Nature. "A Parallel Reconfigurable Architecture for Scalable LVQ Neural Networks" for Neural Processing Letters
- Springer Nature. "A Targeted Universal Attack on Graph Convolutional Network by Using Fake Nodes" for Neural Processing Letters
- Engineering Applications of Artificial Intelligence. EAAI-22-90 Computational approach towards shear strength prediction of RC shear wall using ensemble and hybrid SVM models" (EAAI-22-90) to Engineering Applications of Artificial Intelligence
- Springer Nature "A New Convolutional Neural Network Architecture for Automatic Segmentation of Overlapping Human Chromosomes" for Neural Processing Letters.

1.5 Thesis Structure

Chapter 2 aims to explore the context of Explainable Artificial Intelligence (XAI), highlighting its historical evolution, philosophical foundations, and practical applications. It addresses the importance of explainability in AI systems to foster trust, understanding, and transparency. It discusses concepts such as explanation, understanding, and trust, linking them to human needs and the ethical and legal requirements of AI-based systems. Additionally, it examines the role of counterfactual thinking as a cognitive tool for understanding decisions and indicating changes in AI models' training process. Finally, the text provides a comprehensive view of XAI techniques, emphasizing the relevance of counterfactual explanations as an intuitive and accessible way to interpret complex models.
Chapter 3 delves into the application of counterfactual explanations (CE) within Explainable Artificial Intelligence (XAI), emphasizing their utility in understanding AI model decisions and proposing actionable changes to achieve desired outcomes. It explores various methodologies for generating counterfactual examples, their role in improving model interpretability, and their application across diverse fields, such as finance, healthcare, and image classification. The chapter also highlights challenges, including feasibility, plausibility, and actionability of counterfactuals, as well as risks like data privacy concerns and adversarial vulnerabilities. Furthermore, it outlines criteria for evaluating the quality of counterfactual explanations, offering insights into their effectiveness as a tool for decision support and model refinement.In general, this chapter reviews the literature on methods for generating counterfactual examples.
Chapter 4 presents the theoretical foundation and application of geometric structures, starting from fundamental concepts such as points and dimensions, visiting mathematical abstractions like distances and their variations, to advanced structures such as Gabriel graphs, Voronoi diagrams and Dominating sets. It explores their role in machine learning and pattern recognition, illustrating how these structures are used to analyze data and classify points in high-dimensional spaces. The discussion is enriched with computational experimental insights into the challenges posed by high-dimensional data, such as the curse of dimensionality, and highlights the impact of metrics and thresholds in graph-based classification methods. This section establishes the geometric and computational framework that supports the proposed approaches.

Chapter 5 details novel approaches for generating counterfactual examples and perform counterfactual analysis in classification problems through the use of geometric structures, such as the Gabriel Graph, associated to statistic metrics.
Chapter 6 describes the performance of computational experiments to validate the performance of the proposed geometric methods in order to validate the hypotheses raised by the work. The tests involved testing the performance of the proposed methods against 20 different real databases.
Chapter 7 exposes the conclusions conclusions obtained from the work. Chapter 8, in turn, points out several routes for future work and points still open that can be addressed by new researches. Appendix A comments XAI history and main topics.
Appendix B outlines the historical development of the most common methods for achieving explainability in AI. It also describes a series of experiments using real-world databases to validate such methods. Finally, the chapter briefly presents several actual industrial case studies involving XAI besides diverse referenced in the bibliography.

Chapter 2

Counterfactual Analysis in Explainable AI: Foundations, Evolution, and Key Insights

The characteristic that most differentiates human beings from everything else is the rich cognitive arsenal provided by their minds. This set of mental features embraces the concept of what we call intelligence.

The cognitive capacity of the human mind is the key to understanding reality and making decisions based on conscience rather than individual moment passions. This conscious decision-making is dependent on our ability to organize concepts and discern between good and evil, beauty and ugliness, right and wrong, or, at a higher abstraction level: better versus worse.

When individuals neglect their cognitive abilities, their actions can lead to negative consequences. Philosopher Hannah Arendt, in her concept of the 'banality of evil' [5], suggests that evil often arises from a lack of genuine thought. Failing to exercise this distinctly human capacity inevitably leads to moral decay and potentially catastrophic outcomes. It is our consciousness, our ability to think, reason, and discern, that defines us as human.

Beyond moral discernment, human cognitive capacity enables the development of methods to observe and understand reality to discover the truth. This course generated philosophy in ancient Greece, encompassing disciplines such as logic, reasoning, and argumentation, which are fundamental in pursuing truth. Philosophy initially aimed to establish an inner ordering of the thinking, mind and soul [6].It developed forms of logical reasoning for guaranteeing deductive process correctness, understanding, and explaining the observed reality [7].

The search for truth and the means to understand it led to the development of the scientific method. Such a systematic approach allowed humanity to explore and comprehend the natural world. This method has been a catalyst for numerous benefits for mankind, improving the quality of life from essential medical treatments, increasing food production, passing by to enhancing transportation and knowledge dissemination and even enabling spatial cutting-edge technologies. This progress is a testament to the power of human cognitive capacity.

Seeking to amplify its potential and disseminate the use of intelligence (just as it was done for activities involving physical strength), researchers from different areas investigate ways to automate, facilitate, escalate, and spread human cognitive skills. Initiatives to enhance and mechanize human intelligence features, primarily through computational means, gave rise to research areas such as Computational Intelligence (CI), and Data Science (DS). Using an exaggeration for the sake of didactics, one could say that one goal of such automation would be to "*find a cure for a disease from a command prompt.*"

Many AI-related tasks involve understanding a given reality through the data that represents it. These tasks consist in a way of speeding up the discovery of knowledge about a specific phenomenon. In other words, it attempts to expedite scientific discovery by data-driven approaches. The term "Scientific discovery" here will be limited to "formulating and validating hypotheses seeking to explain observable phenomena."

Concerning simple causal relationships between binary categorical variables, for example, the pioneering works on the automatic generation of hypotheses date back to the 60s [8]. In some more complex situations, where data sets contain a mix of integer, continuous, and categorical variables, the distinguished tool is a subset of AI called Machine Learning (ML). ML methods have considerable learning power and generalization capacity, working as universal function approximators, and are successfully applied in several experiments.

This way, ML models could naturally represent an excellent option in attempts to automate the knowledge-discovering process, phenomenon prediction, and hypotheses proposing.However, several high-performance ML models have notable drawbacks inherent in their way of knowledge representation. The problem is that interpreting, understanding, and explaining the knowledge assimilated by particular models is not a nominal feature. Such a model's original design did not include explanation functionalities. For example, explaining, in a human-friendly language, the knowledge embedded in neuron activation values and synaptic weights across the entire artificial neural network topology, is a challenging task.

Automatic knowledge discovery via ML is feasible, but issues in explaining the models' reasoning process to humans hinder it. AI models often operate as "black boxes," making it challenging for users to understand how they arrive at their decisions [9]. Some "non-explainable" models are employed accepting their black-box condition. But, when the scenario asks for justifications, such models are restricted. For example, in mission-critical systems, healthcare, financial transaction decisions, atomic power plants, civil constructions [10], and hazardous (classified) areas controlling [11, 12].

In order to adopt a specific AI model as part of automatic scientific discovery methods, automatic hypotheses formulation, or critical-mission systems, it is necessary to make their behavior understandable. Providing AI model explanations to end users represents a key challenge for AI system design [13].It is necessary to enhance transparency, explaining the underlying mechanisms of AI models so users can comprehend it, building, this way, trust in the models' outputs [9]. The models' explainability contributes to promoting trust in AI [14].

Aiming to build trust in AI through explainability, researchers have produced a particular study field denominated Explainable artificial intelligence (XAI). XAI has received growing attention, as one can see from the number of works and events involving the topic [4, 14–18]. Nowadays, XAI initiatives seek to meet demands from distinct audiences. The primary needs involve:

- Justifications, seeking to support the results (reliability, trust, acceptance).
- Models Training improvement (maximize accuracy, identify and avoid biases and errors).
- Elucidate findings and formulate hypotheses, causality (automatic knowledge discovery and communication).
- Law, ethics, transparency and justice (justify outputs and propose feasible actions to change them). Laws[1] in various countries and regulations of different institutions and companies address the issue of data protection and the use of artificial intelligence.

[1]It is worth mentioning that the explainability subject is also involved in social, ethical [19], and legal issues

- Interactivity targeting to facilitate human-AI mutual understanding. (good human-in-the-loop fluidity).

- Simulate different scenarios ("possible worlds").

Current year publications indicate some topics regarding the explanations main objectives:

- Explanations should be tailored to users' cognitive styles, social-cultural-cultural factors, and AI literacy levels [27, 28].

- The complexity and depth of explanations should be adjustable based on user expertise and task complexity [29].

- XAI systems should provide both system-level and prediction-level information to build users' mental models to support understanding [29].

- Explanations must reflect the underlying data and model accurately [30].

- User controlling the level of explanation is important [29].

- XAI explanations should enhance understanding, trust, accountability, and transparency [31].

- Even when users struggle to understand explanations fully, they may still prefer their inclusion and perceive benefits such as skill improvement and trust increasing [32].

As explainability took center stage, knowing such a concept closer became interesting. So, what exactly is an explanation? In order to address this inquiry, it is necessary to delve into the domains of philosophy to some extent. It will be discussed in the next section. To see details on the XAI development, survey works and XAI main techniques, please see Appendixes A and B.

2.1 Explaining Explanation, Understanding and Trust

It was noted that XAI research aims to generate *explanations* that clarify AI models decision-making, promoting user *understanding* to build *trust*. However, in a rigorous scientific context, this statement is vague due to the lack of consensus on the concepts of explanation, understanding, and also trust, in both of its typical investigation areas: philosophy and psychology.

Delving deeper into these concepts is crucial, even if their initial clarity is limited, because they are fundamental to the current research. Gaining a clearer understanding of these concepts will enhance confidence in the methodological approach and ensure the investigation remains on the correct trajectory. That said, the main points of this investigation will now be addressed.

[20, 21]. Regulations regarding AI models' decision justification already have drawn the attention of geopolitical leadership centers legislation, for instance, the General Data Protection Regulation (GDPR) from European Union [22]; the Code of Federal Regulations, and the specific executive order on AI in the USA [23,24]; and The Personal Data Protection Law in Brazil [25]. Such regulations impact the activities of machine learning professionals who must meet the requirements that now have law power. While these laws can impose challenges, they also provide opportunities for ML researchers to take the lead in designing algorithms that supply explanation features [26].

2.1.1 About trust

As it was exposed before, in terms of Human-Machine Interface (HMI), the model's behavior understanding and trusting is a requirement of the human psychological factor in the loop. Following this behavior, in everyday life, one tend to trust people more when they can explain their actions [33]. Individuals who clearly express their feelings, intentions and have a tendency to accurately express their emotions and to reveal their inner mental states are more likely to be seen as trustworthy [34]. In this sense one can rise two questions: is there any formal criteria to determine when it is justified to trust someone? And, before that, What is trust?

Trust is a significant topic in psychology and philosophy because it is essential for the success of various human activities, including teamwork in companies, business, politics, religion, scientific research, and the transmission of knowledge. Nevertheless, the concept of trust does not have a narrow area or an obvious boundary. PytlikZillig and Kimbroughb, in their vast survey work [35], have compiled some common elements from many definitions. One can see their statement bellow:

> Common essences of trust include that trust involves a trustor (subject) and trustee (object) that are somehow interdependent; involves a situation containing risks for the trustor (which also implies the trustor has goals); is experienced by the trustor as voluntary (implying autonomy, agency, and intrinsic motivation); and includes (or excludes) different types, forms, or sources of trust concepts, some of which may form the bases of others, and many of which involve or relate to positive evaluations or expectations [35].

One can see a visual representation of such explanation in Figure 2.1.

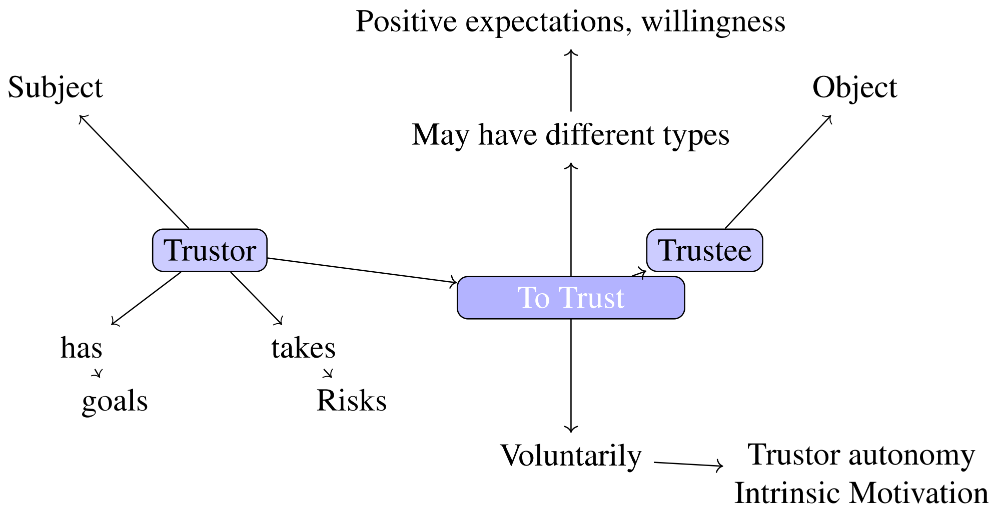


Figure 2.1: Common essences of trust in different conceptualizations.

Trying to compile a direct definition for Trust, using the common elements, one can suggest: "Trust is a psychological state in which an individual presents a good willingness to accept vulnerability or potential losses, driven by a positive expectation of another person's intentions or actions." In this sense, the greater this willingness, the stronger the trust in the other party [36]. It entails reliance upon something else, and the belief that this other will not fail in meeting certain trustor's expectations [33].

PytlikZillig and Kimbroughb comment in their review that researchers continue to engage in debates over various aspects of trust, particularly [35]:

- The types of relationships that must be present for psychological or behavioral states to qualify as trust.
- Whether all conceptualizations of trust necessitate elements regarding to risk, conscious deliberation of risk, volition, and/or active choice on the part of both the trustor and trustee.
- The extent to which risk and trust can be considered separable concepts.
- The distinction between trust as a psychological phenomenon versus a behavioral one.
- The cognitive versus affective dimensions of trust.
- The previous trustor-trustee relationship and expectations about future actions.
- The specific requirements from which trust must originate.

Trust versus reliance

The concept of trust encompasses elements such as emotions, interpersonal relationships, and risk awareness. When referring to inanimate objects, such as IA models, electronics stuff, the sign *trust* may not be the most appropriate. In these cases, a more superficial and functional one is often more suitable: *reliance*.

Some researchers have explored this distinction [33, 37–39]. They have identified a key difference:

Trust involves a conscious decision based on the assessment of risks and available alternatives. It implies a deliberate evaluation of options and a subsequent choice to depend on a system or individual, with full awareness of the potential risks involved.

Confidence (or reliance), in contrast, pertains to a general sense of security or safety, often without the active consideration of risks or alternatives. It reflects an expectation that things will function as foreseen, primarily grounded in past experiences or established norms.

Summarizing: The essential difference is that in case of trust, a *decision is made* to rely or not to rely on the person or system [33].

Talking about the similarity, it is possible to say that trust is an iterative process where individuals gather *information* and adjust their judgments based on new insights. The idea that confidence involves evaluating *evidence* (previous experience) suggests that the processes of "building trust" and "building confidence" are similar [40]. In this sense, the explanations are suppliers of information for trust and evidences for confidence.

In conclusion, the terms trust and confidence are closely correlated yet distinct in certain aspects. Hence, the mission of XAI, previously described as *building trust from explanation*, could alternatively be framed as *building confidence from explanation*, depending on the context. Having said that, one have sufficient trust... I mean, confidence...that is, assurance to proceed.

2.1.2 About understanding

Understanding is the cognitive process by which an individual grasps the meaning, significance, or nature of something. It involves the ability to integrate information [41], form mental models using concepts [42], and apply such concepts in a coherent and context-sensitive manner [43]. Understanding goes beyond mere knowledge acquisition [44–47]. It requires connecting ideas, seeing relationships between them, validating the process and explaining why things are the way they are [48].

Distinct areas researches agree that understanding is a multifaceted cognitive process involving a set of mind-related elements. It requires the formation of mental models, which are developed through internal experiences, imagination, believes and the analysis of both linguistic information. This mind process goes beyond mere word decoding, requiring the integration of text with its broader [49, 50].

Understanding is shaped by three key factors: the generation of thought, the transformation of personal knowledge and experiences, and the individual's mental representation of the world [51].

Understanding process is closely tied to inference and memory activities, which include activating relevant knowledge, maintaining information, and retrieving it from long-term memory [52, 53].

Other important element regarding understanding is mental modeling. This is a distinct cognitive capacity in humans,facilitates unique forms of learning and prediction [54]. In the context of physics education, the construction of accurate mental models of physical phenomena is essential in conferring understanding. Consequently, pedagogical strategies could emphasize explicit modeling processes to enhance comprehension [55].

Bringing together the main elements previously discussed, the Figure 2.2 shows a flowchart to illustrate the process of constructing understanding.

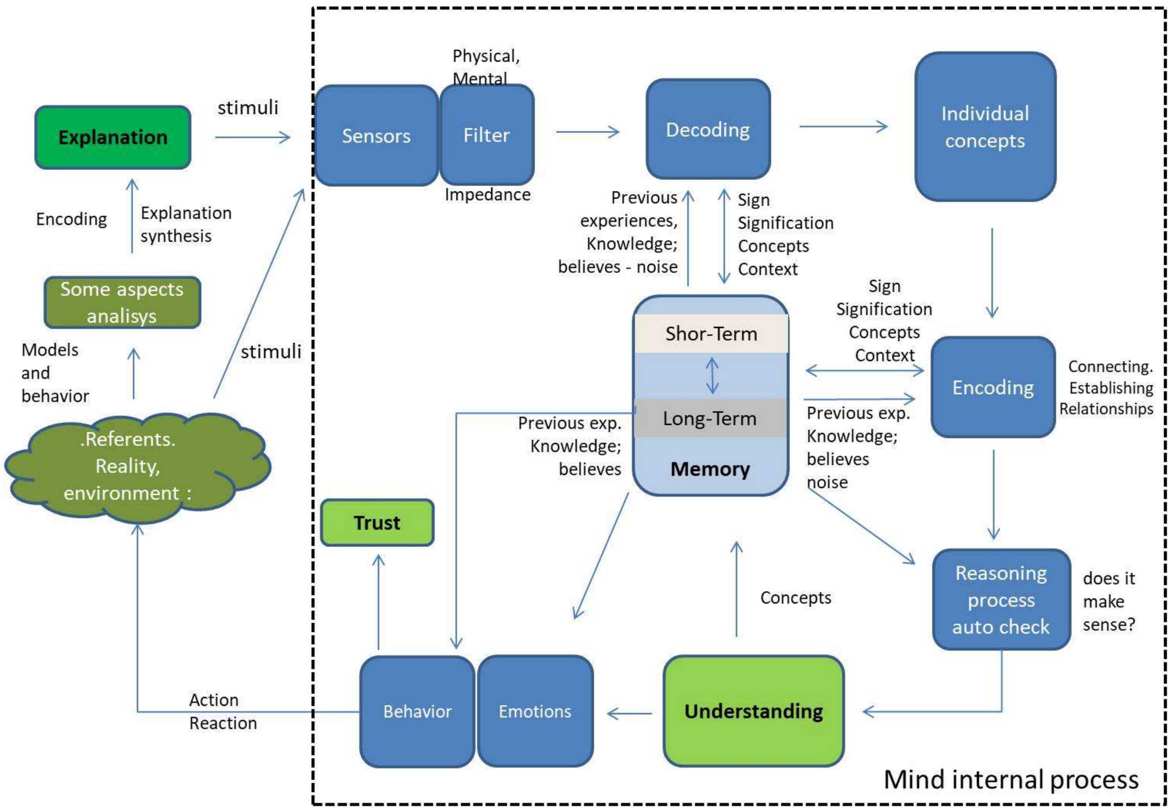


Figure 2.2: Mental process of the understanding building and trust establishment.

The presented flowchart appears to offer a conceptual representation of the mind's internal processes of *understanding* generation. It indicates the real world interpretation, the decoding and encoding of conceptual structures and relationship, the reasoning process and the generation of understanding. In essence, it illustrates how external environmental stimuli are continuously received, processed, and integrated into a dynamic system of perception and reasoning to produce understanding.

Stimuli from the Real World: The understanding generation process begins with the external reality perception. The things (semiotics' *Referents*) in the real world, encompassing the environment, objects,live beings, actions, events...constantly provides an array of stimuli to human minds. These stimuli may include visual or auditory signals, perceptual elements and some few aspects of the reality used to synthesize explanations. Note that, for illustrative purposes only, *explanation* here refers to the behavior/outcome of a given model. The model outcome figures as the *explanandum* and the explanation is the input stimulus. On can see the process in more detail below.

Sensors and Filter (Sensors & Filter): Raw information is first captured by the individual's "sensors" (physical senses and cognitive faculties, such as attention) and then passes through a mind internal filter. This psychological and cognitive filter selects, weighs and interprets incoming stimuli based on prior experiences, beliefs, values, current mental states, and other contextual factors. This mechanism serves to reduce noise and maintain focus on what is deemed most relevant. However, it can filter (diminish or completely remove) important elements due to psychological aspects of the individual.

Decoding (Decoding): Once the information has passed through the filter, it is decoded into elementary concepts/entities. This process involves assigning initial meanings, discerning patterns, and try to relate these inputs to pre-existing concepts. Such decoding is influenced by memories, established knowledge, and belief systems. Here, once again, the individual's belief system and psychological state can introduce noise into the process.

Memory (Short-Term and Long-Term Memory): Short-term and long-term memory systems work in tandem. Long-term memory stores past experiences, knowledge, and beliefs, whereas short-term memory functions as a "mental workspace" where newly decoded information is temporarily held for processing. This ongoing interaction between memory and decoding facilitates the formation or reformation of conceptual understanding.

Concept Formation (Individual Concepts) and Encoding (Encoding): Once information is decoded and connected to existing knowledge structures, it is transformed into individual concepts. Subsequently, these concepts undergo an internal encoding process, wherein connections and different types of relationship between ideas are established.

Reasoning Process: At this stage, internal reasoning mechanisms verify whether the emerging understanding "makes sense" in light of previously established knowledge and logical validation. This acts as an internal consistency check, ensuring coherence and logical alignment with the individual's conceptual framework.

Understanding and Trust (Understanding & Trust): Ultimately, this process yields understanding, a consolidated comprehension of the given phenomenon or information. Building upon this understanding, the individual's emotional responses, behaviors, and the degree of trust in the acquired interpretation begin to form or strengthen.

Emotions, Behavior, and Feedback to the Environment: The culmination of these internal cognitive processes can trigger emotional states and behaviors, which in turn influence the external environment through actions.

Summary: The flowchart illustrates the complex dynamics by which the human mind generates the state of understanding that supports the disposition to trust. It embraces receiving environmental stimuli, filtering and processing them, constructing concepts, fostering understanding, experiencing emotions, and ultimately driving individuals' behavior. All these internal operations are continually shaped and molded by prior experiences and accumulated memories.

The way an explanation is compound can significantly influence how it is filtered, decoded, encoded, and ultimately understood. A carefully structured explanation can reduce cognitive load, avoid communication personal impedance, facilitate the decoding/identification of elementary concepts, facilitate the relationship between individual concepts, thus enhancing the overall internal reasoning process to produce understanding. On can find bellow some key principles to consider:

- **Clarity and Simplicity:** Using clear, concise language, adjusting necessary jargon (proper usage of signs/semiotics). Simple and direct structure reduces cognitive load and noise, ensuring essential content is preserved through initial perception. Clarity/properly usage of signs make the decoding process easier.

- **Contextualization and Relevance:** Embedding previous information and user's familiar contexts. Recognizable patterns linked to prior knowledge streamline decoding, facilitating smoother integration of new concepts.

- **Logical Structure and Coherence:** Organizing content with a clear flow, from introductory statements through logically sequenced points, to a concise conclusion. This coherence aids internal reasoning and comprehension.

- **Use of Examples and Analogies:** Anchoring abstract ideas with concrete examples and analogies. Linking unfamiliar material to well-understood domains accelerates encoding and strengthens retention.

- **Chunking and Segmenting:** Breaking down complex explanations into manageable segments. Smaller units prevent working memory overload and facilitates decoding and encoding process leading to coherent understanding.

- **Meaningful Redundancy:** Reinforcing key concepts through diverse presentations. Meaningful repetition—varying perspectives or contexts—enhances encoding and reduces forgetfulness.

- **Metacommunication and Organizational Cues:** Use signals and markers (e.g., previews, summaries, connection sentences/paragraphs) to guide attention. Such cues help form a mental roadmap, improving coherence checking and integration.

- **Establishing Credibility and Trust:** Building trust through authoritative, reliable sources. Confidence in the explanation's origin encourages open-minded filtering reducing (mind filter/impedance) and deeper processing.

- **Alignment with the Listener's Goals and Intentions:** Highlighting personal relevance and practical benefits. Demonstrating usefulness encourages sustained focus and effective reasoning.

Regarding the correlation between explanation and understanding,one may say that, to understand why a phenomenon occurs means having an accurate explanation for it. Grasping a scientific theory involves being able to create, or at least comprehend, the various possible explanations this same theory offers for different phenomena [56]. Therefore, scientific understanding always depends on having a scientific explanation.

The relevance of explanation in constructing understanding of scientific discovery has been demonstrated thus far. Given that both knowledge embedded in data and insights into model behavior are considered scientific discoveries, it becomes essential to conduct a deeper investigation into the nature of the explanation.

2.1.3 About explanation

The etymology of the word revels that it comes from the Latin prefix *ex-*, the means out, or to put it out, plus *plicare*, that signifies to fold. When someone *explica* (takes the folds out) something, it is like unfolding what is rolled up, twisted, or crumpled so that it is impossible to understand what it is about.

Other sources can indicate that such a word comes from the Latin *explanare*. To explain means make it clear, make it plain. Latin prefix *ex* (from in to out) plus *planus* (flat), generating the concept of "making it flat, playing". Despite any variations, the central concept remains the same: It's about undoing folds, untying entanglements that hide something, making it intelligible. It refers to revealing things that are underlying, bringing to light something that is currently in the shadows, to reveal.

Tracking back the origins of explanation from a philological perspective, it becomes clear that throughout much of history, people understood explanation in terms of causation. Explaining a given reality, event, or phenomenon involves identifying its truth cause. In this sense, explaining would mean *reveal the cause*.

Since the ancient philosophers, for example, Aristotle's "theory of causation" is conceivable as a "theory of explanation". Aristotle considers that understanding something requires knowing its causes. According to him, but *au contraire* to most recent guidelines, causation and explanation cannot be comprehended separately [57]. This way, his theory has linked causation and explanation intrinsically.

But, the idea that explanation promises independent analysis took hold in the twentieth century. Generally, this shift has occurred due to the linguistic turn[1]in philosophy.More precisely, it was the result of philosophers of science trying to understand the nature of modern theoretical science and conciliate staunch empiricism with recent theories involving non-observably entities like atomic and subatomic particles, fields, and genes. The new scenario of modern science has made these concepts more complex. The problem is: now, in philosophy, both the words "truth" and "explanation" can be understood in two different ways,realistic or epistemic [61].

[1]The "linguistic turn" in 20th-century philosophy emphasizes the central role of language in shaping people understanding of the world and philosophical problems [58, 59]. It focuses on the analysis of language rather than metaphysical inquiry. The term was introduced by Gustav Bergmann in 1960 [60] and influenced various philosophical traditions, including phenomenology, critical theory, and pragmatism [59].

In common sense, people can perceive an explanation as a reason, as a motive for something. Asking for an explanation means asking for a reason. Or even, it is a searching for understanding, aiming to comprehend something that in the moment is obscure.

People usually agree that, explanations focus in to produce *understanding*. When one understands a phenomenon, essentially perceives how different pieces of information or elements are logically and/or semantically connected. Good explanation succeed in providing that.

Choosing an explanation among many

Another subject of explanation is the possibility of choosing an explanation, among many. Such matter is known as Inference to the Best Explanation (IBE) [62, 63]. IBE involves generating plausible explanations and selecting the one that best satisfies explanatory virtues, such as power and coherence [64].

The basic idea of IBE is that the best explanation for a particular phenomenon is the one that best accounts for the available evidence, fits our previous knowledge, and is the simplest or most elegant. Essentially, IBE involves inferring from observed evidence to formulate the best possible explanation that accounts for it. Harman, in [65], argues that IBE involves selecting the hypothesis that provides the best explanation of the available evidence. The author argues that the best explanation is the one that most effectively accounts for the evidence, has the strongest explanatory power, and is the most plausible or likely. According to Harman, IBE is applied not only in scientific inquiry but also in everyday life when one makes judgments and decisions based on available evidence. Harman illustrates this by providing examples:

> "...When a detective puts the evidence together and decides that it must have been the butler, he is reasoning that no other explanation which accounts for all the facts is plausible enough or simple enough to be accepted. When a scientist inferences the existence of atoms and subatomic particles, he is inferring the truth of an explanation for various data which he wishes to account for..." [65].

One can see that it is possible to have different explanations regarding a specific context. So, there may be options for choosing between them and some selection criteria based on people's preferences or systems users requirements.

Users' requirements for explanations acceptance

Some researchers have investigated the behavior of system users regarding to the acceptance of explanations. They have found some requirements, primarily:

- Explanations should be coherent, realistic, and feasible.
- They must align with users' mental models to enhance understanding, satisfaction, trust, and performance [66, 67].
- Explanations should support users' sensemaking by providing deeper insights beyond simple loose feature lists or salience images [67].
- User involvement in the explanation generation process, such as through feedback, can improve acceptance and understanding [68].
- Explanations should be user-centered, catering to individual needs and avoiding information flood [69].

Attempts at explanation standardization

Recent research has explored the formalization of explanations in various domains, including applications in computer science. In formal logic and philosophy context, the formalization of explanation has been characterized as formal logical derivations [70, 71]. In cognitive science, formal explanations are intrinsically linked to instances of the representation of concepts [72, 73]. In computer science, formal techniques for compositional verification provide a basis for componential explanation [1] formalization [74], while dialectical explanation support has been formalized for argument-based reasoning [75].

Formal explanations have also been investigated in relation to data models and theory models [76]. Other perspective in recent work has included *timing* component on developing formal concepts for explanation [77].It is possible to find direct application of explanation theories in AI systems, for instance, the development of an explanatory method and a Graphic User Interface (GUI) based on Achinstein's Theory [78]. Some explanation models are widely influential, like Hempel's [79] and Achinstein's [80].

Hempel's deductive-nomological model of explanation reconstructs explanations as logical arguments with laws and boundary conditions as premises [81]. Hempel's views on scientific understanding and explanation have been debated, with some arguing for consistency in his approach [82].

According to Hempel's theory [79], a good explanation must meet two main criteria: it must be logically valid and it must be supported by empirical evidence. The first criterion means that the explanation must follow logically from a set of premises, often in the form of scientific laws or general principles. The second criterion means that the explanation must be based on observations or data that support the premises.

The first criterion of Hempel's theory can be seen a way to formalize the explanation. For example:

$$(P \wedge Q) \rightarrow R$$

constituted by *explanans* :
P represents the law statement
Q represents the initial conditions and
R the *explanandum*

In this notation, the arrow ($\rightarrow$) represents logical implication, which means that if both P and Q are true, then R must also be true. The conjunction symbol ($\wedge$) represents logical conjunction, which means that both P and Q must be true for the statement $(P \wedge Q)$ to be true. Therefore, the statement $(P \wedge Q) \rightarrow R$ captures the idea that a scientific explanation must use a law-like statement and relevant initial conditions to logically entail the explanandum.

The sentences that articulate the explanatory information is known as the *explanans*, while the entire collection of sentences is referred to as the *explanantia*. The components of the explanans are the laws and explanatory theories, along with the initial conditions. The term *explanandum* refers to the fact or phenomenon to be explained.

For instance:

$$(P \wedge Q) \rightarrow R$$

where:
P represents the law of gravity

[1] In simple terms, a componential explanation describes how the properties of a system, which is made up of several components, can be understood by looking at the properties of those components and how they interact with each other [74].

Q represents the initial conditions (the apple's position and the Earth's gravitational field)
R represents the explanandum (why the apple fell from the tree).
This means that if the law of gravity and the initial conditions are true, then the explanandum (why the apple fell from the tree) must also be true.

Another formal approach to scientific explanation related to Hempel's theory is the Causal-Mechanical (CM), developed by Wesley Salmon [83]. Like Hempel's theory, the CM highlights the importance of general laws and initial conditions in scientific explanations. However, the CM establishes a greater emphasis on the causal mechanisms underlying the phenomena. Furthermore, the work suggests that explanations should focus on identifying and describing these mechanisms in detail.

Achinstein's Theory of Explanation, proposed by philosopher Peter Achinstein [80], offers a way to handle scientific explanation, focusing on the concept of explanation as a response to *why-questions* and the use of evidence in generating explanations. Here are some key elements of his theory:

- Explanation as Answering Why-Questions: explanations are not merely causal accounts but are responses to specific why-questions. Explanations aim to show why something happened or is the case, *based on evidence*.

- Difference from Causal or Unificationist Theories: Focus on evidence and understanding. An explanation must offer an account that is informative and relevant to the context of the question being asked.

- Objective vs. Subjective Explanation: distinction between potential and actual explanations. A potential explanation is something that could explain an event under certain conditions, while an actual explanation is one that really does explain it in a given context, depending on the information available and the nature of the why-question.

- *Illocutionary*[1]: the explanations are seen as speech acts related to generate understanding, clarity in someone.

- Evidence and Explanation: One of the core components of Achinstein's theory is the role of evidence. An explanation to be successful, it must be supported by relevant evidence. This contrasts with some earlier views of explanation that didn't emphasize the necessity of having evidence linked to the explanatory claim.

2.1.4 Type of explanations in AI

Human explanations are inherently complex, influenced by the cognitive biases of both the person explaining and the person receiving the explanation. It's important to consider these biases when generating explanations [84, 85]. Some researchers emphasize the importance of aligning AI explanations with human cognitive processes [84, 86].

In order to meet various requirements, operating within distinct constraints, researchers have implemented various strategies to explain models' behaviors to users. One can see some

[1] Illocutionary acts, also known as "speech acts," occur when actions are performed through speaking. In this context, spoken words are not merely sounds but are understood as actions. For example, when a teacher says, "You are dismissed," signaling the end of class, the teacher's words create a change in the situation. Similarly, when a judge announces a final decision, the act of speaking brings about a corresponding action. In an illocutionary act, words serve as vehicles for carrying out another action.

of them in the summary table below. Table 2.1 shows the main approaches in terms of method. Table 2.2 presents the main formats of presenting the explanation and Table 2.3 exposes the main GUI strategies.

Explanation Type	**Description**
Feature Importance/Attribution	Highlights key input features that influenced the decision (e.g., SHAP [87,88], LIME [89]).
Counterfactual Explanations	A counterfactual explanation reveals what should have been different in an instance to observe a diverse outcome [90].
Model Simplifications	Simplifies complex models into interpretable forms like decision trees.
Saliency Maps/Visualizations	Highlights important parts of input data (e.g., image pixels or words) that influenced the model's output.

Table 2.1: Types of Explanations in XAI Systems

Explanation Format	**Description**
Textual Explanations	Provides plain-language descriptions of the decision-making process.
Numerical Summaries	Quantifies feature importance or probabilities in numerical form.
Graphical/Visual Explanations	Uses visual aids like bar charts, partial dependence plots, and heatmaps to explain decisions.
Set of Rule-Based Explanations	Expresses decision-making logic as conditional "if-then" nest rules.

Table 2.2: Common Formats for Explanations in XAI

Among this various methods, counterfactual Explanations have gained prominence in XAI, offering insights into how predictions could change under different conditions. This is an intuitive approach fully suitable to the human cognitive processes because it uses simple concepts to help users understand an AI model decisions.

Studies have shown that counterfactual explanations elicit higher user satisfaction and trust compared to causal explanations [91]. Counterfactual Explanations figures as the focus of the present work. One can see some important concepts regarding to such an approach in the next section.

2.1.5 Counterfactual reasoning and explanation

In explanation theory, explanations clarify complex systems by providing reasons or justifications for specific outcomes, fostering user trust. Formalizing the concept of explanation allows a structured approach, ensuring coherence, logic, and consistency with observed model results. Within this context, Counterfactual Explanations have become an progressively relevant method, offering an intuitive way to understand complex decision-making processes by highlighting how slight input changes could lead to different outcomes.

Delivery Method	Description
Interactive Dashboards	Users can interact with explanations and adjust input values to see how predictions change.
Reports/Static Summaries	Generates fixed reports summarizing the model's decisions and logic.
Visual Overlays	Provides interactive visual indicators (e.g., heatmaps) on images to show areas of focus in decision-making.
Inline Explanations	Shows explanations within the user interface next to the AI's outputs (tables, labels, hints).
Dialogues/Chatbots	Allows users to query the system for explanations using conversational agents.
Multimodal Explanations	Combines textual, graphical, and numerical formats in a single user interface.

Table 2.3: Delivery Formats of Explanations in XAI (user interaction)

The origins of *counterfactual thinking* can be traced back to early philosophers such as Aristotle and Plato.These philosophers were interested in exploring the epistemological status of subjunctive suppositions (statements that express hypothetical situations). Socrates, in many Plato's Dialogues, often uses hypothetical questioning, equivalent to modern counterfactual analysis concept. By presenting "if-then" scenarios or inverting the prepositions, he encourages his interlocutors to envision different outcomes, consider various conditions, and examine the implications of their beliefs. This approach, a key aspect of the Socratic method, helps clarify concepts, verify the truth, uncover logical inconsistencies, and build trust in rational inquiry.

Jäncke and friends, in their survey work, have established that individuals employ counterfactual reasoning to examine and make sense of the past, plan courses of action, make emotional and social judgments, and guide adaptive behavior [2]. The researches proposed that counterfactual reasoning depends on three processing stages as exhibited by Figure 2.3.

Counterfactual thinking involves mentally simulating alternatives to actual past events. Much of human knowledge relies on this type of subjunctive reasoning and "contrary-to-fact" conditional statements, rooted in our ability to consider hypothetical events that *could* occur. For instance, it includes reasoning about what *might* have happened, what *would have happened* if certain conditions were met, or what tendencies, faculties, or potentialities an object *could* manifest in suitable environments [92]. In other words, counterfactual thinking is a process of mentally generating alternatives to facts, speculating a better or worse counterpart of one's current situation [93].

It can be describe as a "If *Antecedent* then *Consequent*" structure, where the *Antecedent* is *factually false* [94]. It deals with the elucidation of events that have not occurred. Such an approach involves the conduct of mental experiments which enable the recreation of hypothetical versions of historical events, divergent from the actual (factual) sequence of occurrences.

The primary objective of such a cognitive exercise is to facilitate the testing of past hypotheses against the extant evidence [95]. The representations and cognitive processes underlying the imagination of alternatives to reality (facts) are similar to those under rational thought, including reasoning from counterfactual conditionals [96].

Counterfactual reasoning can result in behavioral and affective consequences supporting

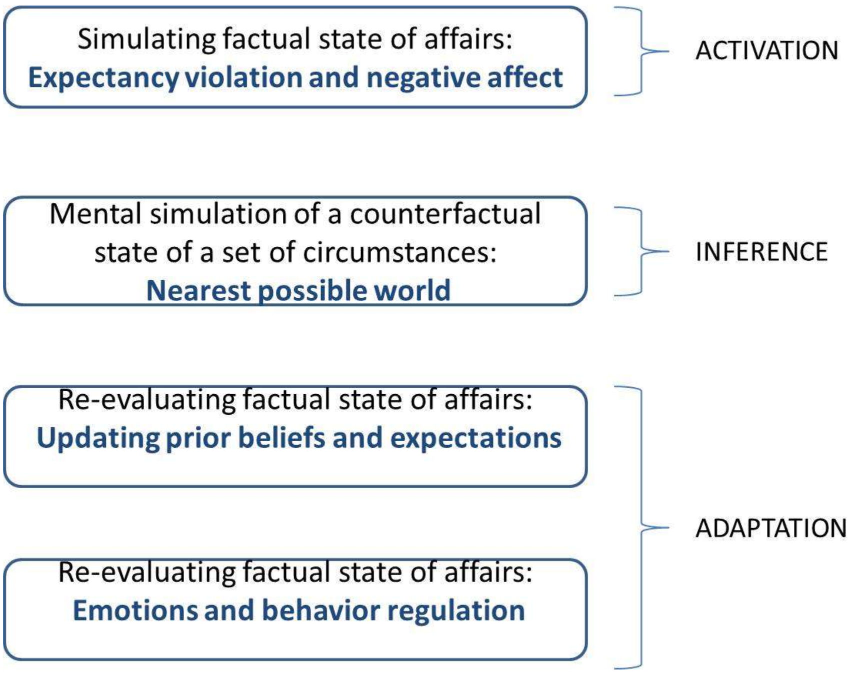


Figure 2.3: Main steps of the counterfactual reasoning process. Figure originally published in [2]

adaptive behavior [2]. It can promotes relational, analytical or creative process style [97]. Supports learning from past experience [98]. Supports future actions plan and predictions [99]. It generate emotions like guilt, regret, blame, and relief are important for managing social behavior. These feelings help people evaluate and control their interactions with others. By thinking about 'what could have been,' individuals learn from past experiences and adjust their future actions to improve or confirm their current behavior [100].

In mathematics, especially in the realm of causal inference, counterfactual reasoning is formalized through methods that enable assessment of hypothetical interventions or modifications in the variables of functions. Structural equation models, potential outcomes, and related tools offer a mathematical language to express and quantify "what-if" scenarios. This allows researchers to derive counterfactual probabilities and effect estimates with clarity and consistency. By providing a reliable computational infrastructure, these methods instill trust in the soundness of mathematical inference and its applications in empirical research. One can cite as examples the Judea Pearl's framework [101, 102] and the Rubin Causal Model (also called the Potential Outcomes framework [103, 103]) that are two influential conceptual and mathematical frameworks used for reasoning about cause-and-effect relationships and can be used for formalizing counterfactual reasoning.

There is a way of formalizing Counterfactual Reasoning via Modal Logic [104]. Modal Logic provides a formal system for reasoning about possible worlds, which is also relevant to counterfactual thinking [105]. For instance, let's consider the statement:

"*If it rains, the streets will be wet.*"

One can express this statement in modal logic as follows:

If it rains, then necessarily the streets will be wet.

Let P = "*it rains*" and Q = "*the streets will be wet*". So, one can express the statement in symbolic form as:

$P \to \Box Q$

The symbol "$\to$" represents the conditional "if...then" and "$\Box$" represents the modal operator "*necessarily*."

This way, the statement "$P \to \Box Q$" can be interpreted as

"If P is true, then Q is necessarily true in all possible worlds."

We can also express counterfactual ideas using modal logic. For example, considering the counterfactual statement:

"If I had studied harder, I would have passed the exam."

We can express this statement in modal logic as follows:

If I had studied harder, then in other distinct scenario (a possible world where I studied harder), I passed the exam.

Setting P = *"I studied harder"* and Q = "I passed the exam", so, one can express the statement in symbolic form as:

$P \to \Diamond Q$

The symbol $\Diamond$ represents the modal operator "possibly." The statement "$P \to \Diamond Q$" can be read as

"If P is true, then Q is "possibly" true in at least one possible world."

In this example, the modal operator $\Diamond$ is used to express the idea that if I had studied harder, it is possible that I would have passed the exam in at least one possible world (other hypothetical scenario, counterfactual scenario), even if I did not in our actual world (factual scenario).

In the context of artificial intelligence, counterfactual explanations of model confidence can improve user understanding and trust in AI systems [106]. The next chapter addresses counterfactual analysis applied to the explainability of AI models.

2.1.6 Conceptual remarks

Some researchers highlight that are a conceptual distinction between the terms Counterfactuals and hypothetical. Both are thoughts about possibilities, however, they differ in *when* a change is imagined to take place. Hypothetical are thoughts about changes that lie in the future, while Counterfactuals are thoughts about changes that lie in the past [107]. But, the present work does not make such conceptual distinction.

There are studies from Psychology of Emotions area investigating the benefits and harms that certain effects of counterfactual thinking can bring to individuals [93, 100, 108]. One primary perspective in this field suggests that counterfactual thinking can be a tool for emotional regulation. By imagining alternative outcomes, people can reduce the intensity of negative emotions associated with a particular event or situation. The psychology of emotions' approach to counterfactual thinking suggests that this process is relevant to emotion regulation, decision-making, personal learning and motivation [109–113].

Chapter 3

Counterfactual explanations applied to AI

Considering the XAI perspective, counterfactual explanations (CE) focus on investigating and identifying differences between some actual (real, factual) situation and a distinct one that could have happened if one or more input features had been different. This approach can help understand which features were most important in the model's decision-making process and how different combinations of inputs can generate different outputs.CE allows to explore how specific input features affected the model's output and contrariwise, given a certain output, verifying what is the necessary input adjustment to obtain it.

Distinct aspects of such an investigation can be found on the literature. Hendrickson establish that CE is about evaluating conditional claims regarding to alternative realities (possibilities) and their correspondent consequences [114]. Crupi, highlights CE can be used to provide a set of input features that need to be changed in order to achieve any desired outcome [115]. Watcher and colleagues enunciate three aims for the analysis: to inform and help the subject understand why a particular decision (output) was reached; to provide grounds to contest adverse decisions (justifications about the factual scenario); and to understand what could be changed (counterfactual scenario) to receive a desired result in the future, based on the current decision-making model [116].

In addition, counterfactual investigation can help in some situations where it is necessary to understand an important phenomena that is very rare. Uncommon events can prove difficult to analyze systematically because they do not generate many sampling observations. In such situations counterfactual analysis can be used to make small sample sizes feasible to be studied scientifically [117]. Indeed, generation of counterfactual samples (counterfactual outputs) for data augmentation purposes consist in a wide investigation field [118–121].

Counterfactual data augmentation (CDA), adding minimally perturbed inputs during training, helps reduce model reliance on spurious correlations and improves generalization to out-of-distribution (OOD) data [122]. For image classifiers, for instance, augmenting a training dataset, using counterfactual examples, has been empirically shown to break spurious correlations [123].

Another highlighted point is that the generated counterfactual examples can be used to provide explanation of model behavior as well as studying the regions in the input space where the model performance deteriorates [124].

Judea brings that counterfactual is one of the most fundamental concepts in theories of causation [125]. Counterfactuals are often posited to indicate a causal connection between an actual event and its imaginary counterpart [126].

Miller point a psychological/social perspective of counterfactual investigation: people sometimes do not ask "why event P happened", but rather "why event P happened instead of some event Q" ? Or even "why $not - Q$?". This has important social and computational consequences for explainable AI [127].

In real world situation, someone can ask: “What do I need to do for my loan to be accepted?” In this case, the user can be told: “if you had the attribute A1 equal to X instead Y you would get the loan" [128]. Using textual explanation it could be: "increase your down payment by 1000 and limit credit card debt to a maximum of 5000 for the next two months" [129]. Such a subject is part of the CE analysis scope [130].

One can find real applications of CE in real activities, for instance: credit applications [131], employability [132], visual explanations in images classification [133], economics [134], medical images classification [135] and also in biological systems modeling [136, 137].

Some concerns are reported in counterfactual applications, mainly regarding data protection regulations, data leaking risks, malicious reverse engineering exploring system vulnerabilities and hypersensitiveness of the models [138]. Despite these concerns, counterfactual explanations remain valuable for enhancing decision support, managing controlled changes, and uncovering new insights [132].

3.1 Methods of Counterfactual Examples Generation

It is possible to find researches on counterfactual explanations dealing with mixed data types through integer programming [139]. The researchers shows how a set of constraints associated to an integer programming solver can be used to find coherent counterfactual explanations. Their proposed method was validated by using FICO explainability dataset [140].

One can see a model agnostic method for finding interpretable counterfactual explanations of classifier predictions by using *class prototypes* [141]. The authors show that encoders or k-d trees can be used to generate class prototypes. These class prototypes can speed up the search for counterfactual instances and result in more interpretable explanations. The proposed method is validated on an image (MNIST) and tabular (Breast Cancer Wisconsin Diagnostic) datasets.

Some works aims to investigate the feature-highlighting explanations and its dependence on several key assumptions that are often overlooked. These assumptions include that suggested changes in feature values translate clearly into real-world actions, that features can be fairly compared by examining only their distribution in the training data, that each feature is relevant only to the specific decision in question, and that the model remains stable over time, follows predictable patterns, and produces binary outcomes [142].

One can find a wide variety of methods involving minimization of an objective function as a mean of solving the generation of counterfactual examples. The objective function can weight distinct terms and samples attributes. For instance, the squared difference between the predicted and the desired outcomes on a perturbed instance scaled by a L1 norm [116].

Clustering-based approaches can also be found [143]. The authors introduced an algorithm called Transitive Global Translations (TGT), for computing Global Counterfactual Explanation (GCE). The strategy involves obtaining a low-dimensional representation of the data, identifying clusters of points within that representation, and then analyzing the dissimilarities between the clusters to infer their underlying meanings. The researches demonstrate that TGT utilizes compressed sensing to detect dissimilarities between each pair of groups, while ensuring that these dissimilarities remain consistent across all groups, make possible identify key features. The solutions from compressed sensing are only able to explain the differences between one pair of groups. As a result, the authors generalize the compressed sensing solution to find a set of consistent explanations among all groups simultaneously. TGT method is assessed on UCI Iris, Boston Housing, and Heart Disease datasets.

Some researches show that counterfactual explanations can be made more robust by using a wider range of plausible counterfactuals instead only the closest counterfactuals [144]. One application of such an approach is generating realistic examples from users actions (user inputs disturbing) by evaluating its reality on the empirical data statistic distribution [145]. To suggest a feasible users actions, the authors define a cost function based on the Mahalanobis' distance [146] and the Local Outlier Factor (LOF) [147]. The proposed method was assessed on FICO [140] and German [148] datasets.

There are works that investigate the intersection between uncertainty estimation and interpretability. For instance, in [149] the authors introduce a method to help explain uncertainties of differentiable probabilistic model by perturbing input features. The method called Counterfactual Latent Uncertainty Explanations (CLUE), indicates how to change an input, while keeping it on the data manifold, such that a model becomes more confident about the input's prediction. Five dataset were employed in the experiments. Four tabular (LSAT, COMPAS, Wine, Credit) and one composed of images (MNIST).

The vulnerabilities of counterfactual explanations was studied in the work [138]. The authors show how counter factual explanations can be manipulated from specif perturbation in the input space. They also demonstrates how counterfactual generation may converge to drastically different samples (examples, observations) even under a modest perturbation. They describe how some models can unfairly provide low-cost recourse for specific subgroups in the data while appearing fair to auditors. Such experiments use Communities and Crime and the German Credit datasets [148].

Some researches brings explanation framework based on adversarial generated counterfactual examples that can handle tabular data, including categorical and discrete variables. Among them one can see in [124] an algorithm that uses a gradient-free optimization technique based on genetic algorithms to adjust adversarial approach to tabular data. The proposed framework is model agnostic and it is specific for classification tasks. It uses adversarial concepts to promote perturbations and generate new examples. The method was validated on German Credit Data [150]

In the research [151] the authors describe a method to learn a consistent estimator of a continuous optimal Transport Map [1] from two empirical distributions.Optimal transport maps define an one-to-one correspondence between probability distributions, however, these maps are generally defined on empirical observations and cannot be generalized to new counterfactual samples. The propose method in [151] enables to extend the transport plan concept to new observations without computing again the discrete optimal transport map. The method provides statistical guarantees to counterfactual explanations, once they were originated from the optimal transport-based.

Many investigations assumed that each user has some limited ability to change (a subset of) their features in the input vector, and the goal of the users is to use this limited capacity to influence the resulting decision of the model. In order to formalize this condition, the researchers in [154] provides a general definition of *recourse sensitivity*, which needs to be instantiated with an utility function that describes which changes are relevant to the user. The authors provides counterfactual examples adopting different attribution methods: LIME [89], SHAP [155], Integrated Gradients [156] and SmoothGrad [157]. Experiments used 53 gray scale figures that were created from the User Icon picture application.

[1] An optimal transport map transforms one probability distribution into another, while minimizing a given cost defined over their respective supports [152, 153]

Other perspective commented in the bibliography involves data causal structure. The authors in [158] approach this matter demonstrating how to instantiate counterfactual examples considering causal structure in the data. They adopt Judea Pearl's method [159] for computing counterfactuals. The idea is first produce a structural causal model (SCM) and associated graph that describe the true data generating process. The authors shows that such approach allows to compute counterfactuals using the true causal relationships between the data, whether it be between input features or between input and output features.

Another frequent subject regarding counterfactual examples generation is the diversity of the generated sample. A typical example is the framework called *Dice* [160]. The researches have proposed a method to capture the samples diversity by building on determinantal point processes (DPP) [1]. The DiCE framework can generate counterfactuals examples that are diverse and well approximate local decision boundaries. They develop an optimization function include three main terms: distance between the counterfactual and the desired outcome; the distance between the factual and the counterfactual; and a diversity metric. The three terms has weight factors to balance the effect of each part. Experiments involves four real-world datasets (Adult-Income,LendingClub,German-Credit,COMPAS). The authors provide an implementation of the framework at *https://github.com/microsoft/DiCE.*

The diversity of generated samples, coupled with the hypersensitivity of the model to some small input changes, can lead to significant output variations, resulting in potentially misleading counterfactual explanations. Addressing such a matter, the work [162] introduce a method to detect unknown classes while still generalizing the new instances of existing classes. The authors propose a dataset augmentation technique that the authors called Counterfactual Image Generation. It is based on generative adversarial networks. The idea is generating new examples that are close to training set examples yet do not belong to any training category.

The research described in [163] proposes a framework called CausaLM for producing causal model explanations using counterfactual language representation models. The approach involves fine-tuning of deep contextualized embedding models with auxiliary adversarial tasks derived from the causal graph of the problem. The main contribution relays on the intersection of natural language processing, deep models interpretations and causal inference. To test their method, it was created novel data sets specifically to the experiments.

Many works point similarities between counterfactual and adversarial examples generation. On can see in [164] a study investigating counterfactual explanations under algorithmic recourse and adversarial perspective. The authors investigates counterfactual explanations through the lens of adversarial examples subjects. In terms of problem-solving, counterfactual and adversarial examples share a common concept. They both involve finding a modified input that satisfies specific constraints about models outputs. The main difference is that the aim of adversarial attacks is to create adversarial examples, inputs that are nearly indistinguishable from natural data but are incorrectly classified by the machine learning model [165]. Practical experiments described in the paper [164] involve UCI Adult, COMPAS and German Credit datasets [148].

It is possible to find researches addressing counterfactuals applied to Natural Language processing [166]. The authors have described a framework that focus on generating contrastive explanations with counterfactual examples in Natural Language Inference (NLI). The method presented in their work is called Knowledge-Aware Contrastive Explanation generation framework (KACE). Initially, the framework identifies key phrases (or rationales) from input

[1] In contrast to traditional structured models like Markov random fields, which become intractable and hard to approximate in the presence of negative correlations, determinantal process points offer efficient and exact algorithms for sampling, marginalization, conditioning, and other inference tasks [161]

sentences and utilizes them as key perturbations to generate counterfactual examples. Once the framework obtains qualified counterfactual examples, it combines them with original examples and external knowledge to feed into a knowledge-aware generative pre-trained language model, which ultimately generates the contrastive explanations. The method adopts a three-part (valid, distance and diversity) loss function to select qualified counterfactual examples. The experimentation involves The SNLI [167] and IMDB datasets [168].

On can find in [169] the description of a general-purpose counterfactual generator that allows for control over perturbation types and locations, trained by fine tuning GPT-2 on multiple datasets of paired sentences. The method called Polyjuice is able to automatically produce diverse sets of realistic counterfactuals examples. It allows users controlling over the kinds and locations of perturbations. Polyjuice framework is open source, and it is available at [170].

Data Augmentation is also a subject involved in counterfactual examples generation problem [122]. It is possible to find retrieval-augmented generation framework for creating diverse counterfactual perturbations, such as the method called Counterfactual Data Augmentation (CDA) [122].

Besides tabular data, many researches involve complex data types. For example, the authors in [171] investigate generating counterfactual examples in a text-to-SQL scenario. The task of text-to-SQL is to convert a natural language question to its corresponding SQL query in the context of relational tables. In addition, the author describes a weakly supervised method called DTE (Detecting-Then-Explaining) for error detection, localization, and explanation. Tests involving NOISYSP, WikiSQL and WTQ.

The researchers in [172] introduce a framework called Datasets Interface, a framework fated to synthesize counterfactual examples to exhibit a specified distribution shift. Distribution shifts are one of the major source of failure of deployed machine learning models. Practical experiments involves ImageNet dataset.

The actions sequence is also investigate in the scope of counterfactual generation. Optimal sequence to generate new examples is discussed [129]. The work deals with the counterfactual and contrastive examples generation problem through the lens of program synthesis. The main goal is to *synthesize an optimal sequence of actions* that, if an individual executes successfully, can change the model classification. The method combines search-based program synthesis and test-time adversarial attacks to construct sequences of actions over a domain specific. The technique for synthesizing action sequences is based on the optimization procedure. The objective function models the cost of each action and business constraints.

Evolutionary approach is also commented in some works [173]. The authors have described an evolutionary approach to, given a set of prototype examples representing a set of categories, a similarity function, and a new example classified in some other category, find the cost-minimizing changes to the attribute values such that the case is reclassified as a member of a (different) preferred category adopting an evolutionary approach.

Some researches brings particular heuristic to find or generate counterfactual examples. The work [174] proposes finding counterfactuals examples through a geometric heuristic search procedure. The idea is to carry out the search by growing spheres around certain instances determined by the heuristics criterion.

A method called Constrained Adversarial Examples (CADEX) is presented [128]. CADEX uses adversarial examples to provide understanding about the model by finding counterfactual explanations close to the original input. CADEX searches for examples obeying constrains regarding to the business domain. The main objective is find the counterfactual example that is as close as possible to the factual while satisfying a number of business constraints. The method adopt adversarial concepts by following the gradient in input space using optimizers. The researches show that this approach typically results in an input that lies right on the decision boundary between the classes.The method was evaluated on German Credit dataset [148]. CADEX framework is open-source, and it is available at [175].

The method named FACE (Feasible and Actionable Counterfactual Explanations) is presented in the research [130]. The work approaches counterfactual examples generation by providing actionable and feasible paths to transform a selected instance into another one that meets certain goals. FACE method employs probability density-weighted metrics associated with distances routes to identify "feasible paths." The idea is conciliate dense populated and short trajectories from the factual example until its counterpart. Such routes can be achieved and customized to suit specific problems. The authors demonstrate that, by using FACE approach, it is possible to generate counterfactual examples consistent with the input data distribution. The algorithm was validated on the MNIST dataset.

A Benchmark Study for Counterfactual Generating Methods can be found in the work [176]. The researchers contributed executing a Benchmark Study for ten Counterfactual Generating Methods on Tabular Data. The authors proposed to evaluate the explanations and the generating process by using a set of proposed metrics.

For more works regarding counterfactual examples generation one can check [177]. The survey seek to review and categorize researches on counterfactual explanations, a specific class of explanation that provides a link between what could have happened had input to a model been changed in a particular way.

3.1.1 General remarks

Current approaches rarely take into account the feasibility of actions needed to achieve the proposed explanations, and in particular they fall short of considering the causal impact of such actions [115].

The two most common approaches for generate counterfactual examples is by using convex optimization methodologies or custom heuristic rules [176].

A polynomial time algorithm can be used to find the optimal counter factual explanation on every possible realization of the counterfactual environment dynamics [178].

Counterfactual explanations can naturally be model-agnostic, as they are suitable to explain the output of any opaque approach in post-hoc [126, 179].

Users find explanations involving categorical features easier to understand than those involving continuous features. Additionally, counterfactual explanations lead to higher prediction accuracy compared to no-explanation controls, but they do not surpass the accuracy of causal explanations. While this isn't directly about the benefits or limitations, it may be relevant to the research question [180].

It is important to note that some researchers tend to either join or intentionally distinguish *contrastive reasoning* from *counterfactual reasoning* despite their conceptual similarity [126].

Contrastive focus on why one outcome occurred *instead of another*. It is a contrasts-based comparison that typically says why a particular decision or prediction was made as opposed to another possible outcome. *Counterfactual* explanations focus on how the outcome *would have been different* if certain values of the input had changed.

3.1.2 Rating Criteria for The Counterfactual Examples

It is possible to use some criteria to rate the counterfactual examples. The most common are:

- Accuracy: refers to how well these examples achieve the intended outcome (target class and constraints).
- Feasibility (Valid): This measures whether the counterfactual example is possible in the real world, given the problem's constraints. For example, in a health care study, a generated counterfactual example that involves reducing the patient's age would not be considered feasible. Alternatively, some situations that break some physical laws or overpass the devices' nominal capacity in engineering applications may not be feasible.
- Plausibility: This feature assesses the degree to which the counterfactual example is believable given a specific context. It tells how much the new example is credible or likely to occur (probability distribution) based on data and domain knowledge.
- Coverage (diversity): This measures the diversity of the generated counterfactual examples.
- Actionability: It refers to whether the changes suggested by the counterfactual are actionable by the individual or system, meaning that the user can take concrete steps to realize the proposed changes.
- Proximity: It refers to the distance between the factual example and its proposed counterfactual.
- Sparsity: It reflects the number of features that were changed in the factual example to generate the counterfactual.

Three of these criteria may be require a better distinction. In few words: *Feasibility* is about whether a change is possible under practical constraints. *Plausibility* is about whether the change is credible considering the data' statistical behavior. *Actionability* is about whether the change(s) can be performed by the user.

Such criteria can be used to construct a loss function with additional factors associated to features' weights: $MyLossFunc = \theta_1 * Loss_{dist} + \theta_2 * Loss_{valid} + \theta_3 * Loss_{covegare}$. This makes it possible to create metrics to quantify the quality of the counterfactual generated and assist in their selection process, for instance, trying to find dominating counterfactuals using a Pareto's frontier.

3.2 Towards a Geometric Paradigm in Counterfactual Reasoning

Although conceptualizing counterfactuals as transformations within a geometrically defined input space is both elegant and intuitively appealing, a systematic theoretical foundation and empirical validation for such approaches are still largely unexplored. Addressing this gap could enhance the theoretical landscape of counterfactual analysis and drive the development of innovative tools and computational techniques. These advancements could enable more nuanced and flexible explorations on modeling counterfactual reasoning process, ultimately opening new avenues for interdisciplinary research and applications. This work aims to address this gap by investigating the use of geometric data structures to model counterfactual analysis for interpreting AI models. Therefore, the next chapter will present the theoretical foundations of the geometric concepts used in the proposed methods, helping the reader better understand how these methods work.

Chapter 4

Geometric Structures

This chapter presents the theoretical foundations supporting the development of the proposed methods for example generation and counterfactual analysis. A thorough understanding of the fundamental concepts related to the data structures used in these methods is crucial for grasping the proposed methodology. The chapter also describes some computational experiments performed to observe details of the theory content.

These computational experiments aimed to better understand the theoretical and geometric nuances associated with distinct distance metrics and their behavior across varying dimensionalities. They provide a practical framework to validate and refine theoretical constructs, offering insights into how geometry properties adapt to different metric definitions and dimensions. By simulating and analyzing scenarios that are otherwise difficult to observe analytically, computational experiments bridge the gap between abstract mathematical theory and its empirical manifestations, fostering a more comprehensive grasp of geometric behavior and its implications in the proposed method of counterfactual examples generation.

4.1 Fundamental Concepts

4.1.1 Point

A point is that which has no part [181]. It has no parts because it can not be divided into smaller components. It has no sizes, no extension: without length, without width, height, area or volume. A point does not occupy any space, it is dimensionless. It's purely a position in space.

Although a point takes up no space, it represents a specific position in space. It signifies a precise location in a given spacial context. In geometry, a point's location is always understood relative to a reference system, typically involving a reference point known as the origin. The origin works as the starting point from which all other positions (points) are measured. Figure 4.1 shows a point p located in the position 3 from the origin point o in one-dimensional space. In this one-dimensional context, the position of the point p consists in the displacement over a

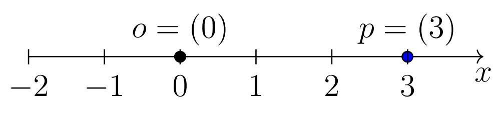


Figure 4.1: A point p located in the position 3 from the origin point o in one-dimensional space.

line. Its position can vary along the line length.

Similarly, in another context, in a two-dimensional space now, the position of the point p is indicated by adopting two values (coordinates), one value to indicate the displacement in each of the two dimension x and y. The Figure 4.2 presents such a context. In this two-dimensional

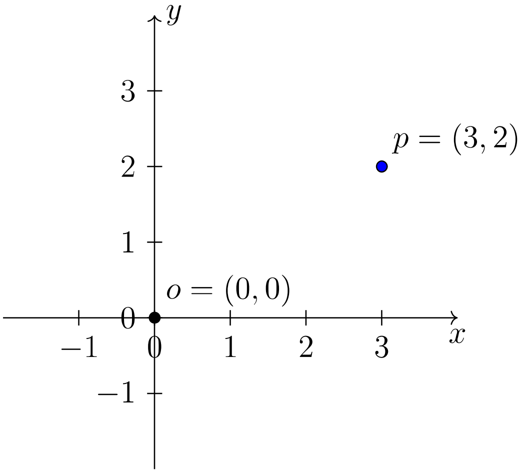


Figure 4.2: A point p located in the position $(3, 2)$ from the origin point o in a two-dimensional space.

context, the position of a given point p consists in the displacement over an area. Its position can vary inside the plane's area.

In one more context, in a three-dimensional space, the position of the point p is indicated by adopting three values, one value to indicate the displacement in each of the three dimension. For instance, x, y and z.

In such a three-dimensional context, the position of a given point p consists in the displacement within a volume. Its position can vary into the volume.

4.1.2 Dimension

Dimension can be understood as each of the extensions an object can have. The number of spatial dimensions consists of the degrees of freedom an object has to move in space. It refers to the types of movement it is possible to do in the space. For instance, in a three-dimensional space we have three mutually perpendicular types of motion (left/right, forward /backward, up/down), and any point in such a space can be reached by combining the three possible types of motion [182].

To better comprehend the concept, it is useful to begin by considering a point as the initial element. A point has neither extension nor size. When the point is continuously displaced in a single direction, tracing a linear path, a straight line is formed, representing an extension into

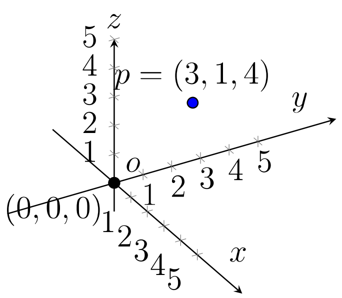


Figure 4.3: A point p located in the position $(3, 1, 4)$ from the origin point o in a three-dimensional space.

one dimension (Figure 4.4). The resulting line contains an infinite number of points and exists solely in one dimension, characterized by a single extension. As Euclid described, "*a line is length without width*" [181]. To indicate the position of a given point p along the length of the

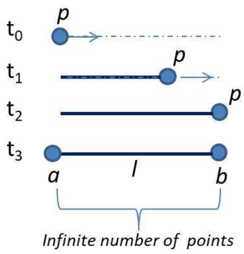


Figure 4.4: Straight displacement of a point creating a line l.

line, a single value is sufficient, for example $p = (3)$. Thus, it is understood that the point p is in position 3 from a given origin.

On its turn, this line l moving in a direction at right angles (perpendicular) to itself for a distance equal to its length, generates a square s , a two-dimensional (two extensions) element, as one can see in Figure 4.5. The produced square contains an infinite number of lines and is bounded by 4 lines and 4 points [183].

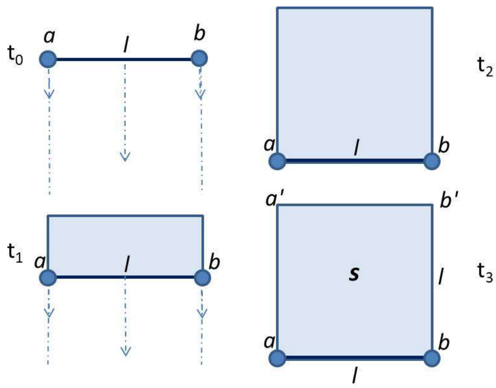


Figure 4.5: Line orthogonal displacement generating a square s.

To indicate the position of a given point along the area of the square, a single value is not sufficient anymore. It is necessary to indicate two values, one for each dimension. For example $p = (3, 2)$. Thus, it is understood that point p lays in the position $(3, 2)$ from a given origin. This pair of values is called coordinates.

This time, by moving the square s perpendicularly to its own plane, over a distance equivalent to its side length, a three-dimensional solid, specifically a cube, is formed (Figure 4.6). The produced cube has an infinite number of continuous planes, it is bounded by six distinct planes (faces), twelve edges, and eight points (vertices) [183]. To indicate the position of a given point into the volume of the cube will be necessary 3 values. For example $p = (3, 2, 4)$. Thus, it is understood that point p lays in the position $(3, 2, 4)$ from a given origin.

Applying a similar procedure, if the cube is dislocated in a direction that is simultaneously perpendicular to all its three spatial dimensions, over a distance equal to the length of one of its edges, a four-dimensional object known as a tesseract, or 4-dimensional hypercube, is generated (Figure 4.7). The tesseract consists of an infinite number of cubes and is bounded by eight distinct cubic cells, twenty-four square faces, thirty-two edges, and sixteen vertices [183].

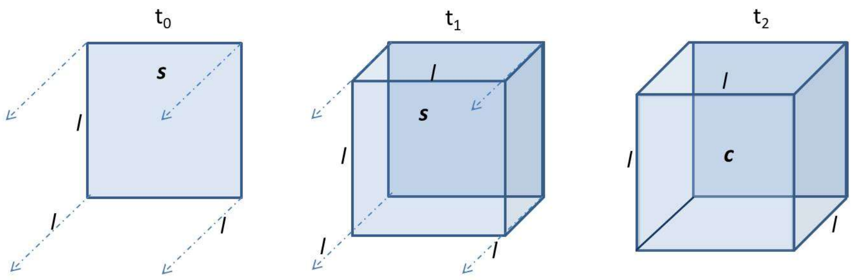


Figure 4.6: Orthogonal displacement of a square s generating a cube c.

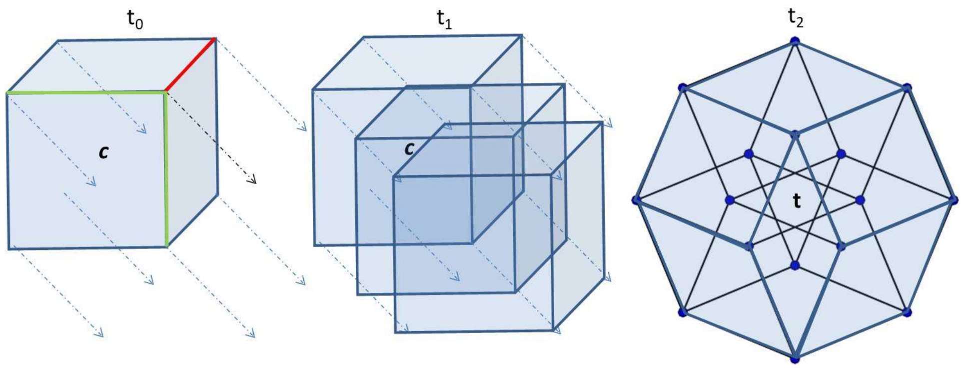


Figure 4.7: Displacement of a cube c in the representative direction of the 4th dimension generation a tesseract t.

The Figure 4.8 shows the tesseract in different perspectives. The Figure 4.8a is taken from the 1977 book, *Geometry, Relativity and the Fourth Dimension*, by Rudolf Rucker [182] while 4.8 was taken from A *Primer of Higher Space*, from 1913, by Claude Bragdon [183]. To

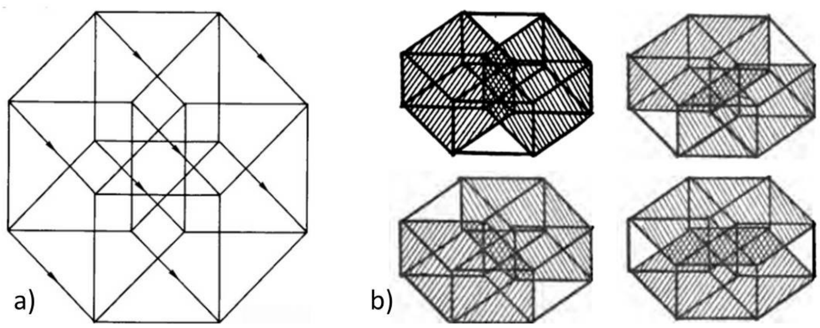


Figure 4.8: 2-D representation of a tesseract in different perspectives. c.

indicate the position of a given point into the hypervolume of the tesseract will be necessary 4 values. One value corresponding to each dimension (extension). For example $p = (3, 2, 4, 1)$. Thus, it is understood that point p lays in the position $(3, 2, 4, 1)$ from a given origin.

It is important to remark that a three-dimensional cube cannot be accurately depicted on a two-dimensional plane. As illustrated in Figure 4.7, the third dimension is typically represented by a diagonal line (in red), as opposed to a line perpendicular to the horizontal and vertical dimensions (in green), to create a notion of depth using perspective. Although the fourth dimension remains theoretical, it can represented by a direction on the plane that is perpendicular (in black) to the diagonal line used for the third dimension. By adopting this approach, the previous series can be extended by continuously dislocating the 3-D cube for one edge length in the direction of the fourth dimension, generating a tesseract [182]

4.1.3 Distance

The concept of distance is basic to human experience. In everyday life it usually means some degree of closeness of two physical objects or ideas, for instance, length, time interval, gap, rank difference, coolness or remoteness [184].From the mathematical point of view, distance measure is defined as a quantitative tool to measure how far apart two vectors are from each other [185].

In this sense distance is correlated with the idea of dissimilarity, difference, distinction, farness. Similarity, contrariwise, indicates how alike two objects are. The relationship between distance and similarity is inversely proportional: as the similarity decrease, the distance tends to increase and vice versa. In this work, the term distance is used as an abstraction of how to quantify the separation between two points in a given n-dimensional space. Distance is taken here as a *metric* to indicate the degree of separation of two distinct points.

The main requirement to perform a metric calculation in a specific problem is to obtain an appropriate distance /similarity function [186]. The performance of many algorithms depends upon selecting a good distance function over input data set [187].

To understand the abstraction of a distance function, one can start from a fundamental mathematical operation, the subtraction (also referred to as the difference operation) that introduces the basic concept of distinguishing between two elements. One can initiate by defining the distance between a and b as: $D(a, b) = a - b$. Going one step ahead, one can use the module, obtaining only positive magnitudes: $D(a, b) = |a - b|$. So, according to the current function, one can say that: $D(a, b) = D(b, a)$.Figure 4.9 shows the difference operation between two points in 1-D space scenario.

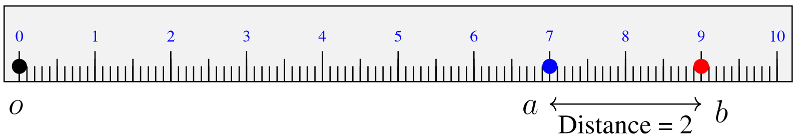
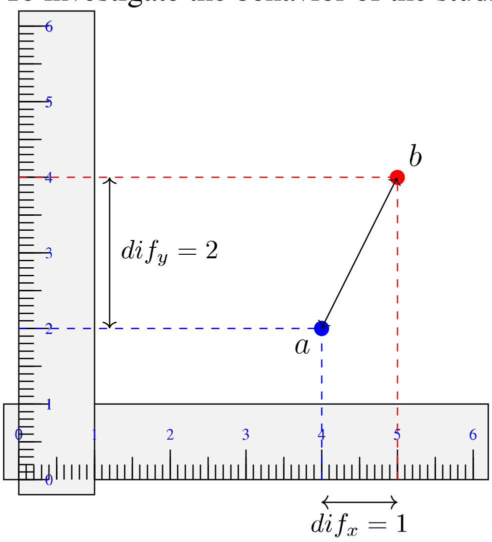


Figure 4.9: 1-D space distance calculation via subtraction operation. Origin(o) = 0, point a = 7, point a = 9 and distance $D(a, b) = 2$.

To investigate the behavior of the studied metric, an additional dimension is incorporated.



Notice that in the context of the two dimensions, we observe two differences, one for each axis (or dimension). Along the x-axis, the difference is 1, and along the y-axis, the difference is

2. When working with multiple dimensions, some questions regarding the distance calculation may arise:

- Should the total distance (or difference) be the sum of the individual differences in each dimension d: $D(a,b) = \sum_{\boldsymbol{i=1}}^{\boldsymbol{d}} |a-b|$? Or would it be more appropriate to say "the distance is the length of the segment $\overline{ab}$" and calculate it using the Pythagorean theorem $(d(a,b) = \sqrt{\sum_{i=1}^{d}(a_i - b_i)^{\mathbf{2}}}$?
- Will The differences in each dimension contribute similarly to the total, or should the metric adopt some weighting strategy:$d(a,b) = \sqrt{\sum_{i=1}^{d} \boldsymbol{w_i}(a_i - q_b)^2}$? Here w_i represents the intensity, the weight of the contribution of each dimension individually.
- It is known that the exponent applied to the difference $(a-b)^{\mathbf{2}}$ amplifies the value of larger differences, giving them more relevance in the distance calculation. Higher values of the exponent can modify the metrics sensitivity to the higher differences.

Over time, various distance metrics have been developed to address specific challenges inherent to different types of data and their spatial behavior. Table 4.1 summarizes the main distance metrics commented in the recent literature.

Distance Metric	**Formula**	**Behavior and Main Properties**	**Suitable Applications**
Euclidean	$$\sqrt{\sum_{i=1}^{d}(a_i - b_i)^2}$$	Measures the straight-line (L2 norm) distance between two points. Sensitive to magnitude and units of measurements. Assumes the space is isotropic (uniform in all directions).	Common in clustering (e.g., K-means), regression, and general distance-based machine learning models. Best for continuous, real-valued data.
Manhattan	$$\sum_{i=1}^{d} \lvert a_i - b_i \rvert$$	Also known as L1 norm or taxicab distance. Measures distance by summing absolute differences. More robust to outliers and works well with high-dimensional data.	Useful in applications where movement is restricted to grid-like paths, such as path-finding algorithms and some optimization problems.

Mahalanobis	$\sqrt{(a-b)^T S^{-1}(a-b)}$	Accounts for correlations among variables. Scales distances according to the covariance matrix S. Invariant to linear transformations and sensitive to statistical properties of data. Indicates how similar a point is to the dataset distribution.	Best applied in multivariate data analysis, anomaly detection, and situations with correlated variables.
Cosine	$1 - \frac{a \cdot b}{\|a\|\|b\|}$	Measures the cosine of the angle between two vectors, focusing on direction rather than magnitude. Ranges from 0 (identical) to 2 (opposite direction). Sensitive to orientation but not scale.	Common in text analysis, natural language processing, and scenarios where vector orientation is more important than magnitude (e.g., document similarity).
Minkowski	$\left(\sum_{i=1}^{d} \lvert a_i - b_i \rvert^p\right)^{\frac{1}{p}}$	A generalization of Euclidean (p=2) and Manhattan (p=1) distances. The parameter p determines sensitivity to large differences in any dimension. For $p \to \infty$, it becomes Chebyshev distance.	Used in machine learning tasks that require flexible distance measures based on data characteristics. Allows adjusting sensitivity through parameter p.
Chebyshev	$\max_i \lvert a_i - b_i \rvert$	Measures the maximum absolute difference along any coordinate dimension. Captures the largest distance along one axis and is unaffected by smaller differences in other dimensions.	Used in chessboard distance problems, games, or when considering maximum variance across dimensions.

Table 4.1: Summary of Main Distance Metrics

To illustrate the behavior of various distance metrics, contour graphs with gradients were generated over a 100x100 points grid. The display uses colors from the visible light spectrum, from red, which indicates closer distances (regions of more similar points), to violet, indicating greater distances (more dissimilar points). Figure 4.10 and Figure 4.11 shows the gradients corresponding to different Minkowski p values.

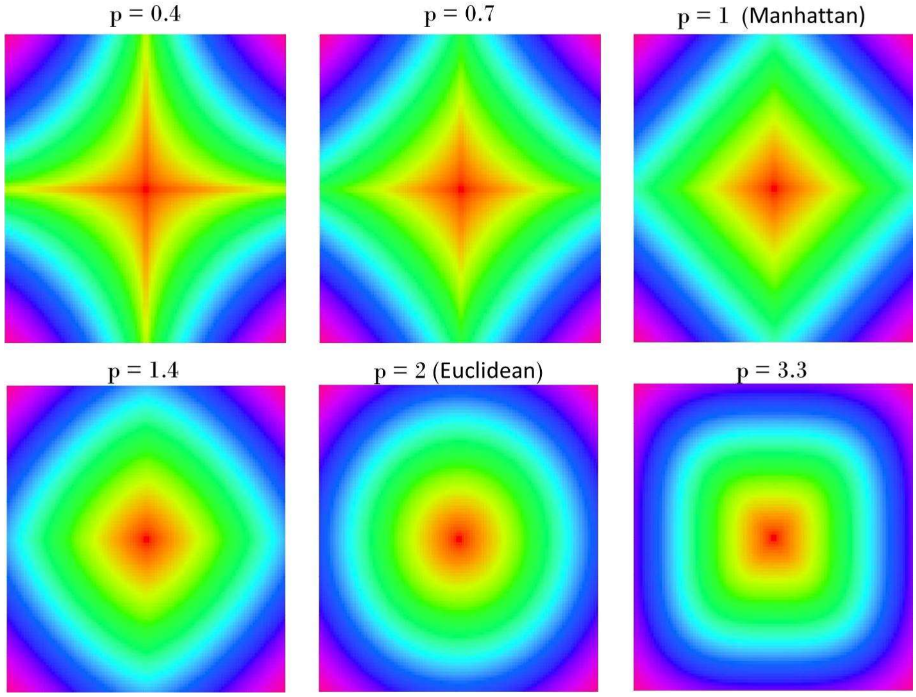


Figure 4.10: Minkowski distance gradient for distinct p.

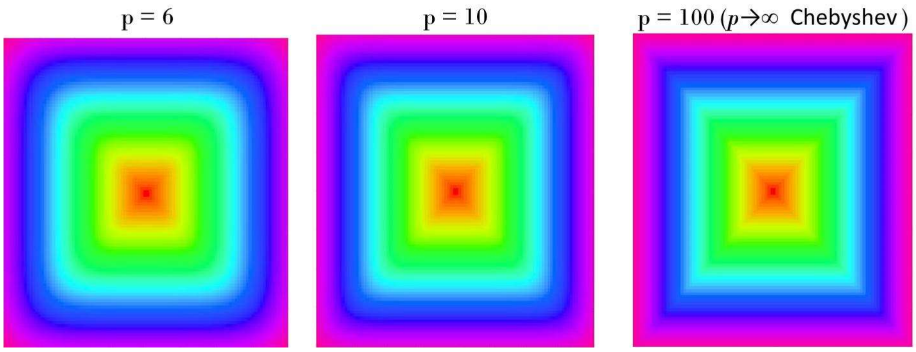


Figure 4.11: Minkowski distance gradient for distinct p higher values.

Figure 4.12 presents the gradient of the Mahalanobis distance according to distinct datasets distribution. The gradient over the grid reflects how far a point is from the dataset's distribution. Smaller Mahalanobis distances indicate that a point is closer and "more similar" to the distribution, while larger distances suggest the point is further away and less likely to belong to the dataset's distribution.

For information purposes, the application of other distance metrics discussed in the literature are presented. The Figure 4.13 shows some gradients of three more different metrics. The Cosine distance shows the regions of the space where vectors point in the same direction as the reference vector, as closer points regions (here in first quadrant). In contrast, regions with vectors pointing in the opposite direction (third quadrant) are considered as 'far points' region. Canberra distance highlights small distances (close to zero) and is often used for data that is scattered around an origin and uses only absolute values.Wave Hedges normalizes the differences allowing to use vectors that adopt different scales. One can see distances grouped by families in the table C.1 of Appendix C.

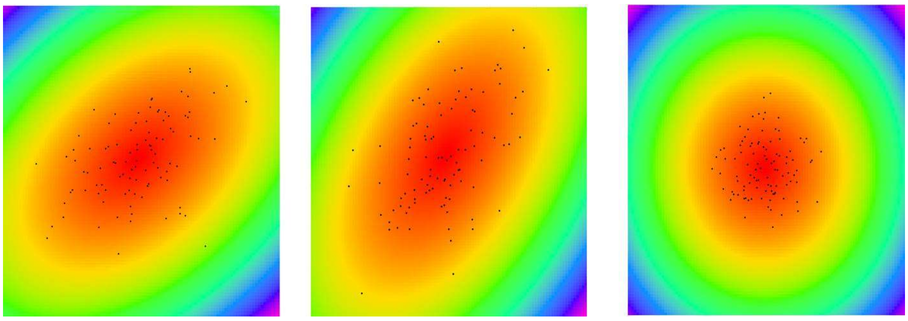


Figure 4.12: Mahalanobis distance gradient for distinct bivariated distributions.

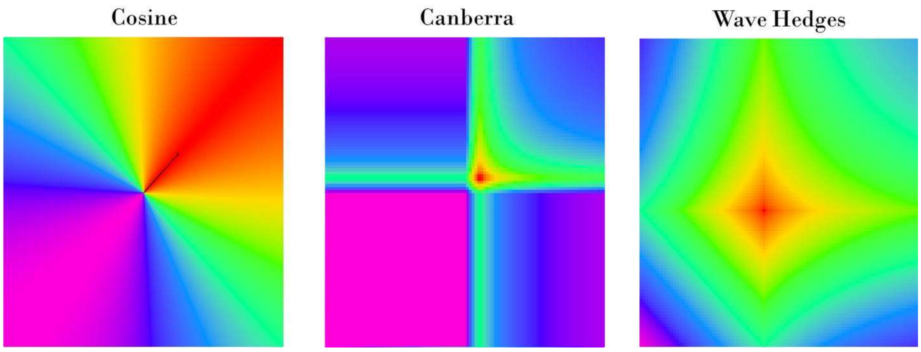


Figure 4.13: Cosine distance metric gradient from the indicated vector (in black). Canberra distance gradient from a reference point close to the origin. Wave Hedges distance from the origin.

4.2 Gabriel graph

A Gabriel graph is a method of connecting points to illustrate their proximity. Two points are connected if they are the closest pair within the circle whose diameter is the line segment between them. In other words, if two points are connected, there are no other points inside the ball defined by the segment connecting them.

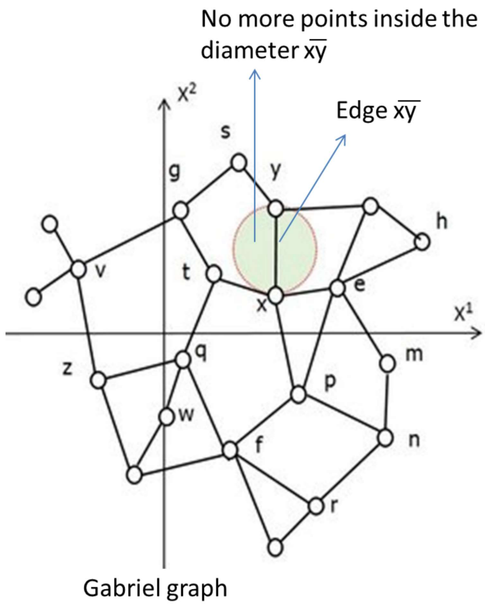
Gabriel graph (GG) is a sub-graph of the Delaunay triangulation [188]. In addition, it is a subset of points on the Voronoi diagram [189]. GG of a set of points S is a graph whose set of vertices $V \in S$ and its set of edges E must obey the following definition: given a data set $S = \{p_1, p_2, p_3, ..., p_x, ..., p_n\}$ where $\mathbf{p} \in \mathbb{R}^n$, a *Gabriel graph* of S is an undirected graph of vertices $V in S$ and edges in E. In that graph a pair of vertices $\{\mathbf{p}_x, \mathbf{p}_y\}$ are connected if, and only if, there is no points closer to one of them.

In this context, S is a set of patterns(points) in the n-dimensional space. The GG of S has vertices $\in \mathbb{R}^n$ and edges connecting pairwise that vertices. The pair p_x and p_y are connected by the Edge $\overrightarrow{E}_{xy}$ if the circle (sphere, hypersphere) having $\overrightarrow{E}_{xy}$ as a diameter, contains no other point of S. Figure 4.14 shows the main entities and concepts handled by a Gabriel graph.

Structural information extracted from a Gabriel graph has been applied to many problems in machine learning, particularly in the construction of large margin classifiers [190–193].

One can note some others features:

- The Gabriel Graph is a planar graph. It can be drawn on a two-dimensional plane without any edges crossing [194].
- It contains an Euclidean minimum spanning tree, which means that it can connect a set of points in a way that minimizes the total length of the edges [194].



Adjacency matrix of the Gabriel graph

V_u	...	r	s	t	...	x	y	z	...
...	...	...	...	...	...	...	...	...	...
r	...	1	0	0	...	0	0	0	...
s	...	0	1	0	...	0	1	0	...
t	...	0	0	1	...	1	0	0	...
...	...	...	...	...	...	...	...	...	...
x	...	0	0	1	...	1	1	0	...
y	...	0	1	1	...	1	1	0	...
z	...	1	0	0	...	0	0	1	...
...	...	...	...	...	...	...	...	...	...

Figure 4.14: Gabriel Graph and its adjacency matrix. If two points are connected then there is no other point inside the n-dimensional ball defined by the segment (edge) connecting them. It connects two points if there is no one else closer.

- The Gabriel Graph is based on a simple criterion: the edge between two points is included in the graph if and only if no other point lies within the circle whose diameter is defined by the two points [193].
- Since it is based only on distance calculations, on-chip embed machine learning model can also be executed [191].
- It can be also appropriate for online and incremental learning, since graph parameters are obtained directly from the data set, without need of user interaction for learning [190, 191].

The literature shows distinct variations of Gabriel Graph, including The *k-Gabriel graph* that allows at most k points in the ball [195–197]. Witness Gabriel graphs introduce a set of "witness" points that must not be in the ball [198]. *Relaxed Gabriel graphs* combining Delaunay triangulations, β-skeletons[1] and Gabriel graph [200].

4.3 Torres at all's GGClassification

The Gabriel graph-based classifier (*GGClassification*) adopts in the present investigation was proposed by Torres at all in [191]. It is a large margin classifier whose discrimination function is driven by margin vectors, structural SVs (SSVs), which are analogous to the SVM's support vectors. The mainly difference lays is the fact that GGClassification's SVs are obtained directly from the own data set structure.The researchers point out that there is only one input needed to generate the model, which is the set of distances between the patterns in the training set.

The idea is generating an adjacency matrix that represents the Gabriel graph of the training dataset. After that, seeking for the graph's edges that are connecting vertices (patterns) labeled

[1]The graph will connect two points if the lens-shaped region formed by two intersecting circles (with radius controlled by a parameter β) does not contain any other points. It was first defined in [199]

with distinct classes. Theses patterns lay on the frontier between classes [3]. Such edges, whose vertices belong to opposite classes, will guide the generation of the separation hyperplanes. The hyperplanes can be generated passing by the middle points of the edges connecting vertices of different classes. So, the combination of all hyperplanes yields a large margin classifier that is based on the spatial information of all patterns [191]. Individual hyperplanes (classifiers) are combined in order to design a global large margin classifier that minimizes the error corresponding to the learning data set.

Figure 4.15 shows the main concepts involving Gabriel graph classification method.

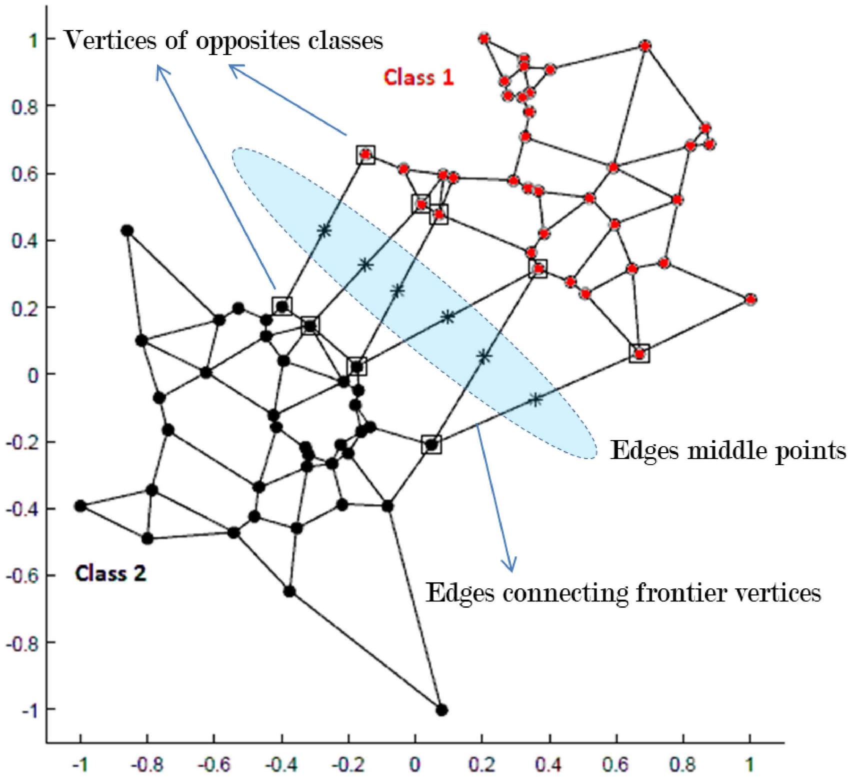


Figure 4.15: Gabriel graph margin maximization - Adapted from [3]

In certain types of problems, noise and overlap between data classes can occur. Consequently, the methodology for identifying the support edges may result in selecting vertices that are distant from the optimized margin [3].

To address this issue GGClassification adopts some metrics to quantify the purity of the examples (samples) based on the descriptors of [201]. Such metrics can help to identify populated and mixed class areas. On can see below some relevant metrics:

- $V_{dt}(x_i)$: Vertex degree. Total number of Gabriel-neighbors of the pattern x_i.
- $V_{eq}(x_i)$: Number of Gabriel-neighbors that belongs to the same class of the pattern x_i.
- $V_{op}(x_i)$: Number of Gabriel-neighbors that belongs to the opposite class of the pattern x_i.
- Sample Quality:

$$Q(x_i) = \frac{V_{eq}(x_i)}{V_{dt}(x_i)} \tag{4.1}$$

Figure 4.16 shows a graph with Q metric corresponding to each vertex (example). The samples belong to exclusive classes (+1 or -1, red or black).

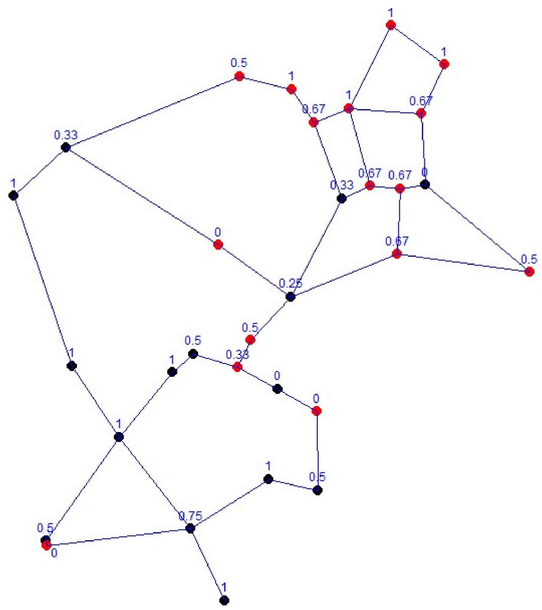


Figure 4.16: Q metric of each example on graph. Colors red and black represent distinct classes.

The Q score of the samples can be used to divide the labels into groups. For instance, $Q(x_i) = 0$: x_i is a noise. In this situation, all the Gabriel-neighbors are from the opposite class. If $Q(x_i) = 1$ then all the Gabriel-neighbors are from the same class of x_i. If $0 < Q(x_i) < 1$: x_i can be select as a support vector or can be labeled as "in overlap region".

As it has been conceived by the authors, Q metric is then used to remove certain samples from the training set. The samples exclusion is determined by thresholds calculated for each class, in other words, there are two thresholds, one for class +1 (TC^+) and another for class -1 (TC^-). If the sample's Q value is lower than its class threshold then it will be removed from the training set [3, 191].

One can see details bellow:

1. For each sample $\mathbf{x}_i \in \tilde{G}$, calculate $Q(x_i)$ via equation 4.1 .

2. Group $Q(x_i)$ by class, such that Q^+ and Q^- represent the quality measure for samples with labels $+1$ and -1, respectively.

3. Group $Q(x_i)$ by class, such that Q^+ and Q^- represent the quality measure for samples with labels $+1$ and -1, respectively.

$$nc^+ = \text{count}(x_i^+), \quad nc^- = \text{count}(x_i^-)$$
$$nc_+^+ = \text{count}(x_i^+ \leftrightarrow x_j^+), \quad nc_-^- = \text{count}(x_i^- \leftrightarrow x_j^-)$$
$$Q^+ = \frac{nc_+^+}{nc^+}, \quad Q^- = \frac{nc_-^-}{nc^-}$$

4. Calculate the threshold values TC^+ and TC^- for each class as the average of the quality measure values belonging to Q^+ and Q^-, where

$$TC^+ = \frac{\sum\limits_{Q(x_i) \in Q^+} Q(x_i)}{|Q^+|}, \quad TC^- = \frac{\sum\limits_{Q(x_i) \in Q^-} Q(x_i)}{|Q^-|}.$$

5. Remove from the graph $\tilde{G}$ all vertices (samples) x_i^+ where $Q(x_i) \leq TC^+$ and all vertices x_i^- which $Q(x_i) \leq TC^-$.

Other important GGClassification method's metric for the present work context is $V_{eq}(x_i)$. It can point to actual samples relayed in a densely populated area of the same class. It can be interesting in tasks involving identifying feasible examples (both factual and counterfactual).

It is also possible to use statistic metrics directly. Depending on the needs such metrics can make it easier selecting samples instances. For instance, Figure 4.17 shows a Gabriel graph containing a probability density metric for each sample.

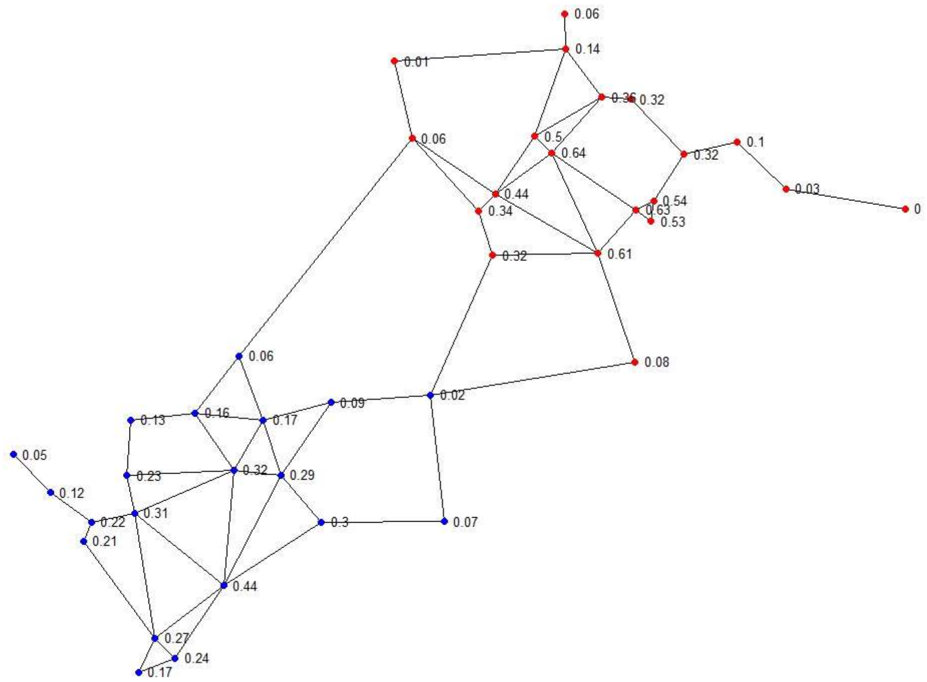


Figure 4.17: Probability Density metric of each example on graph.

4.4 Voronoi Diagram

Voronoi Diagram is a geometric structure that provides a vast amount of information about proximity (similarity) between a given set of points or objects [202]. It consists in a widely studied data structure in the field of Computational Geometry.

The concept is not new, dating back more than a century, however, there are still many applications in several areas such as biology, computer science, computer graphics, pattern recognition, computer vision, engineering among others [203].

The Voronoi Diagram can be understood as being an allocation of space to some key locations called sites or centroids. The space is divided into regions called cells according to how close its points are to these centroids.

Formalizing: consider a set P of key points in space called *sites* or centroids. We have $P = \{s_1, s_2, s_3, ..., s_n\}$, where each point $s_i \in P$ correspond to sites (key locations). One can define the Voronoi diagram of P as the space subdivision in a set of n Voronoi cells $Vc = \{Vc_1, Vc_2, Vc_3, ..., Vc_n\}$, one cell for each site $s_i \in P$, with the important property that, any point p within the cell Vc_i correspondent to the site s_i is closer to that site s_i than to any other site $j\ in\ P$.

In other words, any point p of the space is located within a certain cell Vc_i, if and only if, $dist(p, s_i) < dist(p, s_j) \quad \forall \quad s_j \in P$ with $j \neq i$. The Voronoi cell $Vc_i \in Vor(P)$ corresponds to the site s_i and can be seen as the area of influence of that site s_i [204].

The Voronoi diagram generated from P is denoted here as $Vor(P)$. The geometric data structure $Vor(P)$ consists of: all sites in P, the set of generated cells Vc (corresponding to the complete space subdivisions generated from P), the edges that delineates each cell and, in addition, the vertices of such cells.

Figure 4.18 shows the Voronoi diagram generated from a set of sites. The points showed in the picture correspond to the sites (patterns, samples, examples, observations...). The subdivisions (partitioned areas) of space, like the one yellow highlighted, are called Voronoi cells. The lines that surround the Voronoi cells are called Voronoi edges. In a 2D space, one can treat cells as polygons and the edges as the sides of these polygons. The intersection point between the edges are called vertices.

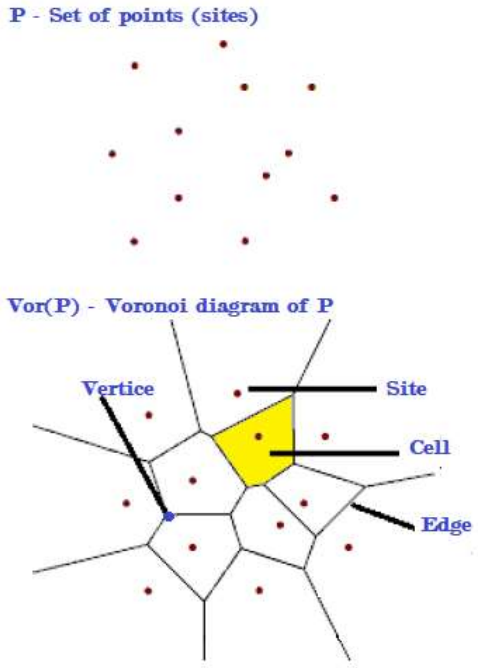


Figure 4.18: Set of points and its Voronoi diagram.

An important problem in pattern recognition system design concerns synthesis of the multidimensional decision surfaces of the pattern categorizer from given sets of previously identified sample pattern measurement vectors [205].

One can use the Voronoi diagram structure to identify such separation surfaces and maximize the margin of the separating hyperplanes. For instance, it is possible to use the border edges (edges that separate cells belonging to sites of opposite classes) to generate the decision boundary.

Under SVM point of view, the frontier sites represent the support vectors and the edges between distinct classes can be seen as the optimized hyperplanes.

Other notable aspect of the Voronoi diagram is its connection to both convex hulls and clustering [204, 206, 207]. This property makes it a valuable tool for dealing with classification matters and approaching the Minimal Connector Problem [208].

The Voronoi diagram's geometric structure facilitates the identification of a spatial partition close to a given point. Additionally, this diagram allows determining which partition or cell a particular point should be located based on its attributes. In other words, it provides information on where a particular point should be placed.

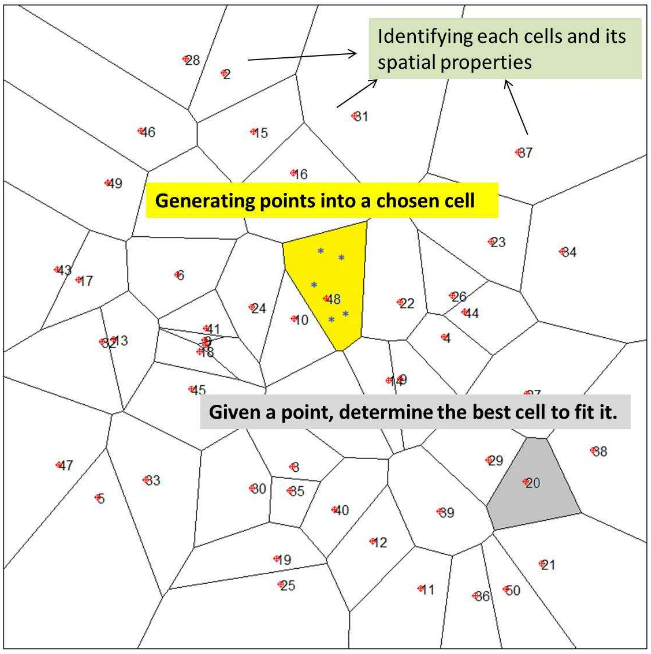


Figure 4.19: Voronoi cells area statistics.

Moreover, the Voronoi diagram enables the identification and instantiation of multiple new points within a cell of influence belonging to a particular site. Another helpful feature of the Voronoi diagram is its ability to extract valuable statistical metrics from the produced space partition. Figure 4.21 show a Voronoi diagram with the area (volume, hypervolume) of each partition (cell). Figure 4.20 show a diagram highlighting cells in function of their statistical properties inside the all data set.

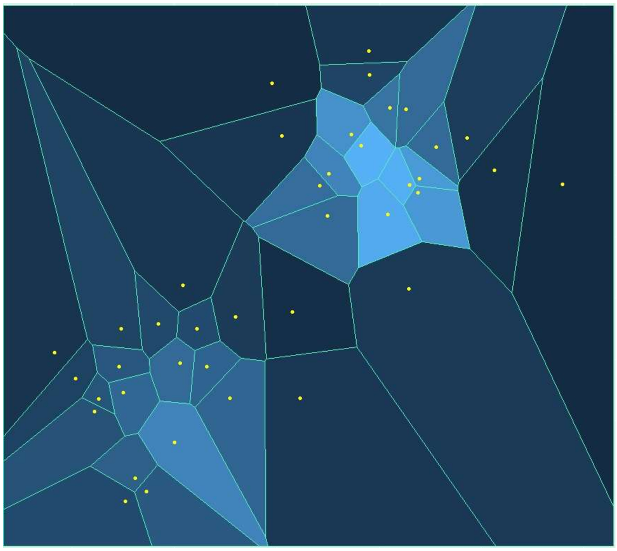


Figure 4.20: Voronoi cells area statistics.

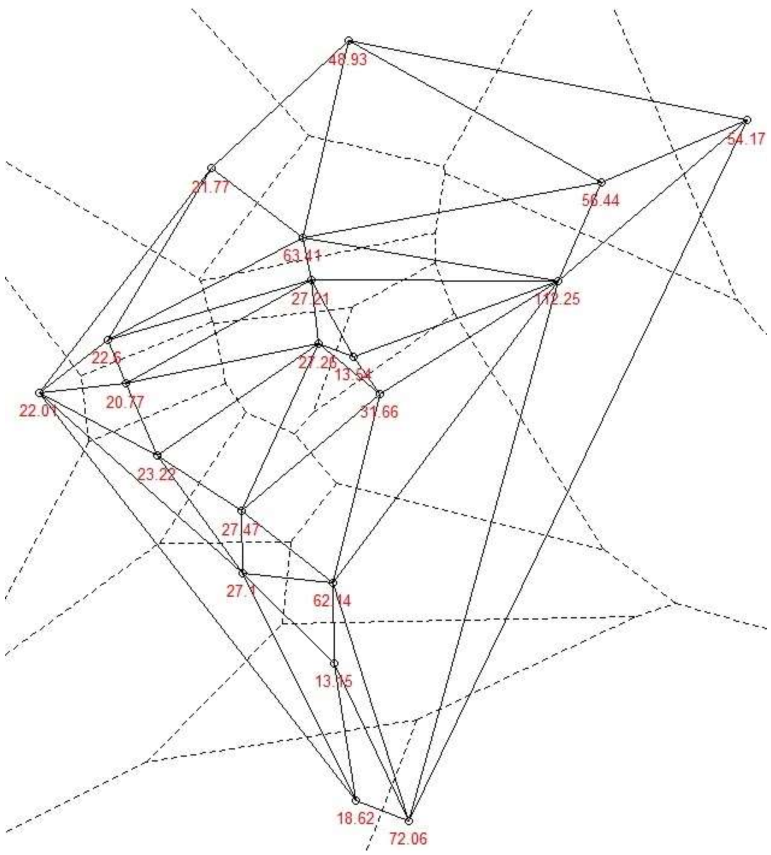


Figure 4.21: Voronoi cells area statistics.

4.5 Dominating Set

Considering a Graph G with vertices and edges (V, E), the dominating set is a subset $D \subseteq V$, given that for every node $v \in V$, either $v \in D$ or a neighbor u of v belongs to D [209].

To explain it simply, a dominating set of a graph is a group of vertices such that every vertex in the graph is either a member of the dominating set or adjacent to a member.

This means that, its possible to access all the graph from the dominating set vertices. From ds it is possible to ensure that all other vertices of the graph are either included in own ds or adjacent to one of its members.

On can see in Fig. 4.22 that all the vertices of the graph $G = \{a, b, c, d, f, g, h, i, j, k\}$ are in the dominating set $DS = \{e, f, g\}$ or can be directly accessed for some one in there.

Dominating set incorporates statistical properties of the data set sampled within a time frame. The elements of the dominating set, which are key points of the sample data in the original graph, provides some level of representativeness to the outcome [189].

The core structural information extract from the data permits the application of dominating sets in different scenarios, mainly in modeling complex networks, For example:

Social Networks: Dominating sets can be used to identify influential individuals in social networks.

Bioinformatics: Dominating sets can be used to identify the most important proteins or genes in a biological structure.

Epidemiology: Immunization optimization studies, dominating sets can be used to identify a group of individuals that should be vaccinated in order to minimize the disease spreading.

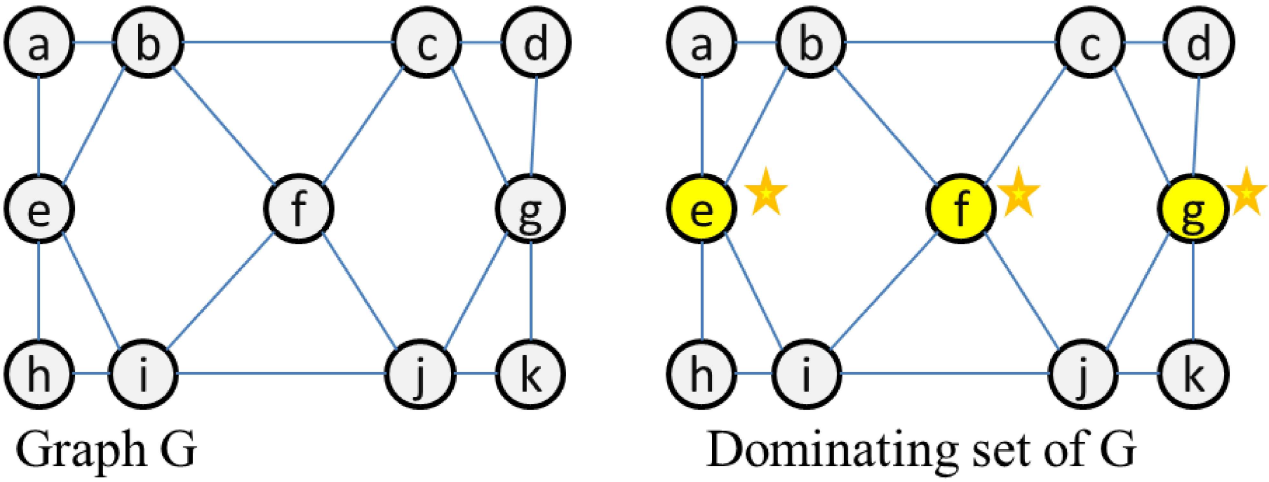


Figure 4.22: A graph G and one of its dominating sets. The highlighted vertices constitute one of the dominating set of the Graph G.

Communication networks: Identifying a minimum dominating set of nodes that can be used for efficient routing networks communication protocol packages; managing power in wireless sensor networks.

Utilities and services positioning: Identifying the optimal location for facilities such as sales points, police stations, fire stations, hospitals.

The dominating set have the examples that have import information regarding the localities. The dominating examples surroundings comprise relevant characteristics that can represent the largest space. Figure 4.23 demonstrate such property. The representativeness of a dominating

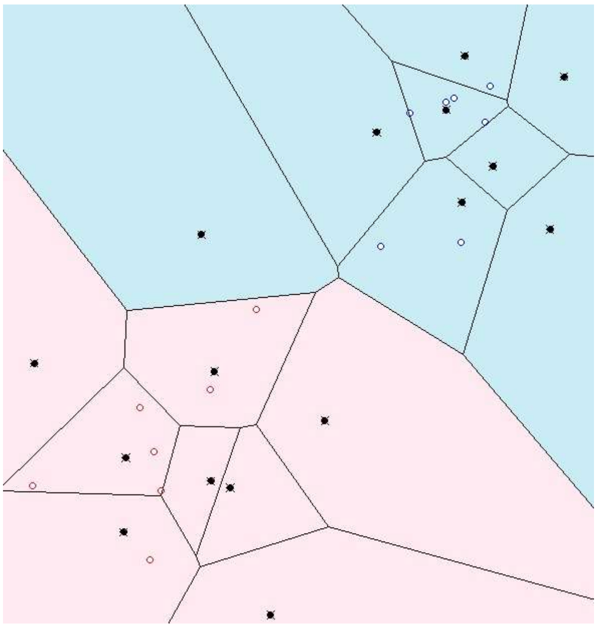


Figure 4.23: Dominating set representativeness

set depends on the specific context and purpose of the analysis. In some cases, a small dominating set may provide a good approximation of the entire dataset, while in other cases a larger dominating set may be needed to capture the full complexity of the dataset.

4.6 The Curse of Dimensionality

In high-dimensional spaces, the Euclidean distance tends to grow disproportionately. All the points can appear far way and almost equidistant from each other. This phenomenon is often referred to as the "curse of dimensionality [210, 211]."

One can see an example of the non-linear increase in distances on Figure 4.24. It shows the diagonals of the square, cube, and tesseract. It is possible to follow the distance increasing by adding unitary-length dimensions in space. By repeating the series shown in the Figure 4.24

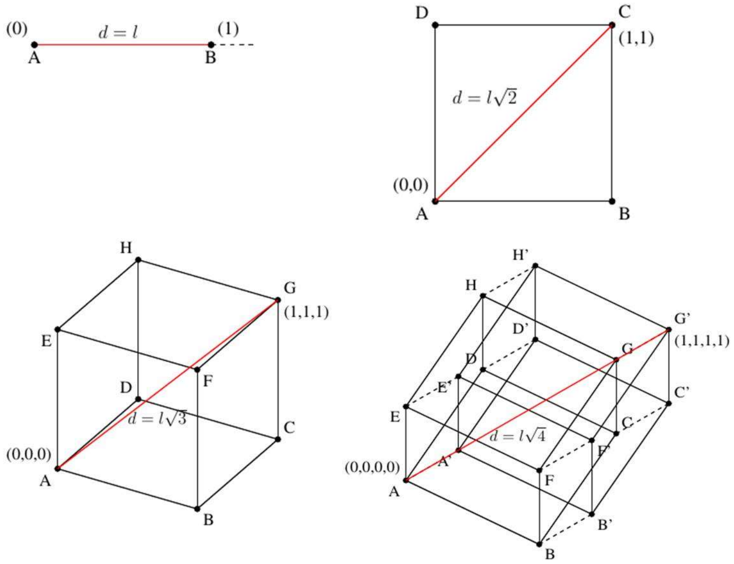


Figure 4.24: Distance non-linear increasing by adding unitary extensions in space.

for dimensions from 1 to 1000 and generating a line chart of the *diagonal length* in the function of the *number of dimensions* of the hypercubes, the result is the curve exhibited in Figure 4.25.

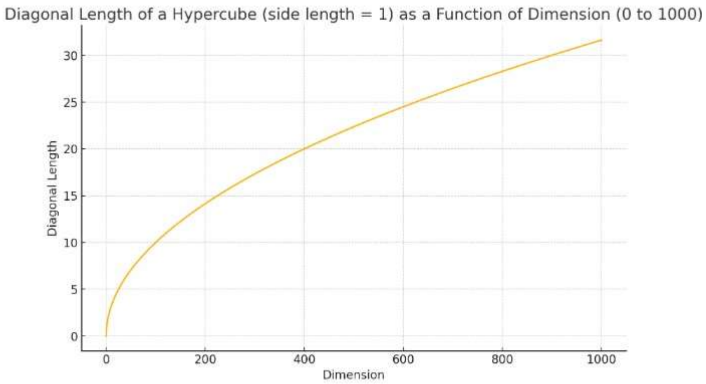


Figure 4.25: Non-linear increasing of the Hypercubes' diagonal length by adding unitary extensions in space.

One can face even more severe conditions by handling hyperspheres instead of hypercubes. The 1-D space inside a 1-D sphere of radius r is the length of a line segment $2r$. The 2-D space inside a 2-D sphere of radius r is the circular area of πr^2. The 3-D space inside a 3-D sphere of radius r is the spherical volume $\frac{4}{3}\pi r^3$. The 4-D space inside a 4-D sphere of radius r is the hypersphere of hypervolume equal to $\frac{1}{2}\pi^2 r^4$ [182]. For Higher dimension spaces, strange-to-intuition effects begin to appear.As one increases the dimension of a n-sphere, with fixed radius, into the realm of hypergeometry, the volume increases. For a while... after that, the counterintuitive result is: as one keeps adding dimensions, eventually the hypervolume start decreasing [212].

Figure 4.26 presents the results of an experiment in which the dimensionality of an unit-radius n-sphere was incrementally increased from 1 to 50.

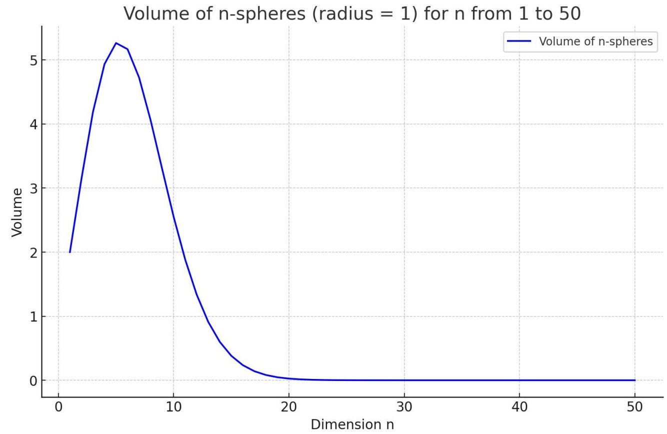


Figure 4.26: Counter intuitive behavior of th n-hypersphere hypervolume. It starts with growth, reaches a maximum value and then immediately decreases exponentially.

Conversely, when the reference radius is increased by a single unit, the behavior of the hypervolume remains the same, but its magnitude increases exponentially. Figure 4.27 compares the hypervolume evolution of two n-hyperspheres, radius = 1 and radius = 2. Due to the high difference between the volumes magnitude, the chart presents two distinct y (Hypervolume) axis.

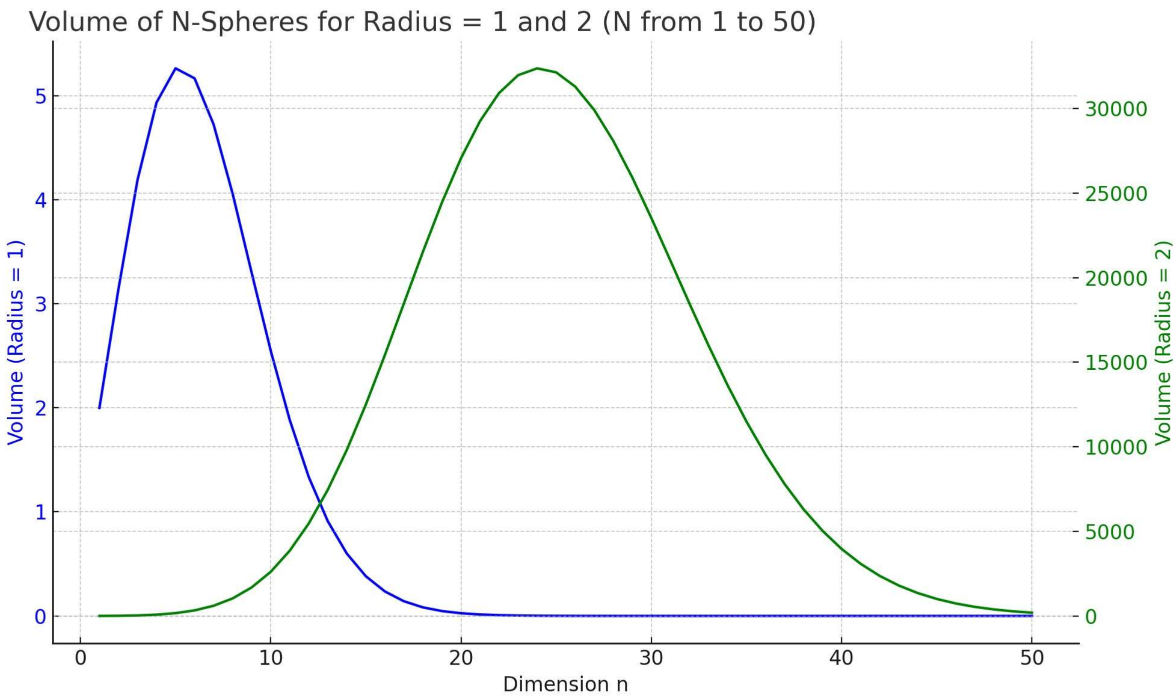


Figure 4.27: Comparison between hypervolumes evolution: radius = 1 vs radius = 2 $n-$hypersphere

As dimensionality grows, the data becomes progressively sparse, and the density of data points diminishes relative to the expanding volume. Consequently, the required data volume must grow exponentially to match this expansion effectively and ensure reliability. Numerically, this increases the average distance between points, while the variation between these same distances tends to decrease, leading to a more homogeneous distribution of distances. On can see such a practical effects on the following section.

4.7 Experiments on the theory exposed

Te present section details the experiments conducted to investigate the hypotheses derived from the theoretical framework outlined earlier.

Creating computational experiments is interesting for several reasons. First, they transform abstract theoretical concepts into visual and interactive forms, making them more tangible and easier to understand—an advantage particularly valuable for geometric ideas, where seeing shapes and transformations can significantly deepen comprehension. Second, these experiments provide a controlled environment for testing and validating theories, ensuring their robustness and reliability. Even well-established models benefit from such scrutiny, as experiments may uncover edge cases or reveal unexpected behaviors. Finally, this process fosters innovation and improvement; for instance, by exposing new insights or inspiring novel approaches, computational with visual appeal experiments can lead to advancements that theory alone might not readily yield.

4.7.1 Dimensionality Effects on distance matrix statistics

An experiment was developed to observe the dimensionality effect on an euclidean distance matrix. In the experiment, dimensions were progressively added to 200 randomly created points. Then, the distance matrix was generated from which some statistics were extracted, including the mean, standard deviation and coefficient of variation. This effect can be observed in Figure 4.28. One can see the increasing behavior of the mean of the distances in function of the dimensionality, while the coefficient of variation (CV)[1] decreases.

To observe the behavior of the homogeneity of the distances in even higher dimension, the previous experiment was re-executed considering dimensions from 1 to 1000. One can see the homogeneity, here represented by the coefficient of variation, decreasing obeying a $1/x$ function, where x is the number of dimensions.

4.7.2 Dimensionality Effects on Gabriel graph

The context is that the data occupies an increasingly smaller fraction of the total available space. As a result, the distances between points become larger and more uniform, leading to a scenario where each point is significantly isolated from the others. This situation challenges the efficiency of methods like the Gabriel Graph, which rely on distances. An experiment was conducted to observe the effect of dimensionality on the characteristics of the Gabriel Graph. In this experiment, the Gabriel graph was generated from various data dimensions, and several statistical metrics of the resulting geometric structure were evaluated, including the mean distance within the distance matrix, the coefficient of variation, the number of connections from the adjacency matrix, and the individual number of connections for each point. Figures

[1]The CV is a measure of relative variability. It is defined as the ratio of the standard deviation to the mean ($CV = \frac{\sigma}{\mu} \times 100$), often expressed as a percentage. It is often used to compare the variability of data sets with different units, means or magnitudes

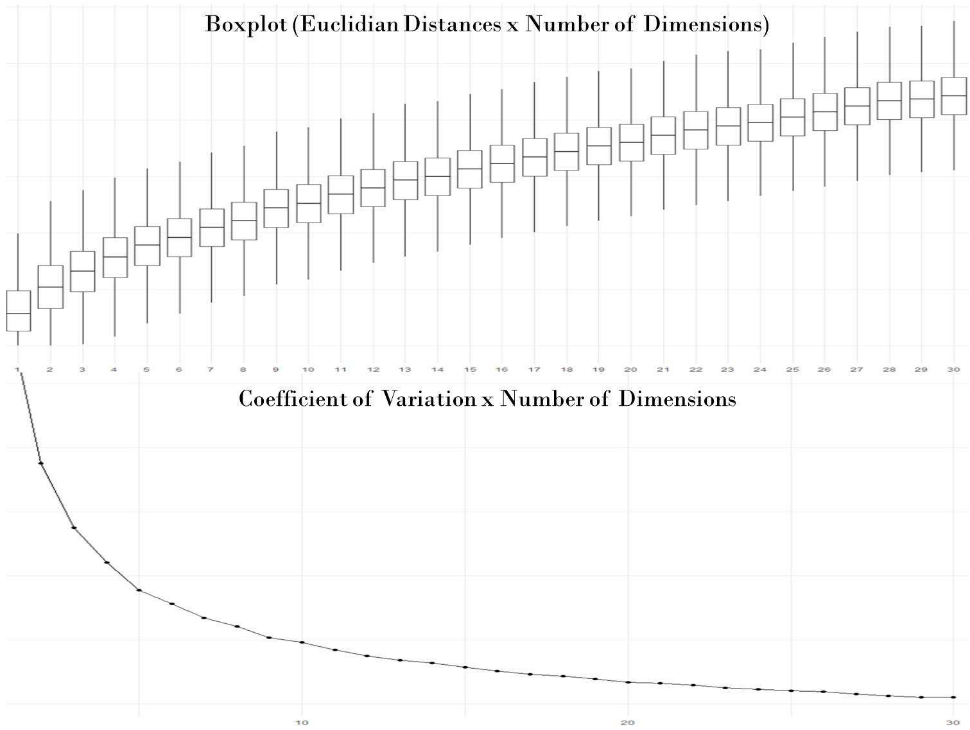


Figure 4.28: Behavior of the mean and the coefficient of variation of the distances.

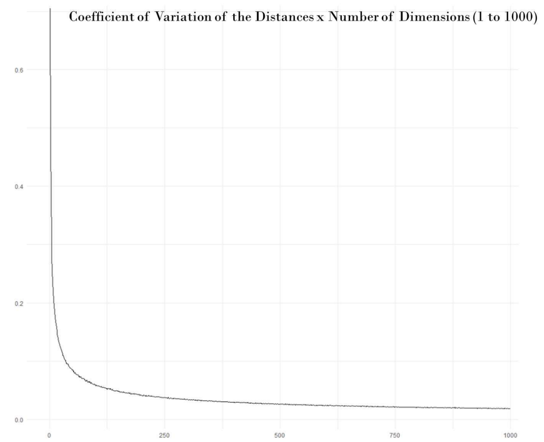


Figure 4.29: Behavior the coefficient of variation of the distances (dimensions 1 to 1000).

from 4.30 to 4.33 shows the results of how the dimensionality affects some of the Gabriel Graph attributes.

Figure 4.30 shows the mean increasing, as expected.

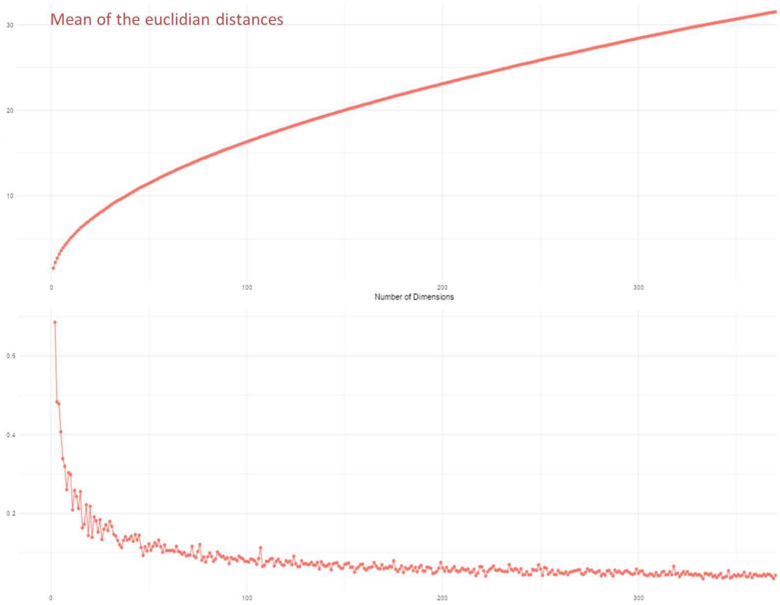


Figure 4.30: Evolution of the GG's distance matrix: average (dimensions 1 to 370).

Figure 4.31 brings the growing homogeneity of the values in the distance matrix. It is reflected by the coefficient of variance decreasing behavior. Figure 4.32 shows the increase in

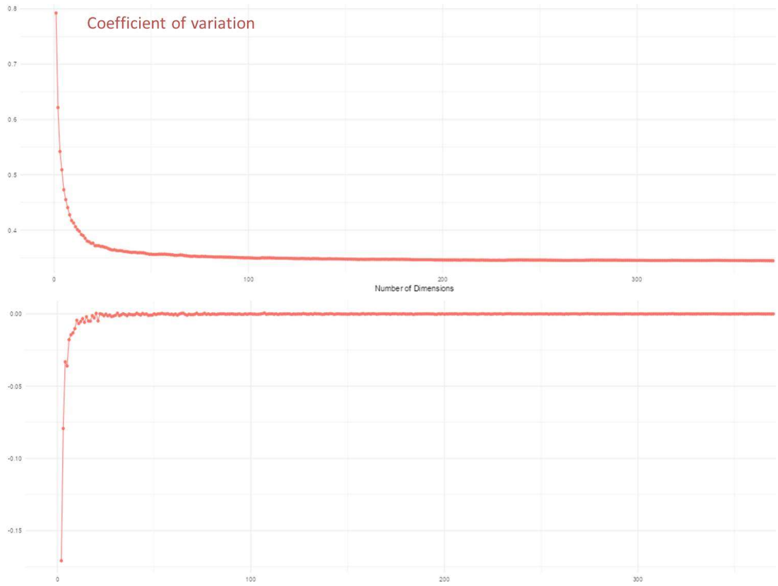


Figure 4.31: Evolution of the GG's distance matrix: coefficient of variation (dimensions 1 to 370).

the number of neighbors (individual connections) for each point in the adjacency matrix.

4.7.3 Dimensionality Effects on similarity-based classifiers

Similarity-based classifiers estimate the class label of a test sample based on the similarities between the test sample and a set of labeled training samples, and the pairwise similarities

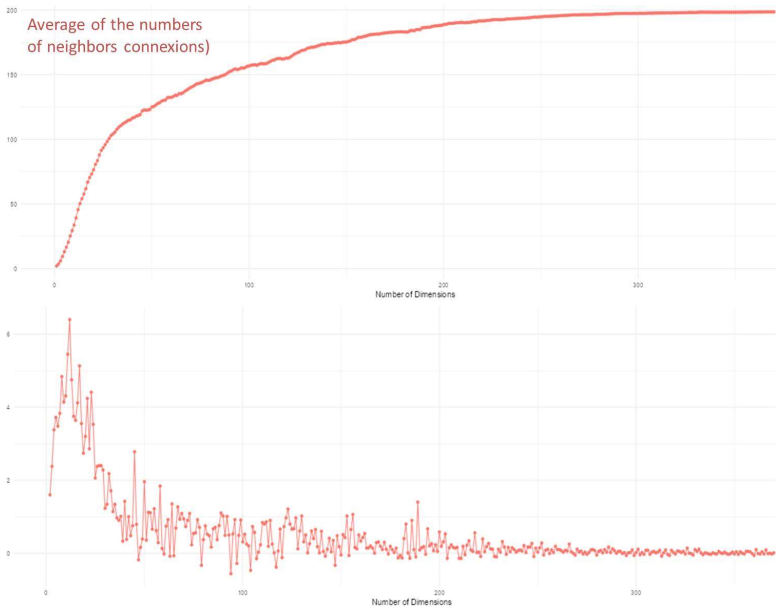


Figure 4.32: Evolution of the GG's adjacency matrix: average of the numbers of neighbors (dimensions 1 to 370).

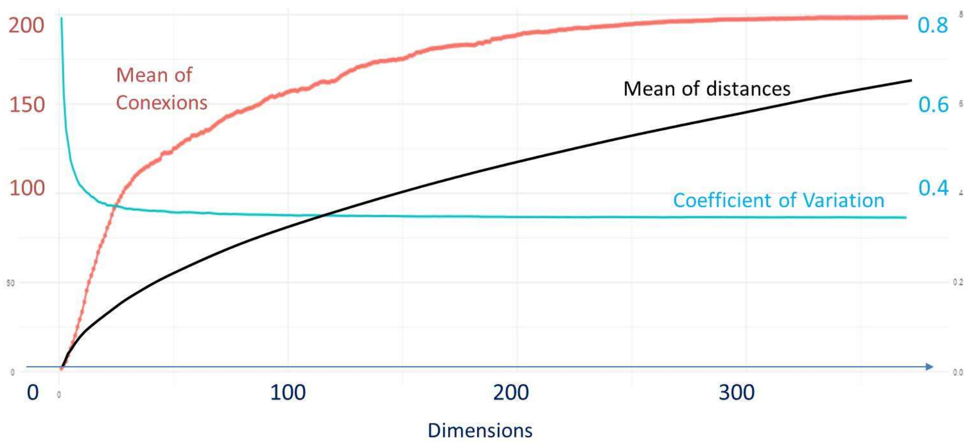


Figure 4.33: Behavior of the distance in higher dimensions leading to multiples connections: at the highers dimensionality, each point in the graph is connected to all the others).

between the training samples [213]. The problem is that models trained on sparse data and non-representative distances variance may not capture general patterns well. The model is more likely to overfit to the specific noise in the training set. With insufficient data, the model tends to overfit the training data [214]. Depending of the degree of the sparsity, each point itself can appear as a cluster.

Tasks regarding searching, organizing, filtering, and clustering usually rely on calculating similarity metrics. The distances metrics calculation and distance values behavior can be tricky in high dimensional scenarios. Objects in high-dimensional spaces tend to appear scattered and far away from each other, while the variation of the distances tend to be more similar (distances become more homogeneous. Vast, but more uniform). This makes traditional distances metrics, organization and classification methods less effective when compared to smaller-dimensions space scenarios.

To qualitatively observe the drop in performance of similarity-based models, an experiment was conducted in which two models (SVM and GGClassification) were trained and tested on databases with progressively larger dimensionality. Both the models are trained from 10 different folds within an iterative loop that increases the dimensionality progressively from 1 to 150. Figure 4.34 presents the mean of GGClassification's AUC (considering each one of the 10 folds) in each dimensionality. Figure 4.35 shows the same performance metric correspondent

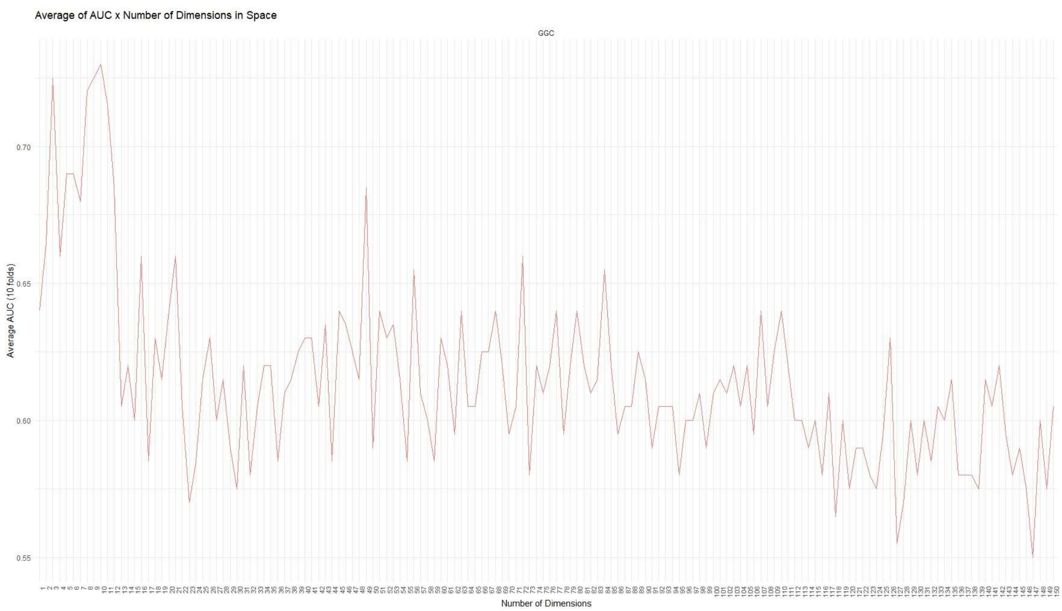


Figure 4.34: GGClassification AUC in function of the number of dimensions (From 1 to 150).

to the SVM.

Several approaches have been proposed to address the curse's issues. Dimensionality reduction techniques, such as PCA and t-SNE, have shown promise in improving classification accuracy and preserving data utility [214, 215]. Feature selection methods that account for the curse of dimensionality and focus on discriminative features have demonstrated competitive performance [216, 217]. Compositionally sparse functions [1] [218].Additionally, strategies such as addressing variance contributed by noisy variables [219] and exploring non-local solutions [220] have been suggested. These approaches aim to enhance the efficiency and effectiveness of learning algorithms in high-dimensional spaces across various applications.

[1]Compositionally sparse functions are functions that can be represented as the composition of a relatively small number of constituent functions, each depending on only a small number of variables [218].

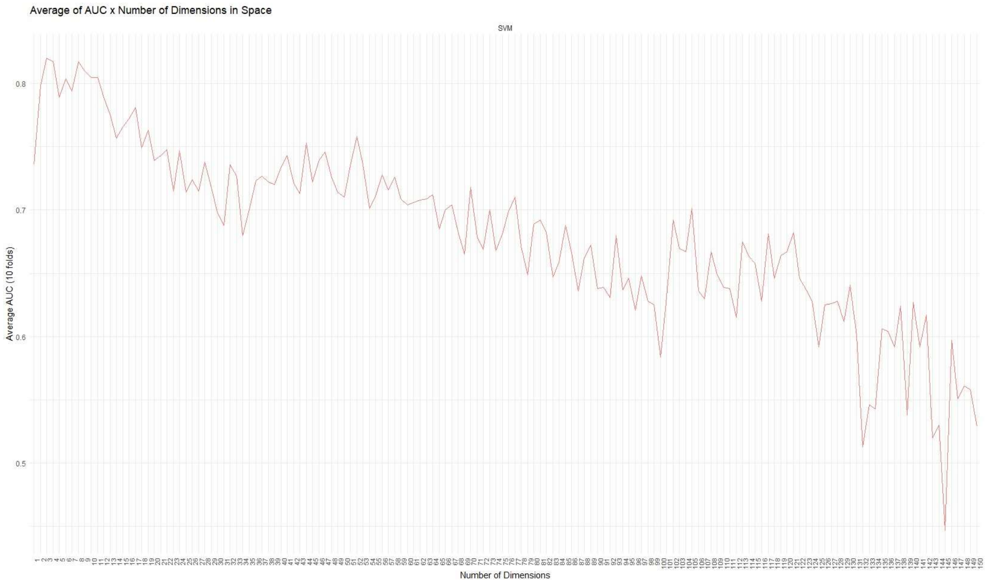


Figure 4.35: SVM AUC in function of the number of dimensions (From 1 to 150).

4.7.4 Distance metric influence on Gabriel graph properties

An experiment was developed to observe the combined effect of distance metric and dimensionality effect on Gabriel graph properties. In the experiment, dimensions were progressively added to randomly created points. Then, the Minkowski distance matrix was generated for distinct p values.

Besides that, it was included weights to the dimensions (features). Each weight was proportional to the inverse of the correspondent dimension variances. Variance-based dimension weights are used to adjust the importance of each dimension in a dataset based on their variability. The idea behind this is that dimensions with lower variance are given more weight as they are considered more stable. Small differences on such dimensions will have a larger influence on the total distance. The equation 4.2 indicates the calculation of the inverse variance weights, as follows:

$$w_i = \frac{1}{\sigma_i^2 + \epsilon} \tag{4.2}$$

where

- w_i : weight of the dimension i.
- σ_i^2 : variance of the dimension i.
- ϵ : small value to avoid *division by zero* error.

One more step was performed aiming to normalize the dimension weights. Equation 4.3 presents the calculation.

$$w_i = \frac{\frac{1}{\sigma_i^2+\epsilon}}{\frac{1}{n}\sum_{j=1}^{n}\frac{1}{\sigma_j^2+\epsilon}} \tag{4.3}$$

where n is the number of dimensions.

Figure 4.36 illustrates the variation in the average distances corresponding to different values of p and weights (weighted distance or not weighted).

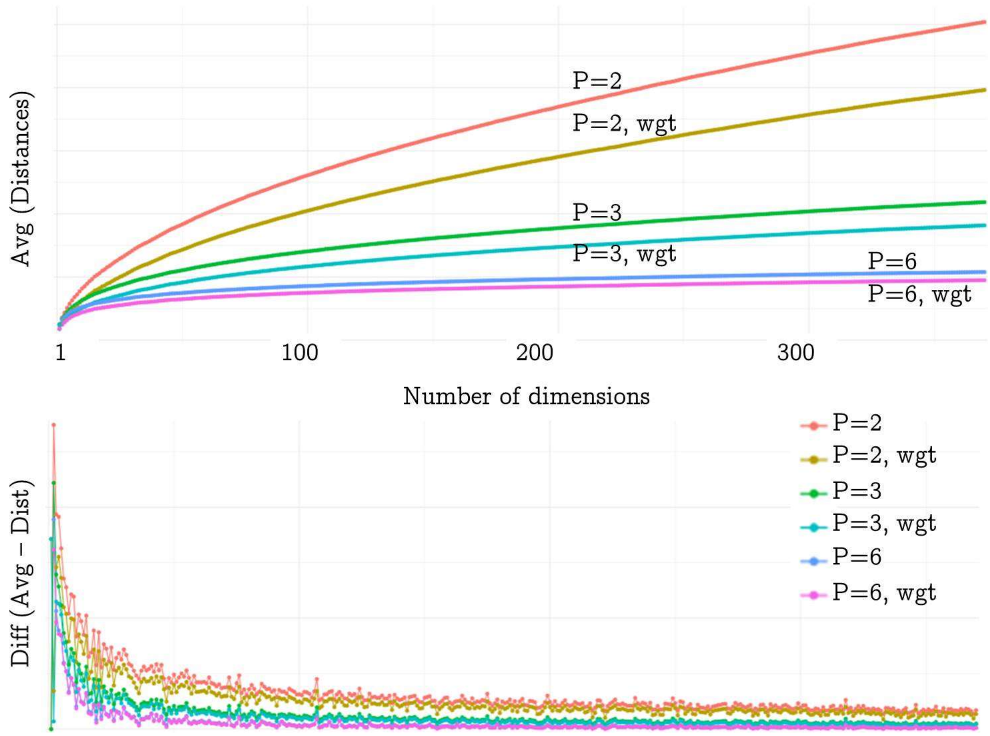


Figure 4.36: Average distances comparison using distinct Minkowski p and weight (dimensions from 1 to 370)

One can observe that the rate of increase in average distance exhibits variation according to the choice of parameter p and the dimension-specific weighting strategy.

Figure 4.37 shows the variation in the coefficient of variation of the distances and the average number of neighbors for different values of p and weights (weighted distance or not weighted).

On can see that the average of the number of neighbors tended to be lower for higher p and. While the coefficient of variation of the distance tended to be slightly larger to weighted distances.

4.7.5 Distance metric influence on similarity-based classifiers

Experiments were performed to observe the distance metric influence in a simple similarity-metric classifier. To do so, a KNN [1] model was adopted. The purpose was investigating the effect of Minkowski distance parameter p on the model performance while facing different classification tasks.

Both a grid searching and a Differential Evolution (DE) [2] were performed to assess distinct combinations of k and p against the classification of four UCI [222] benchmark datasets (Appendicitis, Wisconsin Breast Cancer, Pima and Ionosphere). Each k vs p distinct combination was tested 30 times using different folds (random 70-30 splits). DE optimization's

[1] Roughly, a KNN calculates the distance between the sample x (to be classified) and all the other points in the training data set. So, the closest (smallest distance) K neighbors are taken and the class of the sample x is defined based on the label (class) of these same k neighbors. The resulting label is the majoritarian label considering these k nearest neighbors.

[2] Differential Evolution is an optimization algorithm introduced by Storn and Price ([221]), known for its effectiveness in handling multimodal and non-differentiable objective functions. It iteratively improves a population of candidate solutions by combining and mutating them, followed by individuals selection based on fitness (quality) scores,this way mimicking a biological evolutionary process.

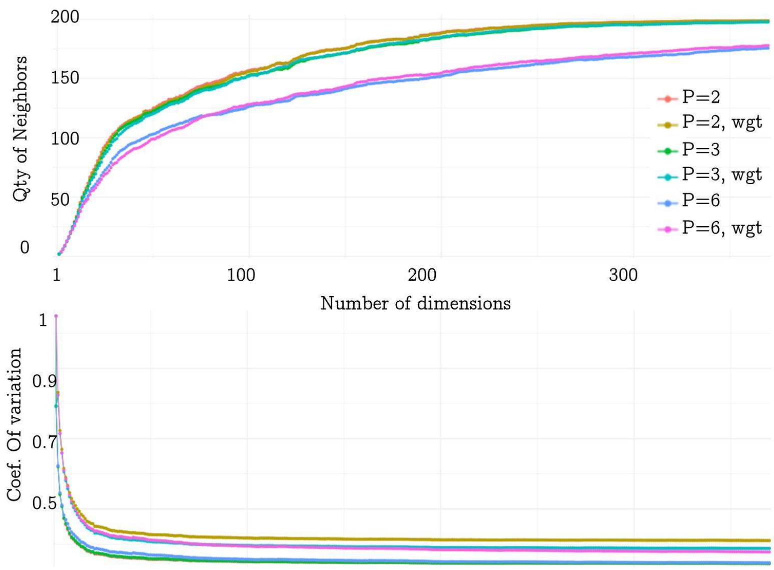


Figure 4.37: Coefficient of variation and number of neighbors comparison using distinct Minkowski p and weight (dimensions from 1 to 370)

objective function was given from the average accuracy in ten runs. In other words, the aptitude (quality) metric of each member (each pair p,k) was obtained from 30 different executions with 70-30% distinct splits. The median accuracy of these ten runs was taken as a fitness metric. The higher, the better. Population size was set as 20 members (ten times the number of parameters) and the max number of generations equal to 200.

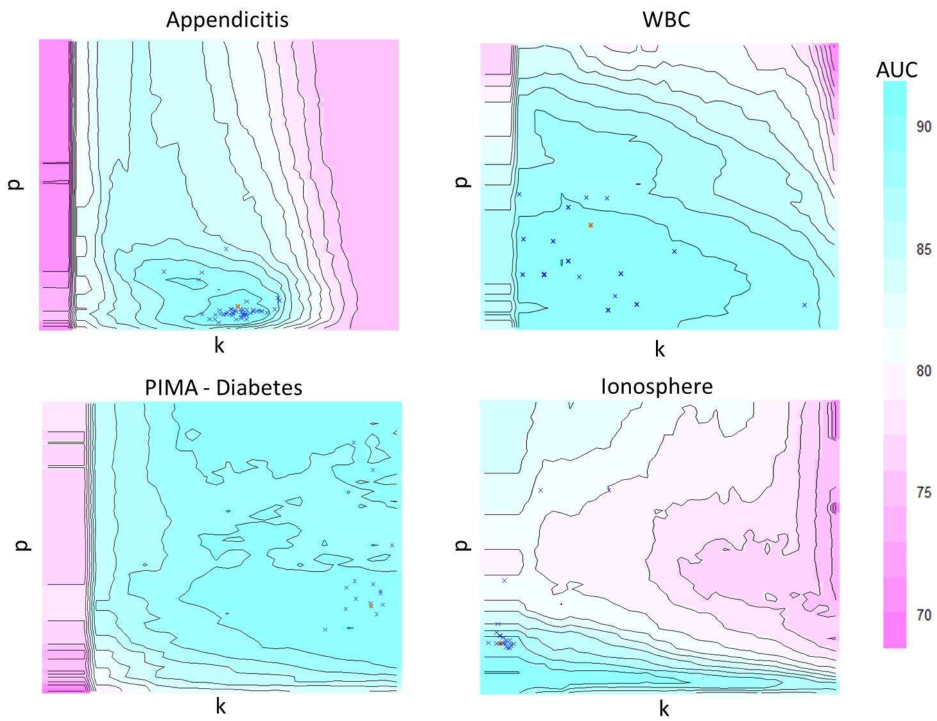


Figure 4.38: Performance (Accuracy) variation for each k vs p distinct combination. Accuracy obtained from 30 runs. p varying from 0.1 to 6 step =0.1. K range depending on the dataset size.

Figure 4.38 shows the DE Optimization best solutions over the grid search results. It is possible to see that the distance metric affects the Knn model performance, even working under the same hyperparameter k. One can find distinct accuracy for the same k along the p variation.

4.7.6 Impact of Quality Thresholds TC+ and TC- on GGClassification Performance

Recalling the theory, GGClassification method does not make use of external parameters. All the information necessary for its classification strategy is embedded in the data set, whose is being represented by Gabriel graph. The descriptors used to deal with noise and class overlapping are fixed metrics, TC^+ and TC^-, which are automatically obtained from the graph itself.

The original method uses fixed values for both class thresholds. However, given the importance of border quality in the present work, experiments were conducted to make the class thresholds more flexible by testing different values for both.

In this sense, the strategy consisted in to perform a grid search ($TC^+ versus TC^-$) from 0 to 1, using resolution equal to 0.05. That is, each distinct combination ($TC^+ vs. TC^-$) in the range [0,0.05,0.1,...,0.95] was assessed. In order to properly evaluate each parameter combination, it was applied k-fold cross validation adopting k=10. The grid search with k-fold cross validation has performed facing 19 binary classification bench march data sets.

In addition to the classes threshold values variation, another change in the original method was evaluated. It is about *shift* the class of the filtered pattern instead of *removing* it from the graph, as originally works. All different parameter selection from the grid search was applied to both filter options ("removing samples" and "shifting the class label of the samples").

To evaluate the experimental results, factorial analysis and paired analysis were performed in order to compare the results of the three distinct approaches (original one, threshold for remove samples, threshold for shift the class).

The experiments employed 19 real-world data sets. Where 16 available are in the UCI repository [222]. Data set Appendicitis is from KEEL-data set repository [223]. Breast Cancer Hess Probes [224]. The multiclass data set Glass was convert into binary classification by doing "7 versus. all". All data sets were normalized with mean $\bar{x} = 0$ and standard deviation $\sigma = 1$, without missing values.

One more set of experiments was carried out in order to investigate TC^+ and TC^- values selection through *differential evolution* (DE) optimization. Differential evolution was modeled by adopting three parameters vector as members, one parameter referring to TC^+, the second one to TC^- and the third to indicate the filter action ("remove" or "class shift").The intervals were set from 0 to 1.

The objective function was given from the average AUC in ten k-fold cross validation runs. In other words, the aptitude metric of each member (each trinominal vector TC^+, TC^-, "type of filter action") was obtained from ten different folds. The average AUC of these ten runs was taken as a fitness metric. The higher, the better. Population size was set as 30 members (ten times the parameters number) and the max number of generation equal to 200.

Using such configuration, differential evolution was performed to select the best parameter combination for each of the 19 evaluated data sets.

All the computational experiments, charts and statistical analysis were carried out in the R language platform and specific libraries (packages) for each matter. The DE implementation employed in these experiments was DEoptim, a R package for global optimization by differential evolution [225], [226].

Experimental Results

The distinct combinations of the GGClassification's parameters were assessed via k-fold cross validation.Four strategies were experimented to be compared:

1. GGClassification's original configuration.
2. Grid search parameter combinations for **removing** the samples filtered.
3. Grid search parameter combinations for **shifting** the class of the samples filtered.
4. DE - differential evolution optimization. Best parameters combination found via DE.

The results will be presented below. Figure 4.39 shows AUC boxplots of each distinct configuration faceted by dataset.

Table 4.2 summarizes the distinct configurations average AUC in front all experimented data bases considering all threshold combinations.The last row brings the average Friedman rank [1] [227] of each model parameter selection (Grid - shift class, original , Grid - remove and DE). Friedman's test has showed that there is evidences of significant difference between DE parameter selection performance and the others. The same for Original parameter selection. In general, there is no evidences of significant difference between Grid Search - remove and Grid Search - class shift.

It is possible to see the parameter selection rank facet by data set besides theirs meta-data on Figure 4.44.

Each distinct combination ($TC^{+}vs.TC^{-}$) visited through grid search was assessed by k-fold cross validation. The grid result can be observed on Figure 4.40. It shows AUC heatmap resulting from the grid search ($TC^{+}vs.TC^{-}$) applying *remove* mode configuration. Figure 4.41 shows the same heatmap to *class shift* mode configuration. Multiple comparisons between the distinct parameter selection methodologies were performed based on the k-fold cross validation results. Figure 4.42 shows Tukey's method results for the "model configuration" factor. The "model configuration" factor here can be "original method", "filter to remove patterns" and "filter to class shift". That, applied individually to all bench march data set. In the majority of cases DE optimization has selected the best parameters values. Figure 4.43 show the coordinates TC^{+},TC^{-} corresponding to the better results of each selection method (original, grid search remove and shift and DE optimization).

Discussion of Results

Grid search with k-fold cross validation results show that TC^{+} and TC^{-} overlapping thresholds has influence on GGClassification performance. That effect occurs with varied intensities for each experimented data set, as it is possible to see in boxplots (4.39) and heatmaps (4.41 and 4.40).

For hardly imbalance data sets, some of TC^{+} and TC^{-} pairs can promote a large exclusion rate of the minority class. This can become a problem, since there may not be any remaining patterns of such a class.

[1] Friedman's test is a non-parametric statistical test used to detect differences in treatments across multiple test attempts. It creates a classifiers rank for each data set. The average rank is used to formulate the hypothesis $H0$ of "All experimented classifiers are equivalent".Basically, the comparative procedure initially involves ranking each row (data set), and calculate the average rank for each column (classifier, "factor").Multiple comparison between the averages ranks promotes p-values that can indicate evidences of significant difference between the classifiers.

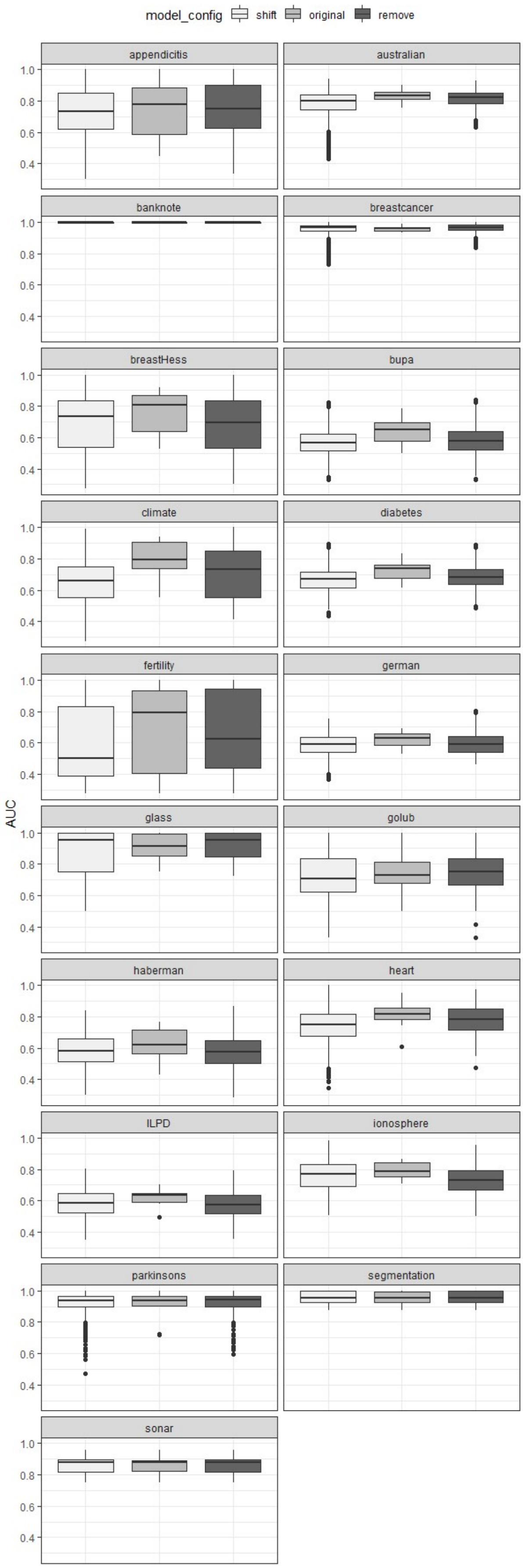


Figure 4.39: Model configuration AUC in front bench march data sets.

	base	shift	original	remove	DE
1	ILPD	0.59	0.62	0.58	0.69
2	appendicitis	0.73	0.75	0.74	0.83
3	australian	0.79	0.83	0.81	0.85
4	banknote	1.00	1.00	1.00	1.00
5	breastHess	0.70	0.76	0.69	0.83
6	breastcancer	0.95	0.96	0.96	0.99
7	bupa	0.57	0.64	0.58	0.67
8	climate	0.66	0.80	0.72	0.75
9	diabetes	0.67	0.72	0.68	0.76
10	fertility	0.61	0.68	0.66	0.82
11	german	0.59	0.62	0.59	0.69
12	glass	0.89	0.90	0.90	0.97
13	golub	0.72	0.74	0.74	0.85
14	haberman	0.59	0.62	0.58	0.64
15	heart	0.74	0.81	0.78	0.82
16	ionosphere	0.76	0.79	0.74	0.84
17	parkinsons	0.90	0.91	0.91	0.94
18	segmentation	0.95	0.95	0.96	0.96
19	sonar	0.86	0.86	0.86	0.89
Rank		3.71	2.10	3.07	1.10

Table 4.2: Model configuration mean AUC - all threshold ($TC^{+} vs. TC^{-}$) combinations

Grid search indicates that depending on the data set to be learned, small thresholds range can promote considerable performance variation.

One can see in the multiple comparative analysis that the GGClassification original configuration has showed better performance than the most of the grid search selection space. Furthermore, it is possible to note that the "shift class" strategy is not efficient in the most of the cases.

As can be seen in Figure 4.43 the best TC^{+} and TC^{-} values found by grid search, DE optimization and original method did not always coincide. That occurred because grid search was performed by using intervals of 0.05, the original method, on the other hand, can find continuous values obtained from the Q descriptor average. DE results are also continuous, however, it can traverse the entire search space. Thus, patterns coordinates (TC^{+},TC^{-}) provided by the three schemes may not match.

DE Optimization has obtained the best parameters selection for the majority of data sets, but spending a huge computational effort. It takes hours to perform completely the parameter search for each data set, considering 30 members population and max generation 100. Two of the data sets (bank note and german) spent almost 3 days, running on an AWS EC2 t2.medium machine (Linux OS, 3.3 GHz processor, 4 GB RAM).

Figure 4.44 shows the parameter selection rank and detailed metadata of each benchmark data set adopt in the present work. N means number of patterns (samples, observations), N_d the number of dimensions (attributes, columns), NC^{+} number of patterns labeled as (+1), and NC^{-} the number of (-1) labeled patterns.

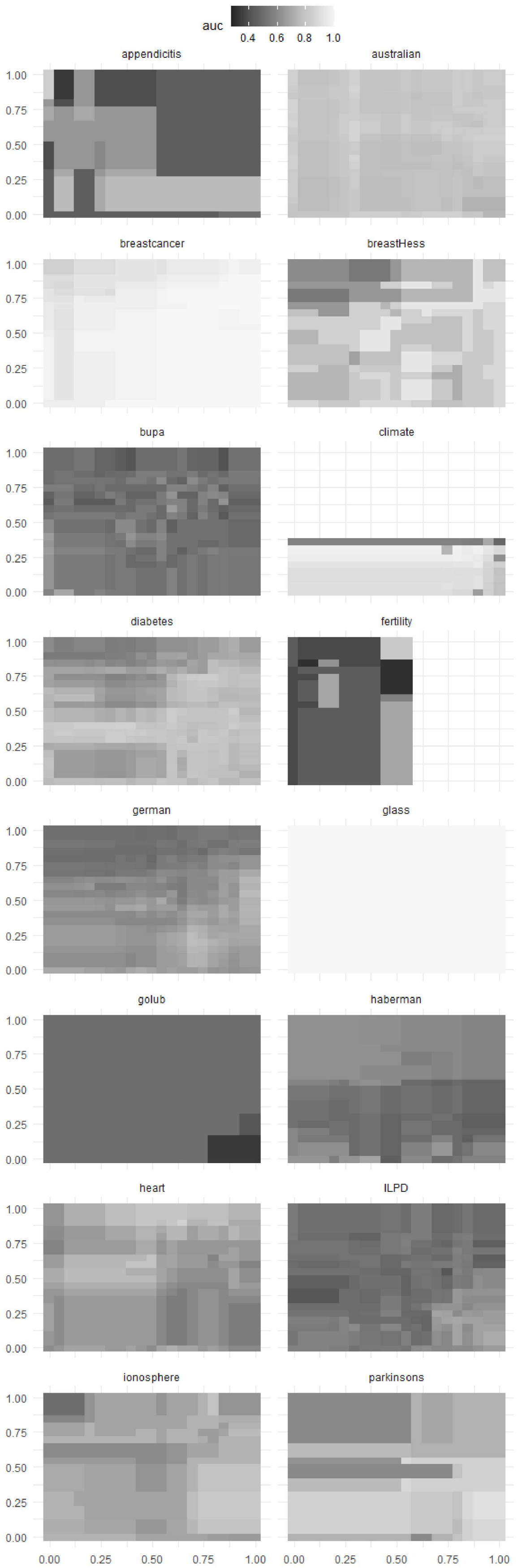


Figure 4.40: $TC^{+}vs.TC^{-}$ Mean AUC Heatmap - "remove" model configuration. The clear the better.

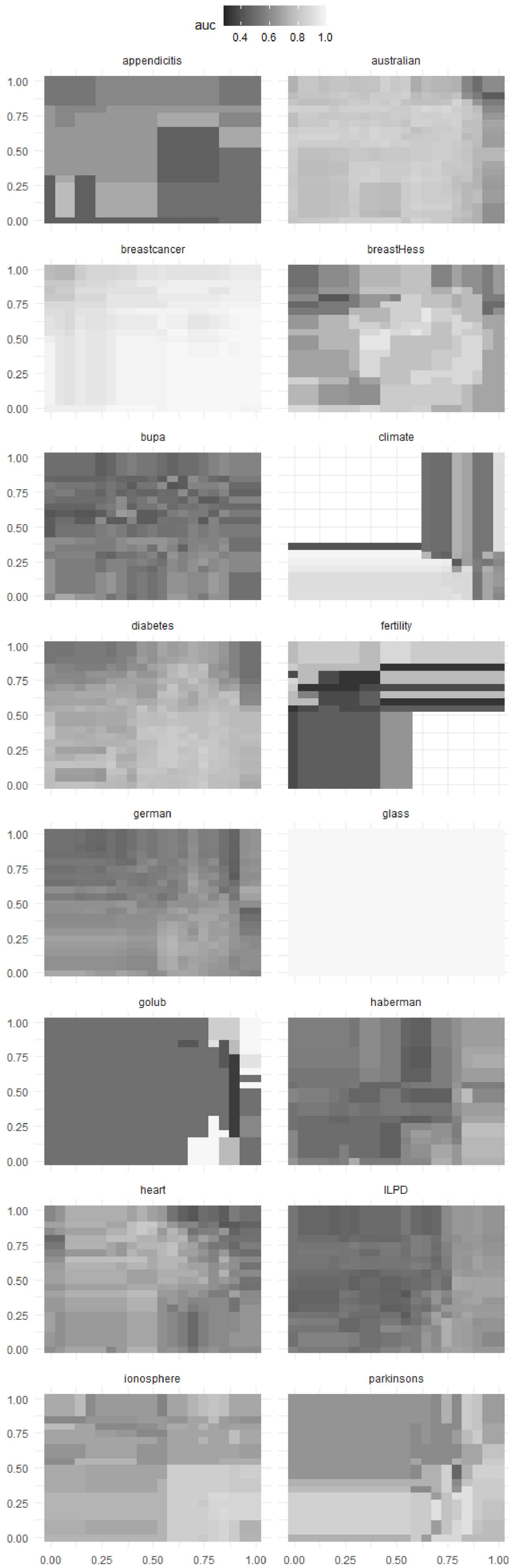


Figure 4.41: $TC^{+}vs.TC^{-}$ Mean AUC Heatmap - "Class shift" model configuration. The clear the better.

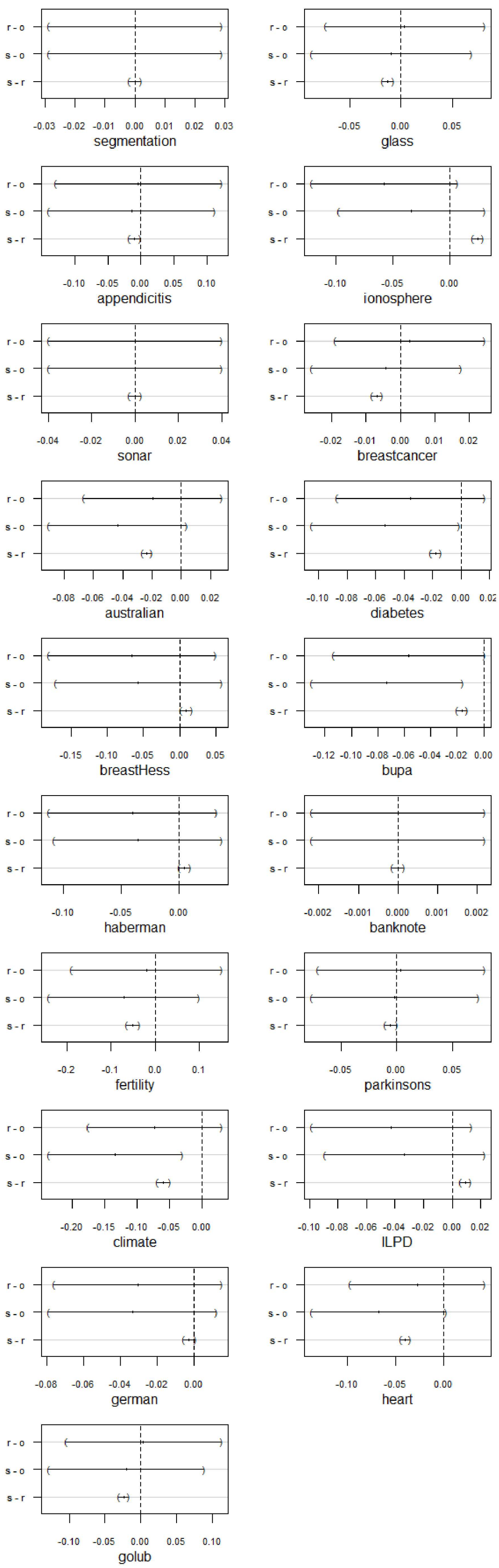


Figure 4.42: Multiple comparisons: all vs all configuration models comparisons using Tukey's method. r: remove; o: original: s:shift

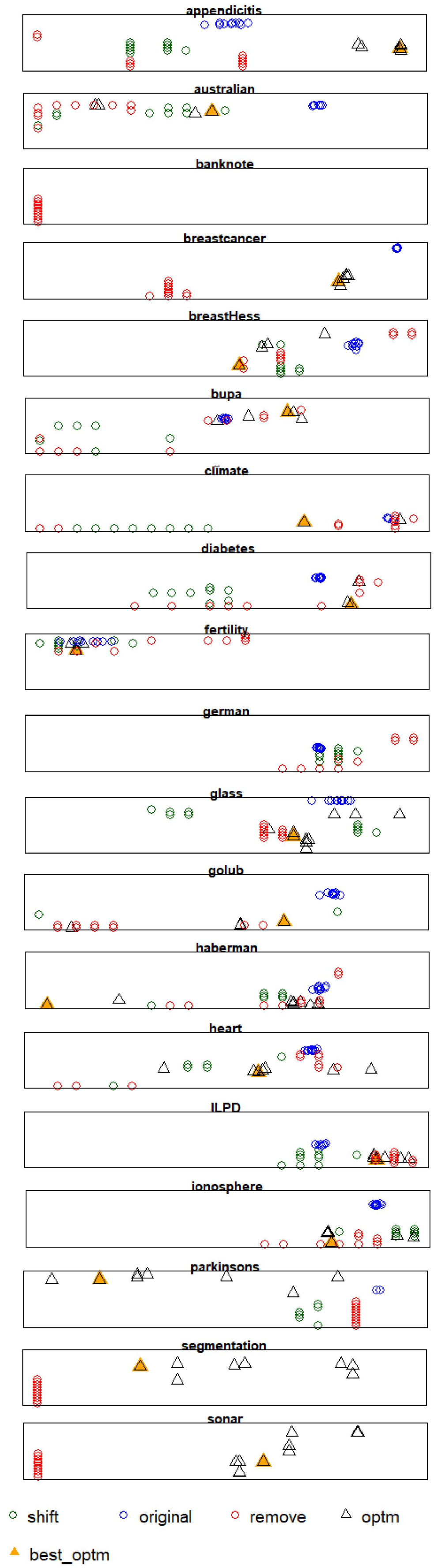


Figure 4.43: Parameter selection comparisons

	base	sht_AUC	org_AUC	rem_AUC	DE_AUC	sht_rank	org_rank	rem_rank	DE_rank	base	N	Nd	NC^{+}	NC^{-}
1	ILPD	0.59	0.62	0.58	0.68	3.00	2.00	4.00	1.00	ILPD	579	10	165	414
2	appendicitis	0.73	0.75	0.74	0.83	4.00	2.00	3.00	1.00	appendicitis	106	7	85	21
3	australian	0.79	0.83	0.81	0.85	4.00	2.00	3.00	1.00	australian	690	14	383	307
4	banknote	1.00	1.00	1.00	1.00	3.50	2.00	3.50	1.00	banknote	1372	4	762	610
5	breastHess	0.70	0.76	0.69	0.83	3.00	2.00	4.00	1.00	breastHess	133	30	34	99
6	breastcancer	0.95	0.96	0.96	0.99	4.00	3.00	2.00	1.00	breastcancer	683	9	239	444
7	bupa	0.57	0.64	0.58	0.64	4.00	1.00	3.00	2.00	bupa	345	6	200	145
8	climate	0.66	0.80	0.72	0.75	4.00	1.00	3.00	2.00	climate	540	18	46	494
9	diabetes	0.67	0.72	0.68	0.76	4.00	2.00	3.00	1.00	diabetes	768	8	268	500
10	fertility	0.61	0.68	0.66	0.82	4.00	2.00	3.00	1.00	fertility	100	9	88	12
11	german	0.59	0.62	0.59	0.69	4.00	2.00	3.00	1.00	german	1000	24	300	700
12	glass	0.89	0.90	0.90	0.97	4.00	3.00	2.00	1.00	glass	214	9	185	29
13	golub	0.72	0.74	0.74	0.85	4.00	3.00	2.00	1.00	golub	72	50	25	47
14	haberman	0.59	0.62	0.58	0.64	3.00	2.00	4.00	1.00	haberman	306	3	81	225
15	heart	0.74	0.81	0.78	0.82	4.00	2.00	3.00	1.00	heart	270	13	120	150
16	ionosphere	0.76	0.79	0.74	0.84	3.00	2.00	4.00	1.00	ionosphere	351	33	126	225
17	parkinsons	0.90	0.91	0.91	0.94	4.00	3.00	2.00	1.00	parkinsons	195	22	48	147
18	segmentation	0.95	0.95	0.95	0.95	3.50	2.00	3.50	1.00	segmentation	210	18	180	30
19	sonar	0.86	0.86	0.86	0.88	3.50	2.00	3.50	1.00	sonar	208	60	111	97

Figure 4.44: Detailed Table - Parameter selection methods performance, rank and bases meta-data

Chapter 5

Methodology

The new proposed method presented by the present work aims to generate counterfactual examples via geometric and statistical data structures. The concept involves the creation of interpretable counterfactual explanations by identifying the minimal changes needed in a sample[1] to alter its classification by a model. The research proposes using geometric structures, that map the similarity between samples, to generate counterfactual examples in classification problems. Accord to the new proposed method, the counterfactual examples are generated based on three principal elements:

- the similarity mapping provided by the geometric structure.
- the classification label (actual from the dataset or predicted by the operational trained model).
- users' requirements (use case constraints).

One significant advantage of employing similarity-based geometric structures, such as the Gabriel Graph, for generating counterfactual instances is the precision and efficiency in producing representative, plausible and diverse examples. Unlike other methods that demand substantial computational resources, the Gabriel Graph relies solely on distance calculations, reducing time and computational effort. This approach maintains the overall structure of the original dataset, facilitating the generation of meaningful and interpretable results by accurately reflecting the similarities and differences among the samples.

Furthermore, the Gabriel Graph structure has a built-in mechanism that can be employed to control the distance between the original and the generated counterfactual examples, allowing users to fine-tune the results according to their specific needs. These features make it a valuable tool for engineers, data scientists, and other researchers working on diverse datasets, from health care to industrial applications.

[1]The terms "samples", "examples", "lines in dataset", "line in dataframes", "register or line in a database", "instances", "patterns", "tuples", "records", "observations", "points" and "data points" are used interchangeably across various fields, though each term is more prevalent in specific contexts. "Examples", "patterns" and "instances" are common in Machine Learning, including Pattern Recognition and Artificial Intelligence. "Samples" and "observations" are widely used in Statistics. "Lines in datasets" and "line in dataframes" are primarily encountered in Data Science, especially in R and Python/Pandas programming. "Register" or "line in a database", and "record" are typical in Database Management and Information Systems. "Tuples" are frequently used in relational databases theory and formal data modeling. "Data points" can be found in several disciplines, such as Mathematics and Data Analysis. Despite these differences in terminology, they all refer to individual rows, entries, or units of information in a structured data format. In the present work they are accept as synonymous.

In this sense, the main idea of the proposed method is to explore counterfactual scenarios via similarity-based geometric structures. This way, facilitating a deeper understanding of the input data and model's behavior, providing information about decision boundaries, model vulnerabilities, training data biases and generalization issues.

The similarity mapping allows, in addition, identifying the most representative samples (examples) in the input dataset. The characteristics of such reference samples can indicate ways of instantiate new samples according to the user requisition.

That approach of directly access the samples' geometric information can reduce the computational complexity of the instances generation procedures and improve the quality of the generated counterfactual examples. The geometric structure representing the data space and its relevant points can directly guide the generation of the new samples. The present work proposes various strategies to guide the generation of counterfactual examples. Broadly speaking, it is about five fundamental steps:

1. Generating geometric data structures.
2. Using the geometric data structures to classify.
3. Choosing the reference points on the other side of the frontier.
4. Instantiating new points based on the references points.
5. Choosing some of the new instantiated points.

5.1 Generating geometric data structures

The basic and most important data structured used in the proposed method is based on GGClassification. One can see details about it in section *4.3:Torres at all's GGClassification*.

The first step consists in generating the Gabriel graph. The inputs consist of the factual dataset and a distance metric. The output is the adjacency matrix ($AdjM$) correspondent to the Gabriel graph.

The $AdjM$ is square,nxn, where n is the number of samples in the input dataset. The vertices of the Gabriel graph corresponds to the samples of the dataset. Each entry $AdjM[i][j]$ indicates whether there is an edge between the vertex i and vertex j. If $AdjM[i][j] = 1$ then there is an edge between i and j, while if $AdjM[i][j] = 0$ there are no connection. The Gabriel graph connects two points, $AdjM[i][j] = 1$, if the n-ball of diameter $\overline{ij}$ contains no other points. So, if a "1" appears at the $[i, j]$ index of the $AdjM$ matrix, it means that there is no other factual example inside the n-ball defined by the samples i and j.

The second step is about filtering. One can decide to filter factual samples, based on some criteria, aiming to suppress eventual noise or class overlapping in the dataset. The filtering process can be driven by samples quality scores and thresholds generated from the samples geometric information, labels and neighborhood (for details, see section 4.3). After any filtering, the Gabriel graph represented by its adjacency matrix is generated again from the new "clean" dataset.

The third step aims to generate all the remaining geometric structures needed to guide the creation of counterfactual examples. It start from obtaining the Support Edges (SE) from $AdjM$. SEs are those edges with vertices i and j that belong to different classes. They are located in the separation margin between classes [191, 193]. Each SE in $AdjM$ generates the following elements:

- *endpoints c and d*: the vertices i and j in $AdjM$ that form a SE are taken as the endpoints c and d.
- *middle point m*: the middle point of the SE is so calculated as $m = (c + d)/2$.
- *hyperplane normal vector w*: $w = c - d$. [1]
- *bias b*: $b = m \cdot w$. [2]
- *the hyperplane's signal s*: The point c's signal is attributed to the hyperplane w. The signal of the hyperplane is the same of the point c.

The steps discussed above generate several pieces of information that support the creation and classification (*predict*) of the counterfactual examples. One can see the available information in the following list:

- **Coordinates of the factual examples**
  - Coordinates (characteristics) of the factual examples in the dataset.
- **Gabriel Graph Properties**
  - Adjacency matrix of the Gabriel graph.
  - Dataset and graph balancing properties:
    - Number of class 1 examples.
    - Number of class 2 examples.
    - Number of class 1 examples that are connected to another class 1.
    - Number of class 2 examples that are connected to another class 2.
- **Quality and Statistics Metrics of each sample**
  - Number of neighbors of the same class.
  - Number of neighbors of the opposite class.
  - Total number of neighbors.
  - Purity metric.
  - Class 1 purity threshold.
  - Class 2 purity threshold.
- **Filtered Examples and Graph Properties**
  - Filtered examples.

[1] The vector w represents the weights (or coefficients) that define the orientation (angle of inclination) of the hyperplane.The vector w connects c and d that belongs to opposite classes. This vector is used to define the disposition of the local separation, ensuring that the separating hyperplane is perpendicular to the support edge. By constructing the hyperplane perpendicular to the edge (represented by $c - d$), an optimal separation of the classes is achieved. The vector w and the hyperplane normal to it provide all the necessary geometric information for distinguishing the classes in their vicinity.

[2] The bias determines the position of the hyperplane in relation to the origin of the input space (characteristics). The bias shifts the hyperplane, allowing it to pass through different regions of the space. The dot product between the midpoint m and the vector w ($b = m \cdot w$) results in a value that positions the hyperplane passing by m. In its time, w, defines the inclination (angle), while bias b defines the position within the space.

- Density.
- Support edges.
- Support edge endpoints.
- Support edge midpoints.

- **Hyperplanes Related to Supporting Edges**
  - Coefficients of the hyperplanes perpendicular to the support edges.
  - Bias of the hyperplanes perpendicular to the support edges.
  - Sign of the hyperplanes perpendicular to the support edges.

5.2 Using the geometric data structures to classify examples

With a comprehensive arsenal of statistical and geometric data, one can classify and guide the creation of new examples based on factual space disposition. The classification process, roughly speaking, works as a mixtures of local experts [228, 229].

Assuming that the data can be effectively described by a set of functions, each defined over a specific local region of the input space, a modular architecture can model this data by assigning different modules to different regions of the space [229]. For the proposed method, each hyperplane w, defined by one distinct SE, represents one specific module (or a distinct *expert*).

The classification process of a given sample x begins by classifying the sample according to all hyperplanes individually. Each hyperplane, generated from the midpoints of the boundary edges, gives its "opinion" regarding the classification of the sample x. This way, each hyperplane can classify the sample x individually.

The classification of the sample x provided by a hyperplane n is determined through its equation $H_n(x) = x^T w_n - b_n$. The $H_n(x)$ result value is subsequently smoothed using a hyperbolic tangent function: $th_n = \tanh(H_n(x))$, which *sign*, $\mathrm{sgn}(th_n)$, represents the prediction made by the hyperplane n.

The individual predictions are then forwarded to a mediator that assigns weights to them. The mediator weighs the opinions and produces a final classification decision. Each hyperplane is assumed to yield optimal performance within its local neighborhood. It is an expert on its own vicinity. The mediator weights the predictions accordingly to such assumption and generates the final output. The Figure 5.1 aims to visually explain such a process.

The process works as a *system of experts and gating networks* [228]. The parameters of each local expert H_n depend only on the frontier vertices c_n and d_n and can be obtained by directly calculation [191], while the weighting parameters Pb is given by the mediator playing a role of gating network [228, 230].

One can interpret the gating network from a probabilistic perspective, where it works as a classifier. It maps a given input x into the probabilities that the various experts will be able to generate the desired output to it [229]. So, each sample x induces a set of probabilities P. Each probability Pb_i in P refers to a specific hyperplane H_i. The idea is that the higher the probability Pb_i, the higher the chance that the corresponding hyperplane H_i is correct in classifying the sample x.

The weighting parameter Pb_i of an expert hyperplane H_i when classifying an example x is calculated according to the equation 5.1, called *Softmax*. It is an alternative to squared

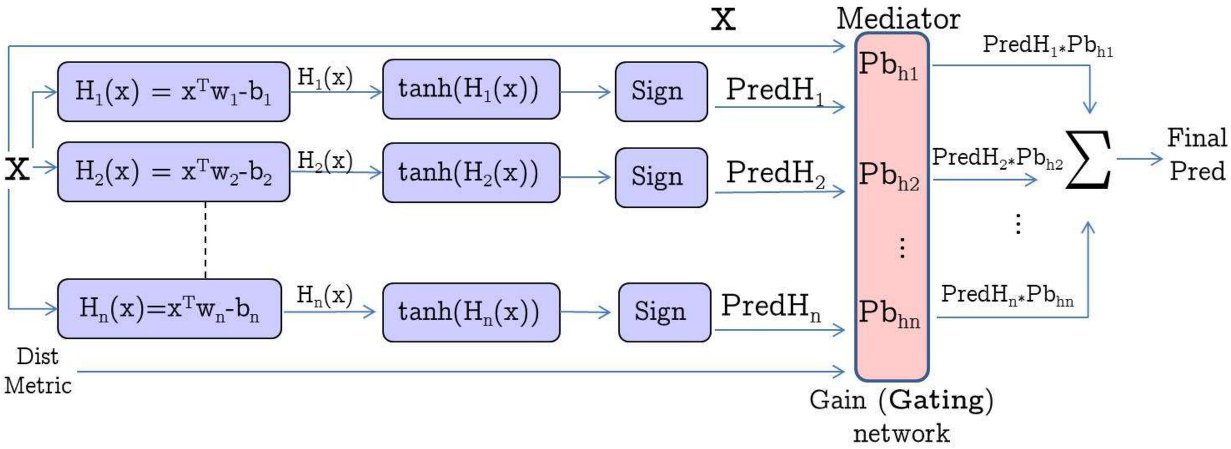


Figure 5.1: Local Experts Mixture.

error minimization, and implement a normalized exponential multi-input generalization of the logistic nonlinearity [231].

$$Pb_i(H_i, \mathbf{x}) = \exp\left(-\frac{\max\left[d(\mathbf{x}, m_k)\right]^2}{d(\mathbf{x}, m_i)}\right), \quad \forall k = 1, \ldots, n \tag{5.1}$$

This function preserves the hierarchical order of its input values and is a differentiable generalization of the "winner takes all" operation for choosing the maximum value [230].

Another interesting feature of Softmax is that it meets the probability constraints of being non-negative and having a sum equal to one. In formal terms, $0 < Pb_i < 1, \forall i = 1, \ldots, n$ and $\sum_{i=1}^{n} Pb_i = 1$, where n in the number of hyperplanes.

The final value is then calculated as $\text{sign}\left(\sum_{i=1}^{n} h_i(\mathbf{x})\, Pb_i(\mathbf{x})\right)$.The probabilities $Pb_i(x)$, which are combined with the respective hyperplane decision $H_i(x)$, give larger values to those hyperplanes that are closer to x according to the distance metric d. Figure 5.2 exemplifies the weights attribution with a 2D spacial perspective. As commented before, the proposed method also counts with signed values of those weights. This metric revels which class label and the respective intensity (importance) correspondent to each individual hyperplane (expert in the mixture). Figure 5.3 exemplifies the weights attribution with a 2D spacial perspective.

The sample x classification process also supplies the following information to drive the generation of counterfactuals and to help understanding the classification behavior:

- **Yhat:** Final classification of a given sample
- **yraw:** raw classification value from the Softmax sum.
- **YSoftMax**: The vector of absolute weights. One value for each expert (hyperplane).
- **YSignedSoftmax**: Signed values of the weights.
- **dist2pms:** Vector of distances (between the sample to each middle point that represents each hyperplane).
- **decishyper:** Classification that each individual hyperplane gave to the sample, based on its own equation.

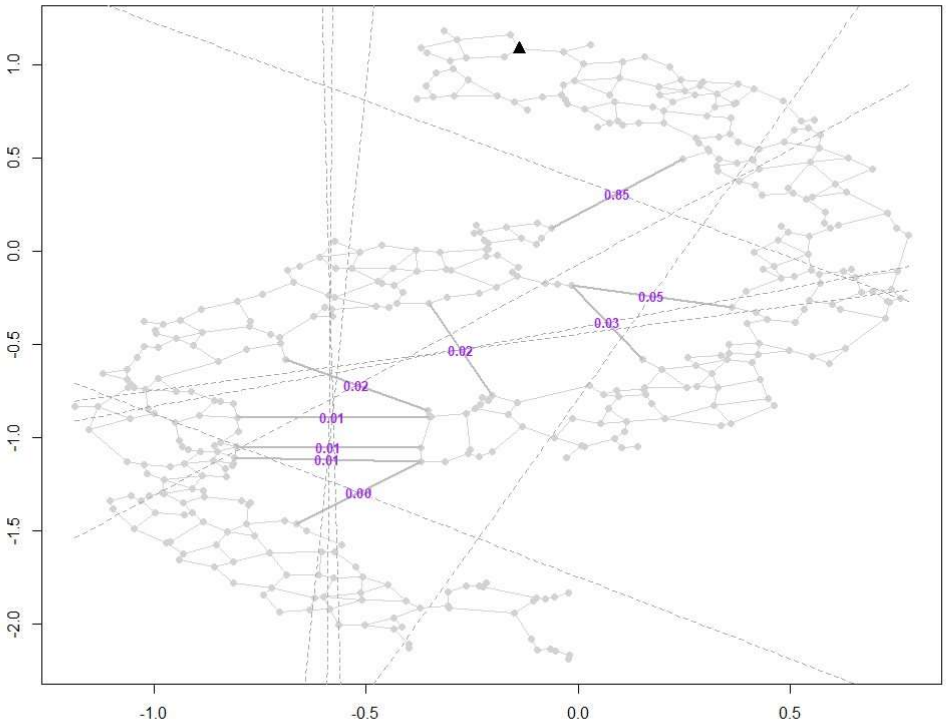


Figure 5.2: Softmax weighting result for a given sample. Dataset Two moons. Hyperplanes: gray dashes. SE: gray continuous lines. Sample x: Black triangle. Weights of each hyperplane: purple values.

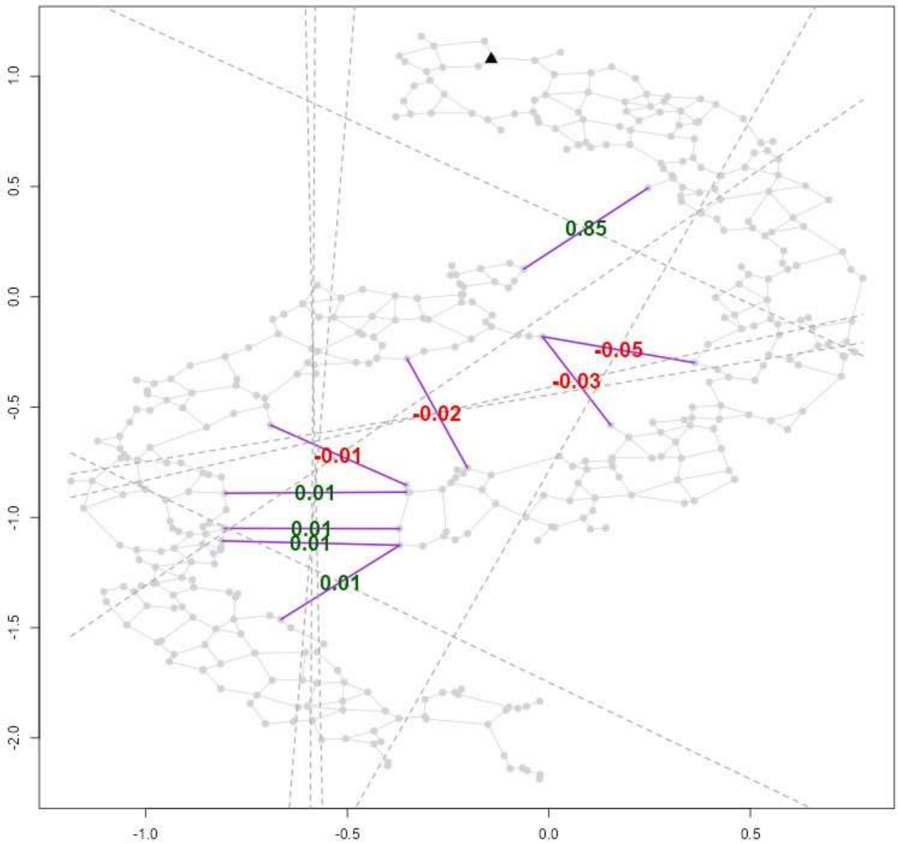


Figure 5.3: Weighted Softmax indicating the result for a given sample. Dataset: Two moons. Hyperplanes: gray dashes. SE: purple. Hyperplanes: gray dashed lines. Sample x: Black triangle. Classification as +1 class: green. Classification as -1: red. Values (numbers): weights (given in function of the distance from x).

- **deciswgt:** Weighted-distance classification of each hyperplane.

The hyperplanes weights combined to their parameters w, can generate a features importance chart, and in each sense (+ or -) each feature is contributing. Figure 5.4 shows an example: It is possible to follow the discussion between the experts through the *weighted classification*

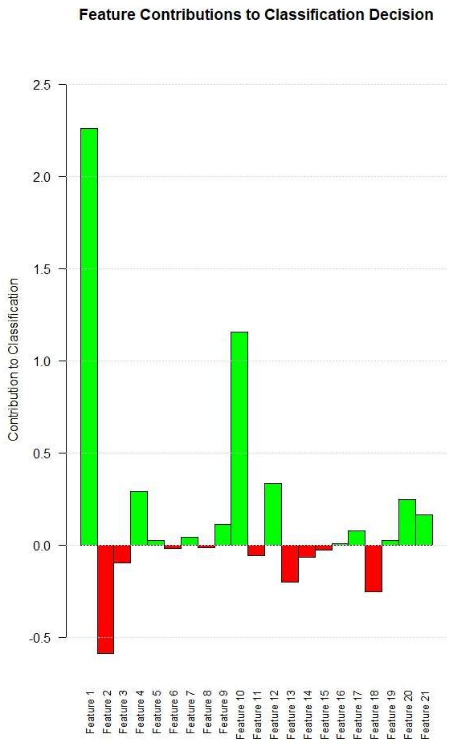


Figure 5.4: Force and feature importance plot calculated from the data generated by the classification process.)

voting. Figure 5.5 shows two distinct classification process from two different datasets.

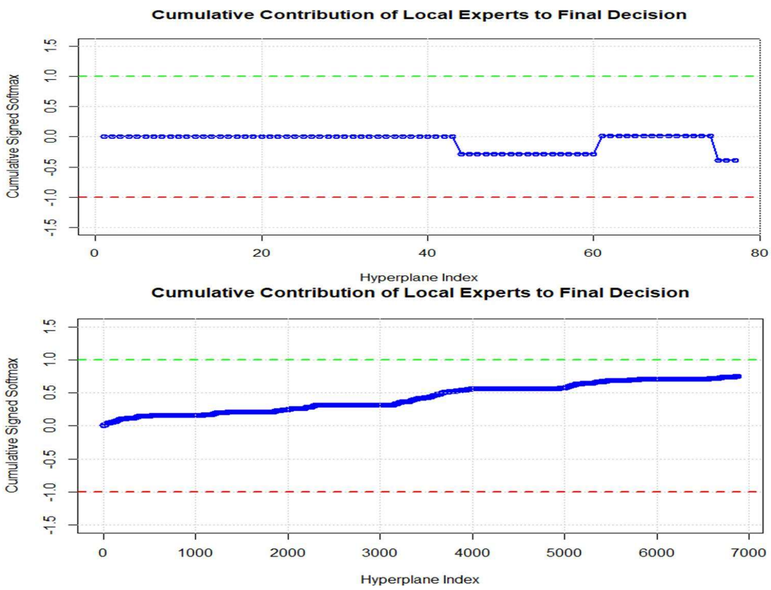


Figure 5.5: Force and feature importance plot calculated from the data generated by the classification process.)

5.3 Instantiating new samples based on the references

Some relevant elements can be raised when analyzing the geometric structures and the classification process from the perspective of counterfactual examples:

- **Points and edges metrics:**
  - Statistical and quality (purity) metrics of samples allow targeting reference points on both sides of the border.
  - Such metrics can be used by cost functions or counterfactual selection criteria.
- **Midpoints and Support Edges as Decision Boundaries:**
  - The edges connecting samples of opposite classes represent the critical transitions between classes. Edges indicate factual direct paths for class change.
  - The midpoints of these edges are pivotal. They lie exactly between factual opposing class samples and are used to generate separation hyperplanes.
  - The distance to the midpoints can parameterize functions and heuristics of counterfactual examples generating.
- **Separation Hyperplanes:**
  - The hyperplanes represent the decision boundaries inferred from the own data or from the model (under investigation) output space.
  - Hyperplanes are locals deciders experts. Prediction in their immediate vicinity have high accuracy.
- **Classification Using Mixture of Experts/Softmax:**
  - It is possible to weigh the influence of each hyperplane on the classification continuously through all the space. Softmax provides a vector of probability values, one for each hyperplane.
  - "Reversing the Mixture of Expert" permits to input a target class expecting a "x" sample in the output.

Several strategies for generating counterfactual samples are proposed based on these principles of creating a new counterfactual example x_{fc} by implement a change Δ on a reference point x_{ac}. Figure 5.6 shows the strategies proposed grouped according to the items previously commented.

These primary methods suggested are summarized in Table 5.1.

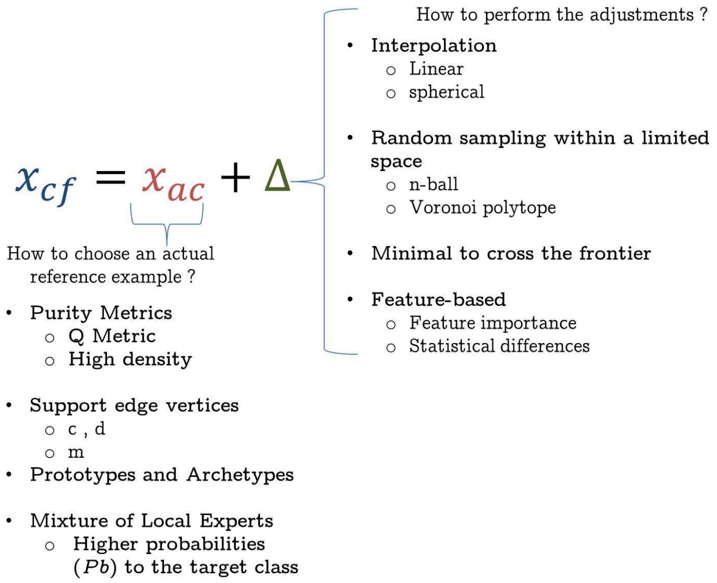


Figure 5.6: Strategies for generation of counterfactual samples.

Method	**Details**
1. Same-Class Edges	**Brief Explanation:** Generate new samples by interpolating between endpoints of edges connecting samples of the *same class*. **Formula:** $$x_{cf} = (1-\alpha)c + \alpha d, \quad \alpha \in [0,1]$$ **Explanation of Variables:** • x_{cf}: Counterfactual sample. • c, d: Endpoints of the edge within the same class. • α: Interpolation factor, controlling the position along the edge between c and d. $if \quad \alpha = 0 \quad then \quad x_{cf} = c$. $if \quad \alpha = 1 \quad then \quad x_{cf} = d$. The α domain permits to run along the entire edge between c and d. **Pros:** • Explores within-class variations; maintains class properties; it can be in high density or purity areas. **Cons:** • Limited to existing points regions.

Continued on next page

Method	Details
2. Opposite-Class Edges	**Brief Explanation:** Create counterfactuals by interpolating beyond support edges connecting *opposite classes*. **Formula:** $$x_{cf} = (1-\alpha)c + \alpha d, \quad \alpha \in [0,1]$$ **Explanation of Variables:** • x_{cf}: Counterfactual sample. • c, d: Endpoints of the edge connecting opposite classes actual samples (c and d). • α: Interpolation factor, allowing extrapolation to generate counterfactuals. $if \quad \alpha = 0 \quad then \quad x_{cf} = c$. $if \quad \alpha = 1 \quad then \quad x_{cf} = d$. The α domain permits to run along the entire edge between c and d. The α value adjustment permits to create counterfactuals closer to one side of the frontier or closer to the opposite side. That is , closer to c endpoint or closer to d. **Pros:** • Cost; slightly crosses the boundary, laying very close to the expert hyperplane. **Cons:** • It may fall in low density areas.

Continued on next page

<table>
<tr><th>Method</th><th>Details</th></tr>
<tr><td>3. Hypersphere Volume Sampling Centered at Reference Points</td><td>Brief Explanation: Generate samples within a hypersphere (hs) with radius r centered at a relevant point x_{ref}.

Formula:
$$x_{cf} = x_{\text{ref}} + r \cdot u, \quad \|u\| \leq 1$$
Explanation of Variables:
• x_{cf}: Counterfactual sample.
• x_{ref}: Center of the hypersphere (reference point).
• r: Radius of the hypersphere.
• u: Unit vector within the hypersphere, determining direction.

Pros:
• Creates diverse samples; explores local neighborhood.
• The radius r can vary according to some distances (between original samples and p, distance between p and the frontier.)

Cons:
• May include invalid regions; choice of r affects validity; curse of dimensionality (volumes calculation).</td></tr>
<tr><td colspan="2" align="right">Continued on next page</td></tr>
</table>

Method	Details
4. Hypersphere Volume from Same-Class Edges	**Brief Explanation:** Sample in the hypersphere defined by an edge connecting two same-class samples i and j (at some place on the opposite side of the boundary). **Formula:** $$x_{\text{ref}} = \frac{x_i + x_j}{2}, \quad r = \frac{d(x_i, x_j)}{2}, \quad x_{cf} = x_{\text{ref}} + r \cdot u, \quad \|u\| \leq 1$$ **Explanation of Variables:** • x_{ref}: Center of the hypersphere, midpoint of edge connecting x_i and x_j. • r: Radius of the hypersphere, half the distance between x_i and x_j. • u: Unit vector within the hypersphere. **Pros:** • Explores larger regions; aiming to keep class consistency. **Cons:** • Computational cost; r can affect validity; curse of dimensionality (volumes calculation).

Continued on next page

Method	Details
5. Utilizing Purity-Based Reference Points	**Brief Explanation:** Use high purity score samples (high percentage of same-class neighbors) as anchors to guide counterfactual generation. **Formula:** $$x_{cf} = x_{\text{ref}} + \Delta$$ **Explanation of Variables:** • x_{cf}: Counterfactual sample. • x_{ref}: High purity reference point. • Δ: Adjustment applied to the reference point. **Pros:** • For small Δ, it ensures counterfactuals are well within target class; enhances robustness. **Cons:** • For large Δ, it can lose accuracy; requires purity score calculation; limited by data sparsity.
6. Sampling In High-Density Regions	**Brief Explanation:** Move towards regions with higher probability density (pd) of the target class. **Formula:** $$x_{cf} = x_{ref} + \Delta$$ $$\text{Subject to:} \quad pd_{\text{target}}(x_{ref}) \geq \theta, \quad pd_{\text{target}}(x_{cf}) \geq \theta$$ **Variables:** • x_{cf}: Generated counterfactual sample. • x_{ref}: Initial reference point. • pd_{target}: Probability density of the target class. • θ: Minimum density threshold for valid counterfactuals.

Continued on next page

Method	Details
7. Convex Hull-Oriented	**Brief Explanation:** Project samples onto the convex hull of target class samples or use the hull's centroid as a reference. **Formula:** $$\text{Projection:} \quad x_{ref} = \min[d(x_{ac} - p)] \quad \forall p \in CH_{\text{tc}}$$ $$\text{Centroid:} \quad x_{ref} = \frac{1}{K}\sum_{i=1}^{K} x_i, \quad \forall x_i \in CH_{\text{tc}}$$ $$x_{cf} = x_{ref} + \Delta$$ **Variables:** • x_{cf}: Generated counterfactual sample. • x_{ac}: Original sample. • CH_{tc}: Convex hull of the target class. • p: Point within the convex hull. • Δ: Adjustment vector applied to the reference point.
8. Reference to Prototypes and Archetypes	**Brief Explanation:** Adjust samples towards prototypes or archetypes of the target class. **Formula:** $$x_{cf} = x_{ac} + \delta(x_{\text{prototype}} - x_{ac})$$ **Variables:** • x_{cf}: Generated counterfactual sample. • x_{ac}: Original sample. • $x_{\text{prototype}}$: Target class prototype or archetype. • δ: Adjustment factor determining the movement towards the prototype.

Continued on next page

Method	Details
9. Local Height-Importance Features	**Brief Explanation:** Modify the most influential features based on their geometric importance in experts' w. **Formula:** $$x_{cf,i} = x_{ac,i} + \delta x_i, \quad \delta x_i = -\epsilon \cdot \text{sign}(w_i)$$ **Variables:** • $x_{cf,i}$: Adjusted value of feature i. • $x_{ac,i}$: Original value of feature i. • w_i: Weight of feature i in the hyperplane. • ϵ: Adjustment step size.
10. Spherical Interpolation	**Brief Explanation:** Interpolate between samples along the surface of a hypersphere. **Formula:** $$x_{cf} = \frac{\sin((1-t)\theta)}{\sin\theta} c + \frac{\sin(t\theta)}{\sin\theta} d, \quad t \in [0,1]$$ **Variables:** • x_{cf}: Generated counterfactual sample. • c, d: Samples being interpolated. • θ: Angle between samples c and d. • t: Interpolation factor controlling the transition between c and d. The shape of the transitioning trajectory between c and d in this method case is spherical.

Continued on next page

Method	Details
11. Statistical Characteristic Adjustment	**Brief Explanation:** Adjust features based on statistical differences towards target class characteristics. **Formula:** $$x_{cf,i} = x_{ac,i} + \alpha(\mu_{\text{target},i} - x_{ac,i})$$ **Variables:** • $x_{cf,i}$: Adjusted value of feature i. • $x_{ac,i}$: Original value of feature i. • $\mu_{\text{target},i}$: Mean value of feature i in the target class. • α: Adjustment factor between 0 and 1.
12. Nearest Opposite Class	**Brief Explanation:** Generate counterfactuals by interpolating between x_{ac} and the nearest opposite-class sample. **Formula:** $$x_{ref} = \min(\mathbf{d}(x_{ac}, x^i_{\text{opcl}}))$$ $$x_{cf} = (1-\alpha)x_{ac} + \alpha x_{ref}, \quad \alpha \in [0,1]$$ **Variables:** • x_{cf}: Generated counterfactual sample. • x_{ac}: Actual original sample. • x_{ref}: Nearest opposite-class sample. • x^i_{opcl}: Any actual sample i of the opposite class. • α: Interpolation factor controlling the movement between x_{ac} and x_{ref}.

Table 5.1: Methods developed for generating counterfactual examples based on geometric and statistics data strictures.

Some figures are presented below to exemplify the geometric approach proposed for generating counterfactual examples. As it was commented before, the main idea is to select reference points based on specific criteria that align with the users' needs during the counterfactual investigation. Then, using these reference points, choose a sampling strategy to generate new samples (counterfactual examples). It is possible to use statistical and geometric information to guide the choice of the references points. It is also worth remembering that the proposed method's data structure still provides purity information and statistical metrics for each sample in the dataset.

Figure 5.7 shows the Gabriel graph with its edges in gray. The factual samples are plotted in two colors (red and green) according to their class label. The support edges (SE) that connect examples from opposite classes are in purple. The middle points of the SE are in blue dots. The hyperplanes instantiated from the SE's middle points in gray dashed lines. The division chain is shown in blue lines. The concave hulls of each class are in red and green lines according to each class. Such geometric structures can supply information to guide the localization of factual reference points and drive the generation of new counterfactual examples.

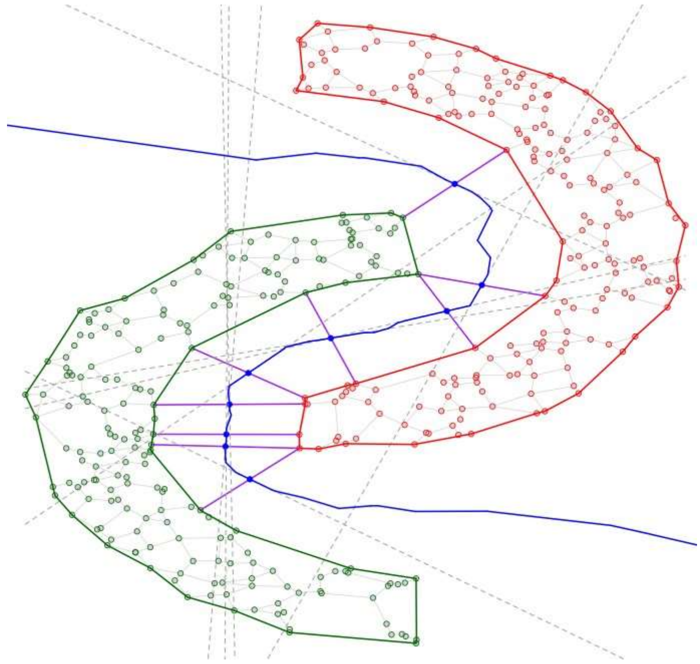


Figure 5.7: Examples of geometric structures that can guide the choice of references points.

Following the selection of reference points tailored to the specific requirements of the ongoing counterfactual investigation, on can proceed to the phase of generating the counterfactual examples. To do so, various strategies, including linear or spherical interpolations, regular or random sampling, and the modification of specific attributes, can be adopted. The Figure 5.8 shows an enlarged local view of the Gabriel Graph. This example

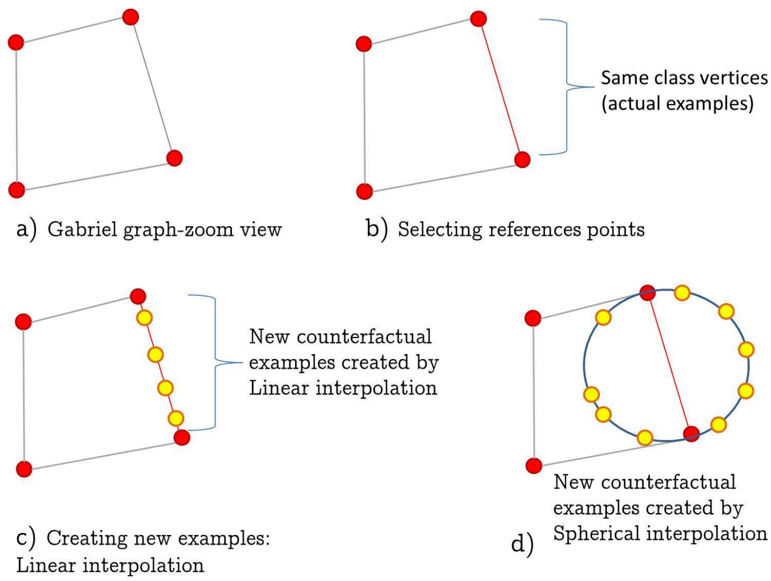


Figure 5.8: Generation of counterfactuals samples via linear and spherical interpolation.

shows the selection of an edge that connects two factual samples (vertices) of the same class.

After choosing the neighboring vertices, the Figure 5.8-c shows the generation of new samples (counterfactuals) through linear interpolation guided by the graph edge. The Figure 5.8-d exemplifies the generation of new counterfactual samples through spherical interpolation.

The Figure 5.9-a shows the generation of new samples (counterfactuals) through sampling in a spherical region. It can be regular sampling or random sampling within the space. Figure 5.9-b shows the generation of new (counterfactual) samples by linear interpolation between real samples of opposite classes. As it was previously pointed, it can adopt regular sampling or random sampling within the trajectory. On can se the vertices (actual samples) c and d belonging to distinct classes, the support edge connecting them (purple line) , the middle point m, the hyperplane w originated from the SE and the new counterfactual samples generated in the inter-class space from c to d.

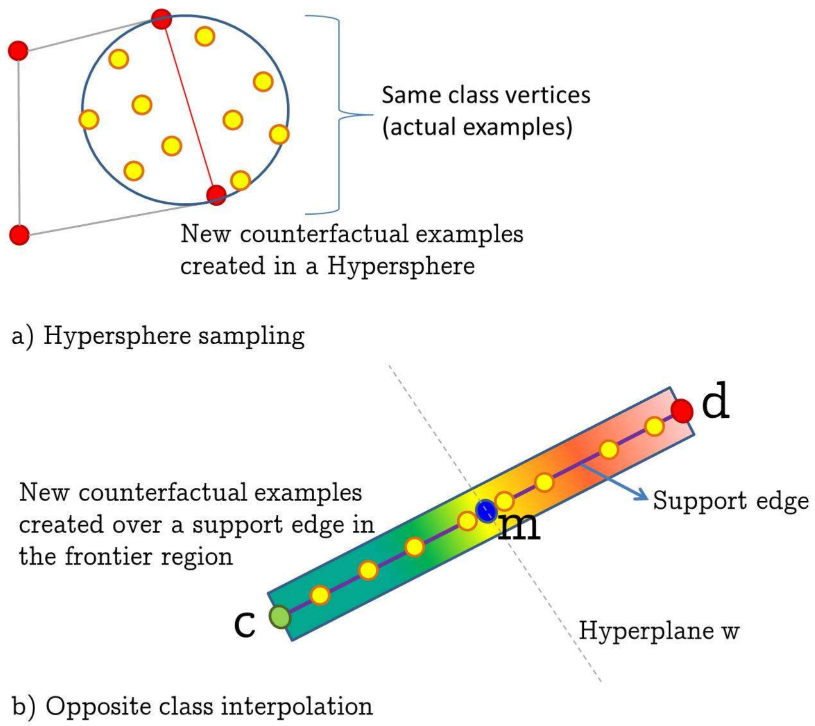


Figure 5.9: Generation of counterfactuals samples in the frontier region (inter-class space).

5.4 Validating and choosing samples based on business rules' constraints

Both the reference points and the evaluation of the generated counterfactual examples can utilize purity and cardinality indexes, statistical information, and various distance metrics calculations.

The Figure 5.10 shows a factual point (green fulfilled circle), a frontier contour (blue continuous line), support edges (purple continuous lines), their middle points (blue dots) and 4 counterfactual examples (orange, magenta, cyan and yellow). On can see that the orange sample is very close to the boundary middle point in a sparsely populated region. The magenta sample is located in a richly populated region in the opposite class to the green factual example. The cyan sample is at the external bound of the opposite class while the yellow sample resides in the shell that encloses the opposite class.

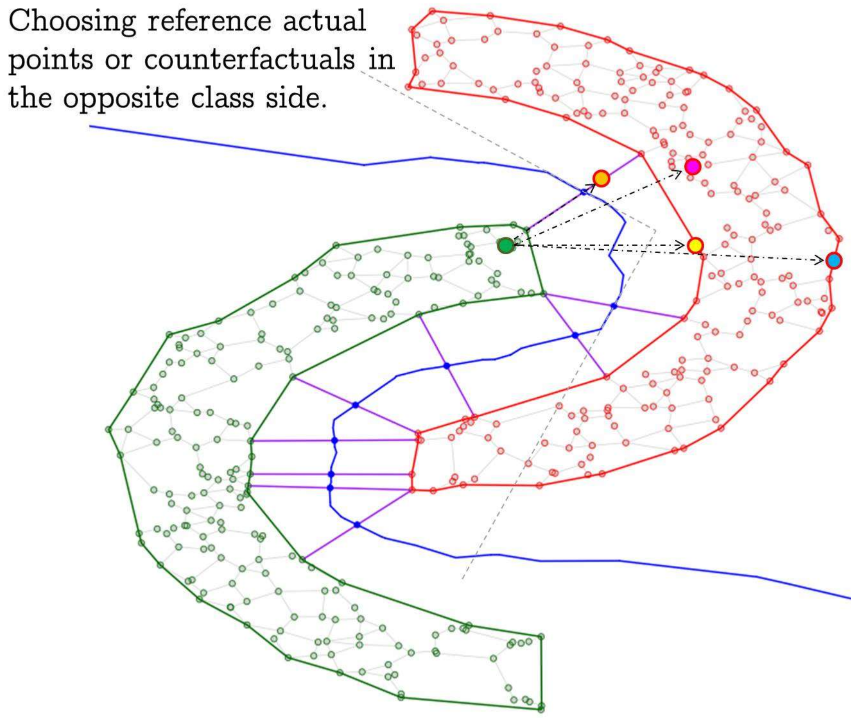


Figure 5.10: Generation of counterfactuals samples in the frontier region (inter-class space).

Given the variety of reference points for generating counterfactual examples and the diversity of the new samples produced, the proposed method offers various metrics to help users select the most suitable samples for their needs. For instance, the optimal counterfactual sample can be chosen by their purity and cardinality metrics or by using metrics such as the Manhattan distance (L1 norm), Euclidean distance (L2 norm), Chebyshev distance (L-infinity), Mahalanobis distance, among others besides the weighted combination of them.

Figures from 5.11 to exhibit the contours of different distance variations to illustrate the variation in distance metrics. It is important to note that the choice of distance metric induces distinct patterns of variation across the distinct space dimensions. Consequently, reference points and the generated counterfactuals can be selected based on these characteristics.

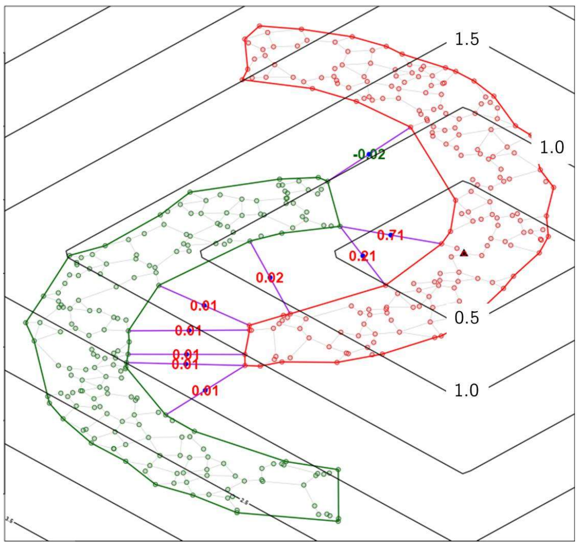


Figure 5.11: Behavior of the L1 Norm variation from a given point (black triangle).

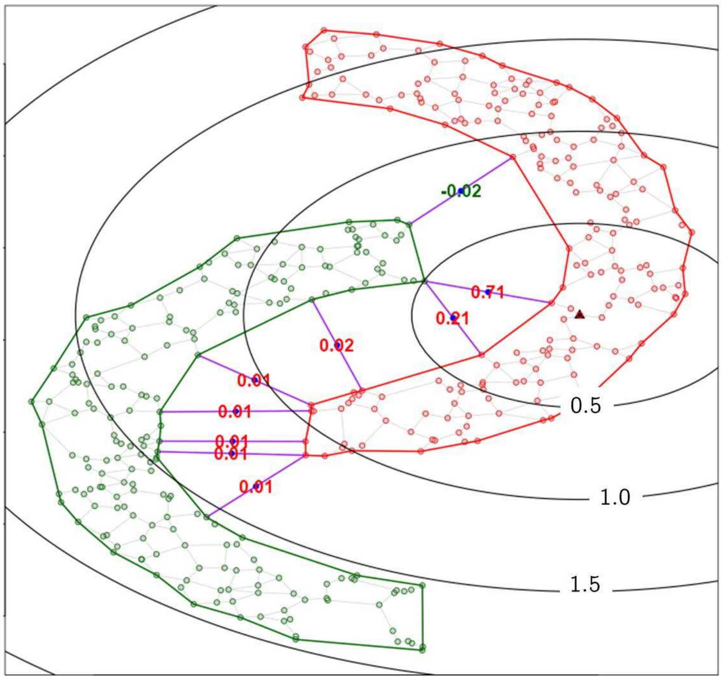


Figure 5.12: Behavior of the L2 Norm (euclidian) variation from a given point (black triangle).

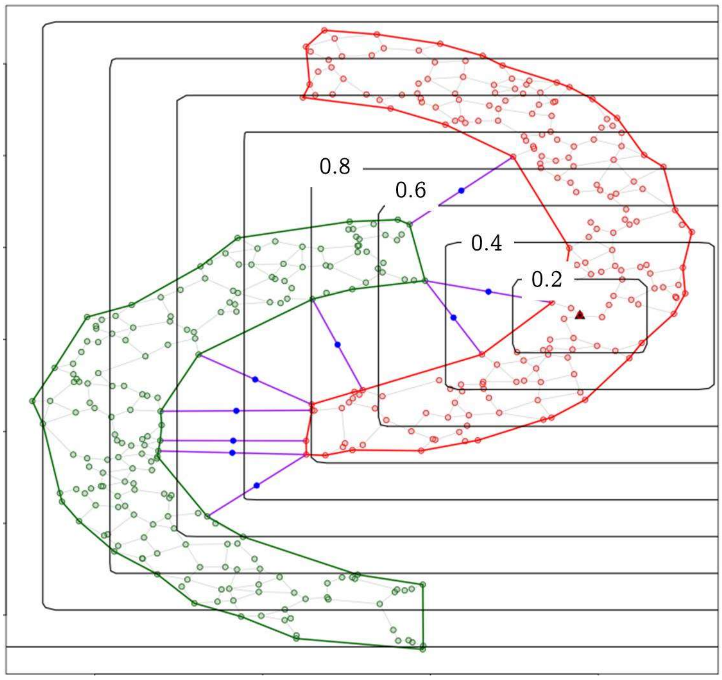


Figure 5.13: Behavior of the L-infinity Norm (Chebyshev) variation from a given point (black triangle).

The proposed method also makes possible to use weighted distance (cost) functions. It allows to use flexible combination of multiple distance norms, for instance L_1, L_2, L_∞, and Mahalanobis. This flexibility enables more precise control over distance measurement by assigning continuous-valued weights to each norm. This approach balances sensitivity and restrictiveness, capturing diverse aspects of the feature space in linear time.

This flexibility is particularly beneficial in counterfactual scenarios, where different data dimensions or definitions of "closeness" are important. Researchers can prioritize norms like Mahalanobis for distribution-sensitive similarity or L_∞, to deal with outliers or large

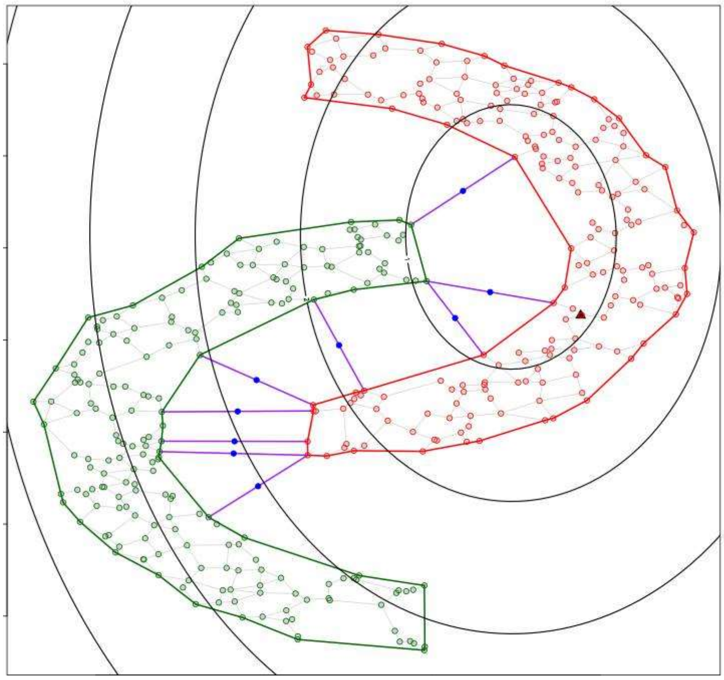


Figure 5.14: Behavior of the Mahalanobis distance variation from a given point (black triangle).

deviations, L_1 for interpretability and discover influential features. One can see a example of a customized distance (cost) function on Figure 5.15. This example showed the use of custom functions involving only distance metrics. However, in the same way, one can use the purity and cardinality metrics of the samples. The possibility of create flexible distance

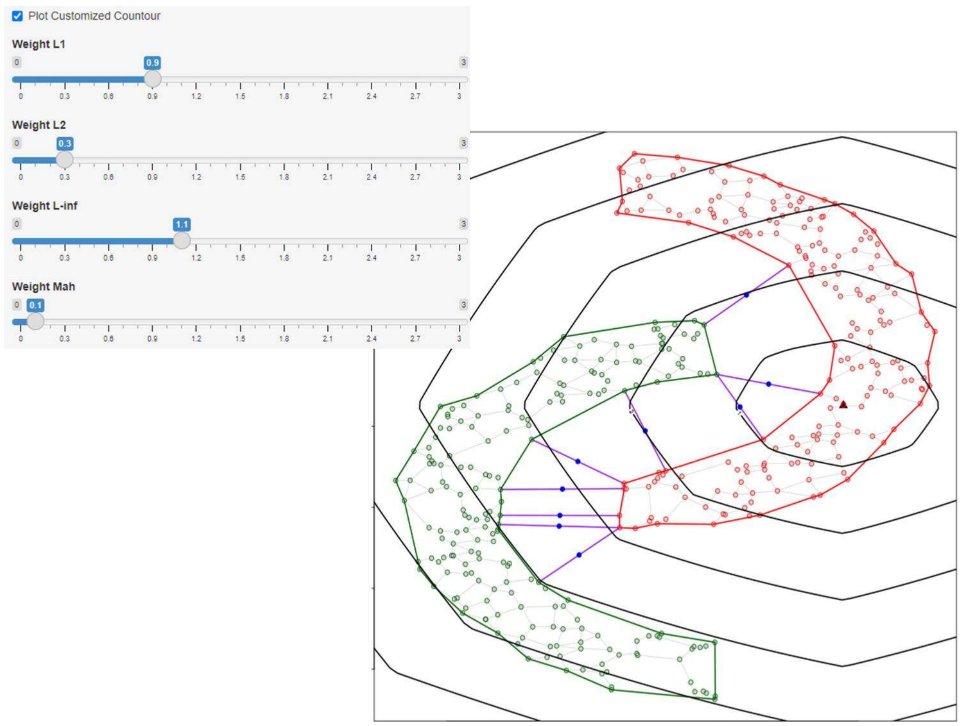


Figure 5.15: Behavior of a given customized distance metric variation from a given point (black triangle). Observe the different weights given to the different distance metrics.

and custom function ensures that counterfactual instances align with the users and business context's objectives and constraints.

Another functionality provided by the counterfactual examples generation method proposed here is compliance with specific business rules. The objective is to model business scenario restrictions related to variables (attributes, characteristics).

Business rules define specific constraints and guidelines within a business scenario. These rules can limit certain values of variables (characteristics) within a given context. It is common to find the following types of business constraints when modeling systems:

- **Immutable Features**: Features that cannot be changed (e.g. gender, birth place).
- **Bounded Features**: Features that have upper and/or lower limits (e.g., temperature, age).
- **Monotonic Relationships**: Features that should only increase or decrease.
- **Conditional Constraints**: Changes in one feature reflects on another (e.g., weight increases, BMI (Body Mass Index) must also increase).
- **Feasibility Constraints**: Only certain combinations of feature values are valid.

To accommodate such a business rules and some customized user-defined limitations, a constraints matrix is proposed. This matrix aims specify restrictions on each feature. including bounds, immutability, monotonicity, and conditional dependencies. It allows the user to dynamically impose relevant business constraints for each generation request. The constraints matrix structure is proposed as following:

Feature	ID	Type	Lower	Upper	Monot	Immut	Depend	Condition
Age	1	Bounded	18	65	none	true		
Credit Score	4	Bounded	300	850	increase	false		
Job Type	2	Categorical			none	false		
Weight	3	Conditional			increase	false	bmi	`bmi <- function(Weight, Height) Weight / (Height*Height)`

Table 5.2: Example Constraints Matrix for Counterfactual Generation

- **Feature Name**: The feature name. *char(30).*
- **Feature ID**: The feature ID (index in the dataset). *integer.*
- **Constraint Type**: Type of constraint, such as Bounded, Immutable, Monotonic, Categorical, or Conditional. *char(15)*
- **Lower Bound**: Applicable lower limit (min). *numeric*
- **Upper Bound**: Applicable upper limit (max). *numeric*
- **Monotonicity**: Specifies if the feature should only increase or decrease. *integer*
- **Immutable**: Indicates if the feature is fixed. *integer*
- **Dependency Feature** and **Dependency Condition**: Defines feature dependencies for Conditional constraints. *function object*

Table 5.2 brings an illustrative example of constraints matrix. Upon gathering the constraint matrix, the counterfactual generation procedure processes it by applying the constraints according to the constraint matrix:

- **Immutable Constraints**: Features marked as immutable remain fixed at their original values ($\alpha = 1$).
- **Bounded Constraints**: Bounded features are clamped (min or max) to their specified lower and upper bounds, when necessary.
- **Monotonicity Constraints**: Features with monotonic constraints are adjusted in the required direction only (min, max).
- **Conditional Constraints**: Adjustments to features are conditioned on related feature values.

5.5 Proposed Geometric - Statistical Methods and The Interpretability of Models' Behavior

The new methods for generating counterfactual examples can assist in gain deeper insights into model behavior. The present section aims to briefly outlines how these methods contribute to understanding and interpreting machine learning models outcomes.

5.5.1 Boundary Exploration and Decision Logic

- **Highlighting Decision Boundaries:** Methods such as *Opposite-Class Edges* and *Convex Hull-Oriented Sampling* directly explore decision boundaries, revealing the conditions under which the model changes its classification. This insight aids in understanding how the model perceives transitions between classes.
- **Identifying Critical Features:** Techniques like *Local Height-Importance Features* expose the key features and their weight contributions in the classification decision, enhancing transparency regarding the model's internal decision-making process.

5.5.2 Validity and Robustness Testing

- **Testing Robustness:** By generating counterfactuals in high-density or low-density regions (e.g., *Sampling In High-Density Regions*), one can analyze the model's consistency and robustness in various data environments.
- **Identifying Weaknesses:** Counterfactuals generated in sparse areas (e.g., *Nearest Opposite Class*) may uncover regions where the model lacks sufficient training or performs poorly, suggesting areas for improvement.

5.5.3 Interpretability and Plausibility

- **Enhanced Interpretability:** Methods like *Hypersphere Sampling Centered at Reference Points* create local counterfactuals, which are intuitive for end-users to understand and relate to their decision-making process.

- **Generating Plausible Scenarios:** Approaches such as *Utilizing Purity-Based Reference Points* ensure counterfactuals are realistic and lie within feasible regions of the data space, enhancing trust in the model.

5.5.4 Understanding Class Characteristics

- **Class-Specific Analysis:** Techniques like *Reference to Prototypes and Archetypes* and *Convex Hull-Oriented Sampling* enable the exploration of class-specific structures. These methods help in understanding what defines each class and how the model differentiates between them.

5.5.5 Diversity and Generalization Insights

- **Exploring Diverse Scenarios:** Techniques such as *Spherical Interpolation* and *Hypersphere Volume from Same-Class Edges* generate diverse counterfactuals that simulate broader spaces and conditions, providing insights into the model's ability to generalize across different situations.
- **Data Representation Analysis:** By examining how the model reacts to variations within and between classes, one can gain an understanding of whether the model has learned meaningful patterns or overfitted to the training data.

5.5.6 Debugging and Improving Models

- **Locating Errors:** Counterfactuals generated in problematic areas (e.g., low-density or boundary regions) can highlight where the model may have learned spurious correlations or errors.
- **Informing Model Refinement:** Insights from counterfactual analysis can guide feature engineering, training data augmentation, or even architecture adjustments to improve the model.

5.5.7 Ethical and Fairness Evaluation

- **Bias Detection:** By observing how the model responds to counterfactuals across different subgroups, one can identify potential biases or fairness issues.
- **Justifying Decisions:** Counterfactuals provide tangible examples that justify model decisions, aiding in compliance with regulatory frameworks such as GDPR and ensuring ethical AI practices.

Chapter 6

Experiments and Results

Experiments were conducted for each main stage of the proposed method described in the previous chapter. To do so, it was used nineteen real-world benchmark datasets from the UCI repository [222], as well as two gene expression datasets: *Golub* [232] and *BcrHess* [233]. One more dataset (*continuous casting*) was taken from a steel making industrial process to be adopted in an actual use case scenario during the counterfactual generation and selection stages. The datasets underwent pre-processing treatment, including previously known noise filtering, removal of missing values (NA) and wrong zeros values, attribute scaling to the range [-1, 1] and class labeling standardization set as +1 or -1. Table 6.1 show the benchmark 20 datasets metadata.

base	**N**	**Nd**	NC^{+}	NC^{-}
ILPD	579	10	165	414
appendicitis	106	7	85	21
australian	690	14	383	307
banknote	1372	4	762	610
breastHess	133	30	34	99
breastcancer	683	9	239	444
bupa	345	6	200	145
climate	540	18	46	494
diabetes	768	8	268	500
fertility	100	12	88	12
german	1000	24	300	700
glass	214	9	185	29
golub	72	3	51	21
haberman	306	3	81	225
heart	270	13	120	150
ionosphere	351	12	225	126
parkinsons	195	22	48	147
segmentation	210	18	180	30
sonar	208	60	111	97
contcasting	838	21	222	616

Table 6.1: Datasets metadata

The next sections describe the experiments according to the proposed method's main steps. To see details of the geometric method proposed in this work, see the previous chapter.

6.1 Generating geometric data structures and factual samples classification

As one can see from the experiments described in the theoretical foundations (Chapter 4), that the distance metric and the samples quality thresholds (TC) can influence on the Gabriel Graph classifier (GGClassification) performance.

This first experiment aimed to assess whether an alternative combination of distinct Minkowski's p parameter and TCs thresholds could enhance classification performance and, consequently, improve the accuracy of generating counterfactual samples. It was adopted an R implementation of Differential Evolution (DE) optimization to search the best combination of such parameters [225], [226].

To ensure statistical significance, the experiments were replicated using ten-fold cross-validation. The median Area Under the Curve (AUC) from the ten-fold cross-validation is presented in Table 6.2.

base	**AUC**	**tc1**	**tc2**	**minkowki_p**
ILPD	0.684	0.699	0.477	1.06
appendicitis	0.736	0.969	0.569	1.56
australian	0.873	0.146	0.569	1.57
banknote	1.000	0.640	0.793	2.00
breastHess	0.786	0.323	0.905	1.01
breastcancer	0.937	0.543	0.587	2.97
bupa	0.679	0.978	0.595	1.12
climate	0.866	0.235	0.0443	4.95
diabetes	0.728	0.400	0.0825	1.59
fertility	0.778	0.405	0.739	1.68
german	0.718	0.198	0.545	1.86
glass	0.988	0.773	0.700	1.13
golub	0.854	0.0077	0.00577	3.07
haberman	0.643	0.814	0.758	4.06
heart	0.82	0.535	0.0836	1.83
ionosphere	0.751	0.585	0.435	5.99
parkinsons	0.97	0.756	0.582	3.54
segmentation	0.974	0.850	0.865	2.00
sonar	0.89	0.0346	0.299	3.26
contcasting	0.88	0.085	0.865	2.00

Table 6.2: AUC of Minkowski distance and quality thresholds parameters optimization

One more experiment was performed in order to assess the use of *weighted-dimensions* Minkowski distance. The median of AUC from the ten-fold cross-validation is presented in Table 6.3. Using a different parametrization for the Gabriel graph classifier instead of the

base	AUC	tc1	tc2	minkowki_p
ILPD	0.674	0.855	0.370	1.49
appendicitis	0.730	0.982	0.0520	2.27
australian	0.877	0.616	0.757	1.45
banknote	1.000	0.477	0.693	2.66
breastHess	0.786	0.539	0.197	1.01
breastcancer	0.937	0.164	0.0899	2.97
bupa	0.683	0.663	0.269	1.11
climate	0.866	0.603	0.500	4.80
diabetes	0.737	0.0635	0.116	1.84
fertility	0.778	0.364	0.611	1.76
german	0.718	0.425	0.0496	1.87
glass	0.961	0.321	0.929	2.26
golub	0.86	0.229	0.375	4.65
haberman	0.627	0.909	0.244	2.92
heart	0.82	0.880	0.451	1.99
ionosphere	0.751	0.789	0.154	5.98
parkinsons	0.97	0.431	0.139	3.63
segmentation	0.974	0.209	0.895	2.94
sonar	0.873	0.982	0.504	3.15

Table 6.3: AUC of dimension-weighted Minkowski distance optimization

default one can enhance its performance.

Table 6.4 presents a side-by-side comparison of all the results obtained from the classification performance experiments. On can see a merged view with datasets' metadata and the correspondent AUC values from different approaches. AUC metric was obtained from the median of 10-fold cross validation strategy.

Base	N	Nd	NC^{+}	NC^{-}	Euc_ori	DE_Euc	DE_Mink_nwg	DE_Mink_wg
ILPD	579	10	165	414	0.62	**0.69**	**0.684**	0.674
appendicitis	106	7	85	21	0.75	**0.83**	0.736	0.730
australian	690	14	383	307	0.83	0.85	0.873	**0.877**
banknote	1372	4	762	610	**1.00**	**1.00**	**1.000**	**1.000**
breastHess	133	30	34	99	0.76	**0.83**	0.786	0.786
breastcancer	683	9	239	444	0.96	**0.99**	0.937	0.937
bupa	345	6	200	145	0.64	0.67	0.679	**0.683**
climate	540	18	46	494	0.80	0.75	**0.866**	**0.866**
diabetes	768	8	268	500	0.72	**0.76**	0.728	0.737
fertility	100	12	88	12	0.68	**0.82**	0.778	0.778
german	1000	24	300	700	0.62	0.69	**0.718**	**0.718**
glass	214	9	185	29	0.90	0.97	**0.988**	0.961
golub	72	3	51	21	0.74	0.85	0.854	**0.86**
haberman	306	3	81	225	0.62	**0.64**	**0.643**	0.627
heart	270	13	120	150	0.81	**0.82**	**0.82**	**0.82**
ionosphere	351	12	225	126	0.79	**0.84**	0.751	0.751

Continued on next page.

Table 6.4 (continued)

Base	**N**	**Nd**	NC^{+}	NC^{-}	**Euc_ori**	**DE_Euc**	**DE_Mink_nwg**	**DE_Mink_wg**
parkinsons	195	22	48	147	0.91	0.94	**0.97**	**0.97**
segmentation	210	18	180	30	0.95	0.96	**0.974**	**0.974**
sonar	208	60	111	97	0.86	**0.89**	**0.89**	0.873
contcasting	838	21	222	616	0.86	**0.88**	**0.88**	—

Table 6.4: Merged table of datasets metadata and AUC values from different methods. The contcasting dataset was not tested in weighted Minkowski distance experiment.

6.2 Generating counterfactual examples

Further experiments were conducted to evaluate the accuracy of the new geometric method proposed in this thesis for generating counterfactual examples that belong to the target class. For the current experiment accuracy is reach the target class. The same twenty datasets listed in Table 6.2 were utilized during the performance test. The results are presented in Table 6.5. The accuracy metric is represented by the median of the percentage of success (correct target class generation) across 100 runs.

Base	**ds**	**fi_20_10**	**mahala**	**hs_surf_int**	**hs_b_2pts**	**hs_ref**	**se**	**prj_ch**	**prot**	**edge**
ILPD	100.00	100.00	85.00	100.00	100.00	100.00	0.00	100.00	100.00	100.00
appendicitis	100.00	0.00	89.00	100.00	100.00	100.00	0.00	100.00	90.00	100.00
australian	100.00	0.00	100.00	100.00	100.00	100.00	0.00	100.00	100.00	100.00
banknote	100.00	100.00	90.00	100.00	100.00	100.00	11.00	100.00	100.00	100.00
breastHess	100.00	0.00	100.00	100.00	100.00	100.00	0.00	100.00	100.00	100.00
breastcancer	100.00	100.00	99.00	100.00	100.00	93.30	99.00	0.00	100.00	100.00
bupa	100.00	0.00	72.00	63.00	25.00	100.00	0.00	100.00	100.00	100.00
climate	100.00	100.00	100.00	96.00	98.00	100.00	99.00	100.00	100.00	100.00
diabetes	100.00	0.00	100.00	100.00	100.00	100.00	10.00	100.00	100.00	100.00
fertility	100.00	100.00	100.00	99.00	100.00	100.00	100.00	100.00	100.00	100.00
german	100.00	100.00	100.00	100.00	100.00	100.00	26.00	100.00	100.00	100.00
glass	100.00	0.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
golub	100.00	0.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
haberman	100.00	100.00	48.00	94.00	91.00	86.70	100.00	0.00	80.00	100.00
heart	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
ionosphere	100.00	0.00	100.00	70.00	65.00	100.00	100.00	0.00	40.00	93.00
parkinsons	100.00	0.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
segmentation	100.00	0.00	100.00	100.00	100.00	100.00	99.00	100.00	100.00	100.00
sonar	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
contcasting	100.00	100.00	100.00	100.00	100.00	100.00	99.00	100.00	100.00	100.00

Table 6.5: Median of the accuracy (%) in 100 runs. Ten different methods facing nineteen benchmark datasets. Description of the methods used in counterfactual generation. Each abbreviation corresponds to a specific technique or approach. The methods are: **ds** (Dominating Set-Oriented), **fi_20_10** (Feature Importance Adjustment considering top 20% most important features and 10% incremental changes), **mahala** (Mahalanobis Distance-based Sampling), **hs_surf_int** (Hypersphere Volume Sampling Centered at Reference Points), **hs_b_2pts** (Hypersphere Volume from Same-Class Edges), **hs_ref** (Utilizing Purity-Based Reference Points), **se** (Support Edge Crossing. Support edge interpolation from side c to side d), **prj_chull** (Projection onto Convex Hull), **proto** (Reference to Prototypes and Archetypes), and **edge** (Same-Class Edges Interpolation).

6.3 Imposing restrictions through the Constraint Matrix

To enforce some business rules, a restriction matrix was created to specify the limitations on modifying each characteristic when generating counterfactual examples. Table 6.6 was constructed from process specialists knowledge to run the present experiment (Imposing restrictions through the Constraint Matrix).

	Feature	Type	Lower	Upper	Monotonic	Immutable
1	VAZAO_AGUA_MIN	Bounded	2157.00	2656.50	none	FALSE
2	VAZAO_AGUA_MAX	Bounded	2212.00	2322.00	none	FALSE
3	LC_VELOC_AVG	Bounded	0.64	1.07	none	FALSE
4	PESO_BARRA	Bounded	3556.00	7648.00	none	FALSE
5	NIVEL_ACO_MAX	Immutable	45.00	125.00	none	TRUE
6	POSICAO_TAMPAO_AVG	Bounded	492.58	1511.13	none	FALSE
7	POSICAO_TAMPAO_MAX	Bounded	555.00	1568.00	none	FALSE
8	POSICAO_TAMPAO_MIN	Bounded	450.00	1448.00	none	FALSE
9	POSICAO_TAMPAO_DEV	Immutable	5.02	341.96	none	TRUE
10	TEMP_ENT_MIN	Bounded	21.10	35.64	none	FALSE
11	TEMP_ENT_MAX	Bounded	25.80	33.70	none	FALSE
12	TEMP_DELTA_AVG	Immutable	3.63	5.37	none	TRUE
13	CORRENTE_AVG	Bounded	299.36	300.13	none	FALSE
14	FREQ_MOLDE_AVG	Bounded	108.81	181.88	none	FALSE
15	PRESSAO_Z1_AVG	Bounded	4.32	10.59	none	FALSE
16	PRESSAO_Z2_AVG	Bounded	2.92	9.80	none	FALSE
17	DIM_PLA_LAM	Immutable	273.00	273.10	none	TRUE
18	ESP_PLA_LAM	Immutable	9.27	21.44	none	TRUE
19	PESO_DISTR_MAX	Bounded	12.05	39.12	none	FALSE
20	PESO_DISTR_MIN	Bounded	3.02	38.77	none	FALSE
21	PESO_DISTR_AVG	Immutable	8.74	38.98	none	TRUE

Table 6.6: Constraint matrix applied to continuous casting use case

The counterfactual generation process under constraints works iteratively, changing one feature at each step. Te procedure aims to update the original factual point while ensuring that feature-specific constraints were respected according to the constraint matrix. The constraints matrix outlined limits on permissible values, monotonic relationships, and whether certain features were immutable. This process ensures that generated counterfactuals adhere to user-specified constraints while making the transition across decision boundaries. Below, one can find the key steps involved in the experiment.

The reference actual sample xi_{ac}, is taken from the dataset with an associated class label. The goal is to identify the closest sample,xj_{ac}, from the opposite class with the highest intensity metric according to *Signed Softmax*. The idea is finding "who is the closest and *more opposite* factual counterpart". Among the opposite class samples, the closest point is then selected based on the same distance metric used in the classification. The resulting sample,xj^*, serves as the opposite class reference for generating counterfactuals. The equation 6.1 formalizes the calculation.

$$xj^* = \arg\min\{\text{dist}(xi_{ac}, xj_{ac})\} \quad \forall \quad xj_{ac} \in (-1) \cdot \text{Class}(xi_{ac}) \tag{6.1}$$

The counterfactual generation is achieved by applying an incremental changes in xi_{ac}, towards the closest opposite class point xj^*, while ensuring adherence to constraints defined in the *constraint matrix*.This process is iteratively repeated for a specified number of counterfactual new samples, a specif number of target success or a stop criteria. The core steps are as follows:

For each feature ft in the references samples, xi_{ac} and xj^*, it was calculated the direction of change. From xi_{ac} (actual sample) towards xj^* (actual sample that was select as the opposite class reference), as described by equation 6.2:

$$\Delta_{ft} = \text{sign}(xj^*_{ft} - xi_{ac,ft}) \tag{6.2}$$

Where $xi_{ac,ft}$ represents each feature of the sample xi_{ac} and xj^*_{ft} are the features of xj^*.

Using this direction, the proposed value for each feature in the counterfactual sample is computed as described by the equation 6.3:

$$x_{cf,ft} = xi_{ac,ft} + \Delta_{ft} \cdot \alpha \tag{6.3}$$

Where $x_{cf,ft}$ is the new counterfactual sample features' values, $xi_{ac,ft}$ represents the features of the actual sample xi_{ac} and Δ_{ft} the adjust factor (the intensity of the modification).

The proposed value is then adjusted to respect the feature constraints as specified in the *constraint matrix*:

- **Bounds:** The feature value is restricted within the lower and upper bounds, using the following operation:

$$x_{cf,ft} = \max(\text{Lower}_{ft}, \min(x_{cf,ft}, \text{Upper}_{ft})) \tag{6.4}$$

- **Monotonicity:** If the feature is defined to be monotonic (e.g., *increasing*), then the value is adjusted to ensure compliance:

$$\text{if Monotonic}_{ft} = \text{"increase" and } x_{cf,ft} < xi_{ac,ft}, \text{ then } x_{cf,ft} = xi_{ac,ft} \tag{6.5}$$

  Similar adjustments are made for features marked as *decreasing*.

- **Immutability:** Features marked as immutable remain unchanged during the generation process.

For the present use case it was possible to find a valid counterfactual under the restrictions in all of 30 runs.

6.4 Supporting Differential Evolution optimization

For investigate supporting differential evolution optimization, it was employed the R language implementation called `DEoptim`. The objective was supporting DE by supplying geometric and statistical information that are available in the proposed data structure. DE optimization aims to minimize the objective function, which in the present experiment represents a combination of distance from a reference sample and penalties associated with the constraint matrix violations.

The constraints were provided through a comprehensive constraint matrix, which included feature bounds, monotonicity, and immutability requirements. Specifically, penalties were introduced into the objective function whenever proposed solutions violated the given constraints, such as exceeding feature bounds, violating monotonic relationships, altering immutable features or the classification do not reaches the target class.

Additionally, to further enhance the performance of the DE optimization, it was supplied geometric and statistical information about the data to guide the process. For instance, feature weights were incorporated into the objective function to emphasize the importance of specific features, and Minkowski distance was used to measure similarity to the reference sample. Moreover, geometric information, such as support edges (frontier) and actual samples statistical information, can be used to initialize the population of candidate solutions. By doing so, we can ensure that the initial candidates are positioned in regions of the solution space that are more likely to yield feasible and optimal counterfactual examples, thereby improving the efficiency of the optimization process.

For the present use case it was possible to find a valid counterfactual under the constraint matrix restrictions via DE for all of the 30 runs.

6.5 Discussion and Analysis of Experiment Results

The results presented in Table 6.5 show the median accuracy from 100 runs of ten different counterfactual generation methods evaluated across nineteen benchmark datasets.

Each dataset has unique characteristics, as detailed in Table 6.1, such as the number of samples, number of features, and class distribution. This section provides a detailed discussion of how was the performance of the counterfactual generation methods during the computational experiments.

6.5.1 Performance of counterfactual methods across different datasets

The hypersphere-based methods (*hs_surf_int*, *hs_b_2pts*, and *hs_ref*) consistently achieved high accuracy rates across almost all datasets, indicating robustness in reaches the target class.These methods performed particularly well even on high-dimensional datasets such as *sonar* (60 dimensions), *breastHess* (30 dimensions), and *german* (24 dimensions).

The *low Mahalanobis distance* method (*mahala*) also demonstrated good performance across several datasets, for instance those with lower dimensionality, such as *golub* (only 3 dimensions), the *appendicitis* (7 dimensions), and *heart* (13 dimensions).The accuracy achieved by this method suggests that it is well-suited for datasets where feature correlations can be effectively captured using a Mahalanobis distance metric, allowing the generation of counterfactuals that align well with the target class.

Conversely, methods based on dominating sets (*ds*) and *features importance 20_10* showed varied performance across datasets.The *dominating sets* method exhibited excellent results in datasets such as *climate* and *german*, both of which have a relatively high number of features. This suggests that the use of dominating sets can be advantageous in complex feature spaces when disposing off significant number of samples (climate 540 and german 1000 samples). DS method could identify representative samples as anchors is such conditions. However, the performance of the dominating sets method declined sharply for datasets like *breastHess* and *bupa*, suggesting its sensitivity to class imbalance and limited sample sizes.

The *support edge crossing* (*se*) and *convex hull projection* (*prj_ch*) methods showed strong results for same unbalanced datasets and also for same with high number of features. However, their performance dropped considerably in datasets with more class overlapping, such as *bupa* and *haberman*. This could be attributed to the fact that these methods rely on the geometric relationships between class boundaries, which become less distinguishable in the presence of class overlapping.

The *clusters' prototypes* (*prot*) method generally performed well across most datasets, achieving 100% of accuracy in datasets such as *breastHess*, *german*, and *sonar* that presents high dimensionality. The effectiveness of this method in diverse datasets suggests its suitability for finding prototypical examples that are representative of the target class, thus ensuring the generation of plausible counterfactuals.

6.5.2 Impact of dataset characteristics on method performance

The number of samples, dimensionality, and class distribution played a significant role in determining the effectiveness of the counterfactual generation methods.

Datasets with a higher number of dimensions, such as *sonar* and *breastHess*, tended to benefit more regular spacing sampling far from the border, as the reference points hypersphere-based methods. The ability of these methods to explore the feature space more uniformly, and far from the frontier, likely contributed to their consistent high accuracy across these datasets.

Class overlapping also played a significant role in determining method performance. The datasets *appendicitis*, *bupa*, and *haberman*, which presents overlapping and some imbalance between positive and negative class instances showed lower performance for methods like *dominating sets* and *support edge crossing*, this last because works close to the border.

6.5.3 Qualitative comparison between other models

This section aims to present a comparative study on some qualitative characteristics of the counterfactual generation methods. For this purpose, part of the table published in the benchmark study [253] is adopted. The table originally presents a comparison of some methods for generating counterfactual examples (the first nine lines). The initial part of the table originally published in [253] was used here and completed with the geometric methods proposed in the present work. The new proposed method appear from the tenth line until the last one (after the vertical separation line).

Table 6.7: Qualitative counterfactual explanation generation algorithm characteristic from the benchmark study [253]. The authors indicated that the Black filled circles indicate the feature is present, while white filled circles represent that the feature is not present. CO: Convex Optimization. HE: Heuristic. SS: Satisfiability Solver. REQ.: Required. NRE.: Not Required. OPT.: Optional. Spar.: Sparsity. Prox.: Proximity. Feas.: Feasibility. Diver.: Diversity. W: (allows using weights). R: Range: (allows using ranges). OHE: One-Hot Encoding. The original table with the nine first lines appears in the benchmark investigation published in [253]

Algorithm	Find Method	Data Access	Model Access	Model Comp.	Spar.	Prox.	Feas.	Diver.	W	R	OHE
ALIBIC [234]	CO	REQ.	OPT.	Agnostic	●	●	○	○	○	○	○
CADEX [235]	CO	NRE.	REQ.	Specific	●	○	○	●	●	●	○
DiCE [160]	CO	OPT.	REQ.	Specific	●	●	●	○	○	○	○
Growing Spheres [174]	HE	OPT.	NRE.	Agnostic	○	●	○	○	○	○	○
LORE [236]	HE	REQ.	NRE.	Agnostic	○	○	○	●	○	●	●
MACE [237]	SS	REQ.	REQ.	Specific	●	●	○	○	●	●	○
ML-Explain [238]	CO	NRE.	REQ.	Specific	●	○	●	○	○	●	●
SEDC [239]	HE	REQ.	NRE.	Agnostic	○	●	○	○	●	●	○
SynAS [240]	CO	REQ.	REQ.	Specific	●	○	○	○	○	○	○
Same-Class Edges	HE	REQ.	NRE.	Agnostic	●	○	●	○	●	●	○
Opposite-Class Edges Interpolation	HE	REQ.	NRE.	Agnostic	○	●	○	●	●	●	○
Hypersphere Sampling Around Reference Points	HE	REQ.	NRE.	Agnostic	○	○	●	●	●	●	○
Spherical Interpolation Across the Border	HE	REQ.	NRE.	Agnostic	○	●	○	●	●	●	○
Utilizing Purity-Based Reference Points	HE	REQ.	NRE.	Agnostic	○	○	●	○	●	●	○
Shortest Path in Graph to Target Samples	HE	REQ.	NRE.	Agnostic	●	●	○	○	●	●	○
Density-Based Sampling Towards High-Density Regions	HE	REQ.	NRE.	Agnostic	○	○	●	○	●	●	○
Projection in the Hull	CO	REQ.	NRE.	Agnostic	●	●	○	●	●	●	○
Voronoi Diagram Regional Sampling	HE	REQ.	NRE.	Agnostic	○	○	●	●	●	●	○
Reference to Prototypes	HE	REQ.	NRE.	Agnostic	○	○	●	○	●	●	○
Reference to Archetypes	HE	REQ.	NRE.	Agnostic	●	○	○	●	●	●	○
Feature Importance Adjustment	HE	REQ.	NRE.	Agnostic	●	●	○	○	●	●	○
Random Spherical Interpolation (Slerp)	HE	REQ.	NRE.	Agnostic	○	○	○	●	●	●	○
Multi-Objective Optimization	CO	REQ.	REQ.	Agnostic	●	●	●	○	●	●	○
Statistical Characteristic Adjustment	HE	REQ.	NRE.	Agnostic	○	○	●	○	●	●	○
Delaunay Triangulation Sampling	HE	REQ.	NRE.	Agnostic	○	●	●	●	●	●	○

Algorithm	Find Method	Data Access	Model Access	Model Comp.	Spar.	Prox.	Feas.	Diver.	W	R	OHE
DE Optimization with Geometric Constraints	CO	REQ.	REQ.	Agnostic	●	●	●	●	●	●	○
Statistical based features adjustments	HE	REQ.	NRE.	Agnostic	○	●	●	○	●	●	○
Combination of proposed methods	HE	REQ.	NRE.	Agnostic	●	●	●	●	●	●	○

6.5.4 General conclusion from the experiments' results

The results of the experiments indicate that there is no silver bullet solution for counterfactual examples generation. Some datasets characteristics, such as dimensionality, class balance, overlapping and sample size, should be considered when selecting the appropriate method. A hybrid approach that combines multiple methods may offer the most comprehensive solution, as it can leverage the strengths of each individual method to mitigate the weaknesses encountered in specific datasets.

Chapter 7

Conclusion

The present research addressed the challenge of enhancing interpretability in Artificial Intelligence classifier models through the geometric-driven generation of counterfactual examples. By employing geometric data structures, particularly the Gabriel Graph classifier (GGClassification), this work demonstrated novel methodologies for generating high-quality counterfactual samples. These methodologies aim to improve the understanding of the AI model's classifier behavior, providing insights into decision-making processes and enabling practical applications in critical domains such as healthcare, finance, and industrial systems. This chapter consolidates the study's findings, contributions, and implications while highlighting potential future directions.

Key Findings

- **Effectiveness of Geometric Structures:**
  - The Gabriel Graph proved to be a highly effective tool for guiding the generation of counterfactual examples. Its ability to capture similarities, spatial relationships, and decision boundaries enabled the creation of realistic and actionable counterfactuals.
  - The use of statistical concepts, such as probability density, Mahalanobis distance, proximity, and sparsity, enhanced the quality of generated samples, ensuring they were both plausible and interpretable.
- **Performance Across Datasets:**
  - The proposed methods demonstrated consistent performance across diverse datasets during the computational experiments, including high-dimensional data. This robustness underscores the adaptability of geometric approaches to varied actual data scenarios.
  - Experiments revealed that the Gabriel Graph's adjacency matrix structure facilitated efficient exploration of decision boundaries, outperforming traditional methods regarding computational cost and interpretability.
  - The GGclassification's geometric approach can provide a useful set of data structures containing a pure geometric mapping of the space, similarities relationships, sample density, sample purity (quality) metrics, hyperplane details, and the traceability of the Exper Mixture classification process. Such information arsenal, combined with statistical metrics obtained from the own dataset, can guide

the generation of counterfactual examples according to the specific needs of the use case in question.

- **Practical Applications:**
  - The methods were effective in actual datasets, including industrial use cases (continuous casting and rolling mill).
  - The study highlighted the potential of geometric counterfactual generation in enhancing fairness, bias detection, and robustness of AI models.
- **Improvement of the Gabriel graph-based classifier (GGClassification) performance:**
  - Empirical proof that the performance of the Gabriel Graph-based classifier (GGClassification) can be improved by adopting different distance metrics and sample quality thresholds. One of the contributions of this work was to extrapolate the original operation of GGclassification, allowing the use of different distance metrics, filtering strategies and sample quality threshold values.
- **Ethical and Explainability Contributions:**
  - The proposed methods align with ethical considerations in AI by promoting justification of the models' outputs.

Contributions

This research makes some contributions to the field of XAI:

1. **Novel Methodology:** The introduction of geometric structures, particularly the Gabriel Graph Classification data structure, as a basis for counterfactual sample generation represents a novel contribution. This approach proposes a deeper investigation regarding using geometric and graph modeling for AI's interpretability purposes.
2. **Empirical Validation:** Comprehensive experiments validated the proposed methods, demonstrating their efficacy in generating high-quality counterfactuals and enhancing model interpretability.
3. **Practical Impact:** In a broader picture regarding XAI, the research realized a vast investigation and tested methodologies for real-world applications, mainly to industrial scenarios, offering a pathway for industries to adopt interpretable AI techniques. The geometric-driven counterfactual examples generation can be applied in industrial shop-floor, for instance, supporting real-time golden batch analysis. For example, suggesting changes in machine settings according to the current/real conditions of the product to be processed.
4. **Theoretical Advancements:** By integrating geometric and statistical metrics, the study advances the theoretical understanding of counterfactual generation, providing new insights for future research.
5. **GGClassification improvement:** extrapolate the original operation of GGclassification, allowing the use of different distance metrics, filtering strategies and sample quality threshold values.

Implications and Limitations

The findings of this research bring some implications:

- **Improved Decision-Making:** The proposed methods enable users to understand model behavior, fostering trust and facilitating better decision-making.
- **Scalability and Efficiency:** The computational efficiency of the geometric approach makes it suitable for real-time applications, addressing a critical need in high-stakes scenarios.
- **Interdisciplinary Applications:** The methodologies have potential applications beyond AI, including operations research, optimization, decision-making processes, machinery setups (settings) in traceability, and control systems.

However, the study also has limitations.

Limitations

- **High-Dimensional Challenges:** While the methods performed well across datasets, handling extremely high-dimensional data remains a computational challenge. Future work could explore optimization techniques or dimensionality reduction strategies that maintain interpretability.
- **Algorithmic Complexity:** The complexity of some geometric operations, particularly in large-scale datasets, may limit the scalability of the approach in specific contexts. The computational complexity of some geometric structures is still a limiting factor in higher dimensions.
- **Two class investigation:** The present investigation dealt only with binary classification problems (datasets with samples classified into only two classes). However, initial tests indicate that investigations of classification problems with more than two classes and regression problems are promising.

Sumary and final remarks

Despite the conceptual simplicity of "how to change the classification of a sample by making minimal changes to its characteristics", its computational implementation is complex. It is not simple to create a generic and robust platform. Much of the complaints about the lack of source code and functional examples for generating counterfactual examples are due to this.

The experiments have indicated that geometric data structures such as Gabriel graphs can be used successfully to generate satisfactory counterfactual examples in different datasets conditions. Experiments conducted using distinct datasets have demonstrated that the analyzed geometric techniques can produce meaningful counterfactuals that can help understand the behavior of models and have insights regarding what changes are necessary to invert a particular classification model output.

The Gabriel graph, in particular, has shown a versatile and computationally low-cost tool. It supplies essential similarities and information that can be conciliated with the statistical characteristics of the dataset under analysis. By generating a graph that connects data points based on their proximity and removing connections that are not significant, the Gabriel graph can effectively capture the structure of the data and highlight key patterns and relationships. Such similarities-based structures associated with statistical information in the space can be valuable in generating counterfactual examples under different requirements. In general, one can conclude from the present study that these geometric structures could be more often employed in the context of XAI and present them as possible tools for future investigations.

Chapter 8

Suggestions for Future Works

In a more focused context on geometric approaches, it is recommended to continue exploring the following topics:

1. **Graph theory incorporation**: Investigate the integration of additional graph structures and graph theory techniques.
2. Include additional **geometric structures**: including other representation such as Delaunay triangulation or Minimum Spanning Trees.
3. **Multi-class classification:** Expand the methodology to handle multi-class classification problems or intraclass clusters.
4. **New distance metrics**: extend the use of distance metrics, for instance investigating the usage of Mahalanobis distance in the Graph construction.
5. **Frontier and overlapping** treatment: Overlapping and frontier analysis is an interesting matter regarding geometric structures.
6. Generating interpretability from the geometric and statistics information.

In the broader context of AI, the literature highlights some points with potential for relevant advancements:

1. **Addressing Security And Privacy Involving Counterfactual Generation**: The use of XAI and counterfactual generation in machine learning introduces privacy risks, which currently remain largely unnoticed [241]. Counterfactuals carry information that adversarial users can exploit to launch model extraction attacks [242].
2. **Answering Counterfactual Questions And Causal Inference**: Investigating ways of standardizing ways of representing counterfactual questions and how to deal with causal relationships in counterfactual generation problems [243].
3. **Addressing Causal Mechanisms In Counterfactual Scenarios**: For instance, to deal with scarcity in "few-shot" learning [244, 245].
4. **Researches In Handling Images Features In Counterfactual Examples Generation**: [246].

5. **Data-Independent Counterfactual Examples Generation**: Investigations of model-agnostic methods and exploring counterfactual generation in multi-modal contexts (integrating multiple types of data, text, image, tables) [247].

6. **Building Explanation Using Psychology-Based Format**: Novel potential to the explanations using *gestalt* techniques [248–250].

7. **Investigating Geometric Approaches**: It can open new branches to counterfactual examples generation researches. It also can be expanded to other tasks involving explaining machine learning models' decisions.

8. **Lack Of Code Implementation**: Although the numerous approaches, many of them lack code implementation or present code issues [251–253].

9. **Causal Understanding And Counterfactuals**: Investigate the integration of causal models with counterfactual explanations.

10. **Robustness And Stability**: Focus on the robustness and stability of counterfactual explanations under diverse data distributions and classes overlapping rates.

11. **User-Centric Explanations**: Tailor counterfactual explanations to the needs of distinct comprehension levels of different user groups (including non-experts) [254].

12. **Integration With Existing Systems**: Explore methods for integrating counterfactual explanations into existing decision-making systems and workflows.

13. **Evaluation Metrics**: Develop standardized metrics for evaluating the quality and effectiveness of counterfactual explanations.

14. **Validity And Sparsity With Human-In-The-Loop**: Future research should explore methods that improve the quality of generated counterfactuals by investigating ways of ensuring their plausibility according to user interactions [255].

15. **Visualization Of Counterfactual Explanations**: There is a significant gap in the methods used to visualize counterfactual explanations effectively. Since these explanations are often presented to users with varying levels of expertise, research on user-friendly visualization methods is crucial [255].

16. **Counterfactuals For Regression Models**: Most counterfactual research has been focused on classification tasks. Future work should consider developing counterfactual methods for regression models, which predict continuous outcomes rather than discrete classes [251].

17. **Unification Of Vocabulary And Terminology In Counterfactual Research**: Current literature on counterfactual examples often uses inconsistent terminology, leading to confusion. Research that standardizes terms such as "contrastive explanation," "algorithmic recourse," etc., is needed to facilitate better communication and development in the field [251].

18. **Investigating Legal And Ethical Issues**: Using counterfactuals in critical artificial intelligence models. Malicious usage and how to avoid it [256].

19. **Exploring Counterfactual Explanations For Reinforcement Learning Models**: How to use counterfactual examples to improve models training. Analyzing borders details and low performance of a given space region.

20. **Deeper Investigating Trade-Offs Between Fidelity And Plausibility**: It is a key point regarding counterfactual generation, but it still requires investigation.

21. **Temporal Counterfactual Explanations**: Many applications in the real world deal with time-dependent variables. Future research could focus on exploring temporal-dependency in counterfactuals analysis. For instance, taking into account the sequential nature of time-series or handling time-series data type.

22. **Algorithmic Optimization**: Developing faster algorithms for geometric computations, particularly for high-dimensional and large-scale datasets, will enhance the scalability of the proposed methods.

23. **Integration With Other Explainability Techniques**: Combining geometric counterfactual generation with methods like SHAP and LIME could provide richer and more comprehensive explanations.

24. **Enhanced User Interaction**: Future work could explore interactive tools that allow users to visualize and manipulate counterfactual examples in real-time, improving accessibility and usability.

25. **Real Time Application of Counterfactual Examples Investigations**: Future work could explore geometric-driven counterfactual examples generation in industrial applications, specially in shop-floor, for instance, supporting real-time "golden batch/optimization quality analysis" by suggesting changes in machine settings according to the current/real conditions of the product to be processed.

Bibliography

[1] Federico Cabitza, Andrea Campagner, Gianclaudio Malgieri, Chiara Natali, David Schneeberger, Karl Stoeger, and Andreas Holzinger. Quod erat demonstrandum? - towards a typology of the concept of explanation for the design of explainable ai. *Expert Systems with Applications*, 213:118888, 2023.

[2] Nicole Van Hoeck, Patrick D Watson, and Aron K Barbey. Cognitive neuroscience of human counterfactual reasoning. *Frontiers in human neuroscience*, 9:420, 2015.

[3] Luis Carlos Bambirra Torres. *Classificador por Arestas de Suporte (CLAS): Métodos de Aprendizado Baseados em Grafos de Gabriel*. PhD thesis, Universidade Federal de Minas Gerais, Belo Horizonte, MG. Brasil", 02 2016.

[4] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador Garcia, Sergio Gil-Lopez, Daniel Molina, Richard Benjamins, Raja Chatila, and Francisco Herrera. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*, 58:82 – 115, 2020.

[5] Hannah Arendt and Jens Kroh. *Eichmann in Jerusalem*. Viking Press New York, 1964.

[6] Olavo de Carvalho. *A filosofia e seu inverso*. Vide Editorial, São Paulo, 2012.

[7] Gisela Striker. *Aristotle's Prior Analytics Book I: Translated with an Introduction and Commentary*. Oxford University Press, Oxford, England, 2009.

[8] P Hájek, I Havel, and M Chytil. The guha method of automatic hypotheses determination. *Computing*, 1:293–308, 12 1966.

[9] Suhas Jangoan, Gowrisankar Krishnamoorthy, Muthukrishnan Muthusubramanian, and Kapil Kumar Sharma. Demystifying explainable ai: Understanding, transparency, and trust. *International Journal For Multidisciplinary Research*, 6(2):1–13, 2024.

[10] M.Z. Naser. An engineer's guide to explainable artificial intelligence and interpretable machine learning: Navigating causality, forced goodness, and the false perception of inference. *Automation in Construction*, 129:103821, 2021.

[11] Andrey Bondarenko and Ludmila Aleksejeva. Methodology for knowledge extraction from trained artificial neural networks. *Information Technology and Management Science*, 21:6–14, 12 2018.

[12] Arun Das and Paul Rad. Opportunities and challenges in explainable artificial intelligence (xai): A survey, 2020.

[13] Samuli Laato, Miika Tiainen, AKM Najmul Islam, and Matti Mäntymäki. How to explain ai systems to end users: a systematic literature review and research agenda. *Internet Research*, 32(7):1–31, 2022.

[14] Andrea Ferrario and Michele Loi. How explainability contributes to trust in ai. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '22, page 1457–1466, New York, NY, USA, 2022. Association for Computing Machinery.

[15] A. Adadi and M. Berrada. Peeking inside the black-box: A survey on explainable artificial intelligence (xai). *IEEE Access*, 6:52138–52160, 2018.

[16] R. Roscher, B. Bohn, M. F. Duarte, and J. Garcke. Explainable machine learning for scientific insights and discoveries. *IEEE Access*, 8:42200–42216, 2020.

[17] Blerta Abazi Chaushi, Besnik Selimi, Agron Chaushi, and Marika Apostolova. Explainable artificial intelligence in education: A comprehensive review. In *World Conference on Explainable Artificial Intelligence*, pages 48–71. Springer, 2023.

[18] Erik Cambria, Lorenzo Malandri, Fabio Mercorio, Navid Nobani, and Andrea Seveso. Xai meets llms: A survey of the relation between explainable ai and large language models. *arXiv preprint arXiv:2407.15248*, 2024.

[19] Tetyana Tanya Krupiy. A vulnerability analysis: Theorising the impact of artificial intelligence decision-making processes. *Computer Law and Security Review*, 38:105429, 2020.

[20] Margarita Robles Carrillo. Artificial intelligence: From ethics to law. *Telecommunications Policy*, 44(6):101937, 2020. Artificial intelligence, economy and society.

[21] Miriam C Buitten. Towards intelligent regulation of artificial intelligence. *European Journal of Risk Regulation*, 10(1):41–59, 2019.

[22] EPRS. The impact of the general data protection regulation (gdpr) on artificial intelligence. Technical Report 1, EPRS - European Parliamentary Research Service - Scientific Foresight Unit (STOA), The address of the publisher, 6 2020. Panel for the Future of Science and Technology.

[23] CFR. Code of federal regulations - title 12 - chapter x - part 1002 - 1002.9. on line publication., 10 2020.

[24] The-White-House. Executive order on maintaining american leadership in artificial intelligence. Published on line, 02 2019.

[25] Presidência da República. Lei geral de proteção de dados pessoais (lgpd) - lei n° 13.709. agosto de 2018. Published on line, 07 2018.

[26] Bryce Goodman and Seth Flaxman. European union regulations on algorithmic decision-making and a "right to explanation". *AI Magazine*, 38(3):50–57, Oct. 2017.

[27] Hana Kopecka, Jose Such, and Michael Luck. Preferences for AI Explanations Based on Cognitive Style and Socio-Cultural Factors. *Proceedings of the ACM on Human-Computer Interaction*, 8(CSCW1):1–32, apr 17 2024.

[28] Dimitrios P Panagoulias, Maria Virvou, and George A Tsihrintzis. A novel framework for artificial intelligence explainability via the technology acceptance model and rapid estimate of adult literacy in medicine using machine learning. *Expert Systems with Applications*, 248:123375, 2024.

[29] Harishankar V Subramanian, Casey Canfield, and Daniel B Shank. Designing explainable ai to improve human-ai team performance: a medical stakeholder-driven scoping review. *Artificial Intelligence in Medicine*, page 102780, 2024.

[30] Ghada Elkhawaga, Omar M. Elzeki, Mervat Abu-Elkheir, and Manfred Reichert. Why should i trust your explanation? an evaluation approach for xai methods applied to predictive process monitoring results. *IEEE Transactions on Artificial Intelligence*, 5(4):1458–1472, 2024.

[31] Georgios Pavlidis. Unlocking the black box: analysing the EU artificial intelligence act's framework for explainability in AI. *Law, Innovation and Technology*, 16(1):293–308, jan 2 2024.

[32] Chinasa T. Okolo, Dhruv Agarwal, Nicola Dell, and Aditya Vashistha. "if it is easy to understand then it will have value": Examining perceptions of explainable ai with community health workers in rural india. 8(CSCW1), apr 2024.

[33] Wolter Pieters. Explanation and trust: what to tell the user in security and ai? *Ethics and Information Technology*, 13(1):53–64, March 2011.

[34] Tobias Kopp. Facets of trust and distrust in collaborative robots at the workplace: Towards a multidimensional and relational conceptualisation. *International Journal of Social Robotics*, 16(6):1445–1462, 2024.

[35] Lisa M. PytlikZillig and Christopher D. Kimbrough. *Consensus on Conceptualizations and Definitions of Trust: Are We There Yet?*, pages 17–47. Springer International Publishing, Cham, 2016.

[36] Han Lin, Jiatong Han, Pingping Wu, Jiangyan Wang, Juan Tu, Hao Tang, and Liuning Zhu. Machine learning and human-machine trust in healthcare: A systematic survey. *CAAI Transactions on Intelligence Technology*, 9(2):286–302, 2024.

[37] W. Pieters. Acceptance of voting technology: between confidence and trust. In K. Stølen, W. Winsborough, F. Martinelli, and F. Massacci, editors, *Trust management: 4th international conference (iTrust 2006), proceedings*, volume 3986 of *Lecture Notes in Computer Science*, pages 283–297, Berlin, 2006. Springer.

[38] Niklas Luhmann. *Trust and power*. John Wiley & Sons, 2018.

[39] N. Luhmann. Familiarity, confidence, trust: Problems and alternatives. In D. Gambetta, editor, *Trust: Making and breaking of cooperative relations*. Basil Blackwell, Oxford, 1988.

[40] Barbara D Adams and Robert Webb. Trust in small military teams. In *7th international command and control technology symposium*, pages 1–20, 2002.

[41] W. V. Camp. Explaining understanding (or understanding explanation). *European Journal for Philosophy of Science*, 4:95–114, 2014.

[42] M. Ford and P. Johnson-Laird. Mental models: Towards a cognitive science of language, inference, and consciousness. *Language*, 61:897, 1985.

[43] Gregory D. Abowd, Anind K. Dey, Peter J. Brown, Nigel Davies, Mark Smith, and Pete Steggles. Towards a better understanding of context and context-awareness. In Hans-W. Gellersen, editor, *Handheld and Ubiquitous Computing*, pages 304–307, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg.

[44] Jonathan L. Kvanvig. *The Value of Knowledge and the Pursuit of Understanding*. Cambridge University Press, Cambridge, 2003.

[45] Stephen Grimm. 'the value of understanding'. *Philosophy Compass*, 7(2), 2012.

[46] Catherine Z Elgin. Considered judgment. 1999.

[47] Linda Trinkaus Zagzebski. *Virtues of the mind: An inquiry into the nature of virtue and the ethical foundations of knowledge*. Cambridge University Press, 1996.

[48] Stephen R Grimm, Christoph Baumberger, and Sabine Ammon. Explaining understanding. *New perspectives from epistemology and philosophy of science*, 2017.

[49] Rand J. Spiro, Bertram C. Bruce, and William F. Brewer. Theoretical issues in reading comprehension : perspectives from cognitive psychology, linguistics, artificial intelligence, and education. 1980.

[50] Rand J Spiro, Bertram C Bruce, and William F Brewer. *Theoretical issues in reading comprehension: Perspectives from cognitive psychology, linguistics, artificial intelligence and education*, volume 11. Routledge, 2017.

[51] Vladimir Shadrikov. Understanding: Definition and mechanisms. *Cultural-Historical Psychology*, 2019.

[52] Tom Trabasso and Joseph P. Magliano. Conscious understanding during comprehension. *Discourse Processes*, 21:255–287, 1996.

[53] Tom Trabasso and Soyoung Suh. Understanding text: Achieving explanatory coherence through on-line inferences and mental operations in working memory. *Discourse Processes*, 16:3–34, 1993.

[54] Yan M. Yufik. The understanding capacity and information dynamics in the human brain. *Entropy*, 21, 2019.

[55] Ileana Maria Greca and Marco Antonio Moreira. Mental, physical, and mathematical models in the teaching and learning of physics. *Science Education*, 86:106–121, 2002.

[56] Michael Strevens.

[57] Nathanael Stein. Causation and explanation in aristotle. *Philosophy Compass*, 6(10):699–707, 2011.

[58] Emmanuel Monod, Jessie Pallud, and Tony Vance. The implications of the linguistic turn on is. In *Americas Conference on Information Systems*, 2006.

[59] Colin Koopman. Rorty's Linguistic Turn: Why (More Than) Language Matters to Philosophy. *Contemporary Pragmatism*, 8(1):61–84, apr 21 2011.

[60] Peter M. S. Hacker. *The Linguistic Turn in Analytic Philosophy*. Oxford University Press, jun 20 2013.

[61] G. Randolph Mayes. Theories of explanation. *Naturalistic Epistemology*, 2020.

[62] Leah Henderson. Putting inference to the best explanation into context. *Studies in History and Philosophy of Science*, 94:167–176, 2022.

[63] Peter Lipton. Inference to the best explanation. *A Companion to the Philosophy of Science*, pages 184–193, 2017.

[64] Jonah N Schupbach. Inference to the best explanation, cleaned up and made respectable. *Best explanations: New essays on inference to the best explanation*, pages 39–61, 2017.

[65] Gilbert H Harman. The inference to the best explanation. *The philosophical review*, 74(1):88–95, 1965.

[66] Maria Aslam, Diana Segura-Velandia, and Yee Mey Goh. A conceptual model framework for xai requirement elicitation of application domain system. *IEEE Access*, 11:108080–108091, 2023.

[67] Robert R Hoffman, Shane T Mueller, Gary Klein, and Jordan Litman. Measures for explainable ai: Explanation goodness, user satisfaction, mental models, curiosity, trust, and human-ai performance. *Frontiers in Computer Science*, 5:1096257, 2023.

[68] Muhammad Suffian, Pierluigi Graziani, Jose M. Alonso, and Alessandro Bogliolo. Fce: Feedback based counterfactual explanations for explainable ai. *IEEE Access*, 10:72363–72372, 2022.

[69] Francesco Sovrano and Fabio Vitali. Explanatory artificial intelligence (yai): human-centered explanations of explainable ai and complex data. *Data Mining and Knowledge Discovery*, pages 1–28, 2022.

[70] Francesco A Genco and Francesca Poggiolesi. Defining formal explanation in classical logic by substructural derivability. In *Connecting with Computability: 17th Conference on Computability in Europe, CiE 2021, Virtual Event, Ghent, July 5–9, 2021, Proceedings 17*, volume 12813, pages 237–247. Springer, 2021.

[71] Francesco A. Genco. Formal explanations as logical derivations. *Journal of Applied Non-Classical Logics*, 31(3-4):279–342, oct 2 2021.

[72] Sandeep Prasada. The scope of formal explanation. *Psychonomic Bulletin Review*, 24(5):1478–1487, apr 3 2017.

[73] Sandeep Prasada. *Formal Explanation and Mechanisms of Conceptual Representation*, pages 269–286. Routledge, feb 1 2021.

[74] T. Bosse, C. M. Jonker, and J. Treur. Componential explanation in philosophy, cognitive science and computer science. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 28, 2006.

[75] Alejandro J. García, Carlos I. Chesñevar, Nicolás D. Rotstein, and Guillermo R. Simari. Formalizing dialectical explanation support for argument-based reasoning in knowledge-based systems. *Expert Systems with Applications*, 40(8):3233–3247, 6 2013.

[76] Dwight Read. A formal explanation of formal explanation. *Structure and Dynamics: eJournal of Anthropological and Related Sciences*, 3(2), 01 2008.

[77] Akhila Bairy, Willem Hagemann, Astrid Rakow, and Maike Schwammberger. Towards formal concepts for explanation timing and justifications. In *2022 IEEE 30th International Requirements Engineering Conference Workshops (REW)*, pages 98–102, August 2022 2022.

[78] Francesco Sovrano and Fabio Vitali. From philosophy to interfaces: an explanatory method and a tool inspired by achinstein's theory of explanation. In *Proceedings of the 26th International Conference on Intelligent User Interfaces*, IUI 21, page 81–91, New York, NY, USA, 2021. Association for Computing Machinery.

[79] Carl G. Hempel and Paul Oppenheim. Studies in the logic of explanation. *Philosophy of Science*, 15(2):135–175, 1948.

[80] Peter Achinstein. *The Nature of Explanation*. Oxford University Press, USA, 1983.

[81] Stathis Psillos. The deductive-nomological model of explanation. In *The Routledge Handbook of Logical Empiricism*, page 9. Routledge, 1st edition edition, 2021.

[82] Xingming Hu. Hempel on scientific understanding. *Studies in History and Philosophy of Science*, 88:164–171, 2021.

[83] Wesley C. Salmon. *Scientific Explanation and the Causal Structure of the World*. Princeton University Press, 1984.

[84] Ruth M. J. Byrne. Good explanations in explainable artificial intelligence (xai): Evidence from human explanatory reasoning. In *International Joint Conference on Artificial Intelligence*, 2023.

[85] David Tuckey, Alessandra Russo, and Krysia Broda. A general framework for scientifically inspired explanations in ai. *arXiv preprint arXiv:2003.00749*, 2020.

[86] Robert R. Hoffman, Timothy Miller, and William J. Clancey. Psychology and ai at a crossroads: How might complex systems explain themselves? *The American Journal of Psychology*, 2022.

[87] Lloyd S. Shapley. *A Value for n-Person Games. Vol II of Contributions to the Theory of Games*. Princeton University Press, Santa Monica - CA, 1953.

[88] Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Proceedings of the 31st Int. Conf. on Neural Information Processing Systems*, NIPS'17, page 4768–4777, Red Hook, NY, USA, 2017. Curran Associates Inc.

[89] 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 Int. Conf. on Knowledge Discovery and Data Mining*, KDD '16, page 1135–1144, New York, NY, USA, 2016. Association for Computing Machinery.

[90] Riccardo Guidotti. Counterfactual explanations and how to find them: literature review and benchmarking. *Data Mining and Knowledge Discovery*, 38(5):2770–2824, 2024.

[91] Greta Warren, Ruth M. J. Byrne, and Mark T. Keane. Categorical and continuous features in counterfactual explanations of ai systems. In *Proceedings of the 28th International Conference on Intelligent User Interfaces*, IUI 23, page 171–187, New York, NY, USA, 2023. Association for Computing Machinery.

[92] Roderick M. Chisholm. The contrary-to-fact conditional. *Mind*, 55(220):289–307, 1946.

[93] Qiaohui Zheng, Mengtian Liao, Bingqing Liu, Weiqiang Ou, Wei Chen, Jianhui Liu, and Yan Zhang. Counterfactual thinking-related emotional responses in patients with major depressive disorder. *Frontiers in Psychiatry*, 11, 2021.

[94] George G. S. Murphy. On counterfactual propositions. *History and Theory*, 9:14–38, 1969.

[95] Mohammad Reza Taleban. Using counterfactual analysis for providing historical explanations in social sciences. *Journal of Social Sciences Ferdowsi University of Mashhad*, 2010.

[96] Ruth M.J. Byrne. *The Rational Imagination: How People Create Alternatives to Reality*. MIT Press, 2005.

[97] K. D. Markman, M. J. Lindberg, L. J. Kray, and A. D. Galinsky. Implications of counterfactual structure for creative generation and analytical problem solving. *Personality and Social Psychology Bulletin*, 33:312–324, 2007.

[98] R. Smallman and K. C. McCulloch. Learning from yesterday's mistakes to fix tomorrow's problems: when functional counterfactual thinking and psychological distance collide. *European Journal of Social Psychology*, 42:383–390, 2012.

[99] M. J. Tobia, R. Guo, U. Schwarze, W. Boehmer, J. Gläscher, B. Finckh, et al. Neural systems for choice and valuation with counterfactual learning signals. *NeuroImage*, 89:57–69, 2014.

[100] Giorgio Coricelli and Aldo Rustichini. Counterfactual thinking and emotions: regret and envy learning. *Philosophical Transactions of the Royal society B: Biological sciences*, 365(1538):241–247, 2010.

[101] Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2000.

[102] Judea Pearl. Causal diagrams for empirical research. *Biometrika*, 82(4):669–688, 1995.

[103] Donald B. Rubin. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology*, 66(5):688–701, 1974.

[104] Patrick Blackburn, Johan FAK van Benthem, and Frank Wolter. *Handbook of modal logic*. Elsevier, 2006.

[105] Nicholas Rescher. *Hypothetical reasoning*. North-Holland Publishing, Amsterdam, 1964.

[106] Thao Le, Tim Miller, Ronal Singh, and Liz Sonenberg. Improving model understanding and trust with counterfactual explanations of model confidence. *arXiv preprint arXiv:2206.02790*, 2022.

[107] Tobias Gerstenberg. What would have happened? counterfactuals, hypotheticals and causal judgements. *Philosophical Transactions of the Royal Society B*, 377(1866):20210339, 2022.

[108] Sarah R Beck, Daniel P Weisberg, Patrick Burns, and Kevin J Riggs. Conditional reasoning and emotional experience: A review of the development of counterfactual thinking. *Studia Logica*, 102:673–689, 2014.

[109] Neal J Roese and James M Olson. Counterfactual thinking: A critical overview. *Annual Review of Psychology*, 48:33–58, 1997.

[110] David R Mandel, Denis J Hilton, and Patrizia Catellani. *The psychology of counterfactual thinking*. Routledge, 2005.

[111] Kai Epstude and Neal J Roese. The functional theory of counterfactual thinking: New evidence, new challenges, new insights. *Advances in Experimental Social Psychology*, 40:1–34, 2008.

[112] Ruth M. J. Byrne. Counterfactual thought. *Annual Review of Psychology*, 67:135–157, 2016.

[113] Nicole Van Hoeck, Ning Ma, Lisa Ampe, Kris Baetens, Frank Van Overwalle, and Marie Vandekerckhove. Counterfactual thinking: An fmri study on changing the past for a better future. *Social Cognitive and Affective Neuroscience*, 12(9):1493–1502, 2017.

[114] Ph D Noel Hendrickson. *Counterfactual reasoning: A basic guide for analysts, strategists, and decision makers*. Lulu. com, 2011.

[115] Riccardo Crupi, Alessandro Castelnovo, Daniele Regoli, and Beatriz San Miguel Gonzalez. Counterfactual explanations as interventions in latent space. *Data Mining and Knowledge Discovery*, pages 1–37, 2022.

[116] Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harvard Journal of Law & Technology*, 31(2):841–887, 2018.

[117] Rose McDermott. On the scientific study of small samples: Challenges confronting quantitative and qualitative methodologies. *The Leadership Quarterly*, page 101675, 2023.

[118] Md Golam Moula Mehedi Hasan and Douglas A. Talbert. Counterfactual examples for data augmentation: A case study. *The International FLAIRS Conference Proceedings*, 34(1), apr 18 2021.

[119] Md Golam Moula Mehedi Hasan and Douglas Talbert. Data augmentation using counterfactuals: Proximity vs diversity. *The International FLAIRS Conference Proceedings*, 35, may 4 2022.

[120] Tommaso Dreossi, Shromona Ghosh, Xiangyu Yue, Kurt Keutzer, Alberto Sangiovanni-Vincentelli, and Sanjit A. Seshia. Counterexample-guided data augmentation. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization, 7 2018.

[121] Phillip Howard, Gadi Singer, Vasudev Lal, Yejin Choi, and Swabha Swayamdipta. Neurocounterfactuals: Beyond minimal-edit counterfactuals for richer data augmentation. 2022.

[122] Tanay Dixit, Bhargavi Paranjape, Hannaneh Hajishirzi, and Luke Zettlemoyer. Core: A retrieve-then-edit framework for counterfactual data generation. *arXiv e-prints*, pages arXiv–2210, 2022.

[123] Abbavaram Gowtham Reddy, Saloni Dash, Amit Sharma, and Vineeth N Balasubramanian. Counterfactual generation under confounding. In *NeurIPS 2022 Workshop on Causality for Real-world Impact*, 2022.

[124] Masoud Hashemi, AI Borealis, and Ali Fathi. Permuteattack: Counterfactual explanation of machine learning credit scorecards. *stat*, 1050:24, 2020.

[125] Judea Pearl. Causal inference in statistics: An overview. 2009.

[126] Ilia Stepin, Jose M. Alonso, Alejandro Catala, and Martín Pereira-Fariña. A survey of contrastive and counterfactual explanation generation methods for explainable artificial intelligence. *IEEE Access*, 9:11974–12001, 2021.

[127] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial intelligence*, 267:1–38, 2019.

[128] Jonathan Moore, Nils Hammerla, and Chris Watkins. Explaining deep learning models with constrained adversarial examples. In *PRICAI 2019: Trends in Artificial Intelligence: 16th Pacific Rim International Conference on Artificial Intelligence, Cuvu, Yanuca Island, Fiji, August 26–30, 2019, Proceedings, Part I 16*, pages 43–56. Springer, 2019.

[129] Goutham Ramakrishnan, Yun Chan Lee, and Aws Albarghouthi. Synthesizing action sequences for modifying model decisions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5462–5469, 2020.

[130] Rafael Poyiadzi, Kacper Sokol, Raul Santos-Rodriguez, Tijl De Bie, and Peter Flach. Face: feasible and actionable counterfactual explanations. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 344–350, 2020.

[131] Rory Mc Grath, Luca Costabello, Chan Le Van, Paul Sweeney, Farbod Kamiab, Zhao Shen, and Freddy Lecue. Interpretable credit application predictions with counterfactual explanations. In *NIPS 2018-Workshop on Challenges and Opportunities for AI in Financial Services: the Impact of Fairness, Explainability, Accuracy, and Privacy*, 2018.

[132] Raphael Mazzine Barbosa de Oliveira, Sofie Goethals, Dieter Brughmans, and David Martens. Unveiling the potential of counterfactuals explanations in employability. *arXiv preprint arXiv:2305.10069*, 2023.

[133] Yash Goyal, Ziyan Wu, Jan Ernst, Dhruv Batra, Devi Parikh, and Stefan Lee. Counterfactual visual explanations. In *International Conference on Machine Learning*, pages 2376–2384. PMLR, 2019.

[134] Mingli Chen, Victor Chernozhukov, Iván Fernández-Val, and Blaise Melly. Counterfactual: An r package for counterfactual analysis. *R Journal*, 9(1), 2017.

[135] Sumedha Singla, Motahhare Eslami, Brian Pollack, Stephen Wallace, and Kayhan Batmanghelich. Explaining the black-box smoothly—a counterfactual approach. *Medical Image Analysis*, 84:102721, 2023.

[136] Jeremy Zucker, Kaushal Paneri, Sara Mohammad-Taheri, Somya Bhargava, Pallavi Kolambkar, Craig Bakker, Jeremy Teuton, Charles Tapley Hoyt, Kristie Oxford, Robert Ness, and Olga Vitek. Leveraging structured biological knowledge for counterfactual inference: A case study of viral pathogenesis. *IEEE Transactions on Big Data*, 7(1):25–37, 2021.

[137] Bernard W.T. Coetzee and Kevin J. Gaston. An appeal for more rigorous use of counterfactual thinking in biological conservation. *Conservation Science and Practice*, 3(6):e409, 2021.

[138] Dylan Slack, Anna Hilgard, Himabindu Lakkaraju, and Sameer Singh. Counterfactual explanations can be manipulated. *Advances in neural information processing systems*, 34:62–75, 2021.

[139] Chris Russell. Efficient search for diverse coherent explanations. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 184–193, 2019.

[140] Fico dataset - real dataset of home equity line of credit (heloc), 2018.

[141] Arnaud Van Looveren and Janis Klaise. Interpretable counterfactual explanations guided by prototypes. pages 650–665, 2021.

[142] Solon Barocas, Andrew D Selbst, and Manish Raghavan. The hidden assumptions behind counterfactual explanations and principal reasons. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pages 80–89, 2020.

[143] Gregory Plumb, Jonathan Terhorst, Sriram Sankararaman, and Ameet Talwalkar. Explaining groups of points in low-dimensional representations. In *International Conference on Machine Learning*, pages 7762–7771. PMLR, 2020.

[144] André Artelt, Valerie Vaquet, Riza Velioglu, Fabian Hinder, Johannes Brinkrolf, Malte Schilling, and Barbara Hammer. Evaluating robustness of counterfactual explanations. In *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 01–09. IEEE, 2021.

[145] Kentaro Kanamori, Takuya Takagi, Ken Kobayashi, and Hiroki Arimura. Dace: Distribution-aware counterfactual explanation by mixed-integer linear optimization. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 4489–4495, 2020.

[146] Prasanta Chandra Mahalanobis. On the generalized distance in statistics. *Proceedings of the National Institute of Sciences - (National Institute of Science of India)*, 2:49–55, 1936.

[147] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. Lof: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 93–104, 2000.

[148] Dheeru Dua and Casey Graff. Datasets., 2017.

[149] Javier Antoran, Umang Bhatt, Tameem Adel, Adrian Weller, and José Miguel Hernández-Lobato. Getting a clue: A method for explaining uncertainty estimates. In *Proceedings of the International Conference on Learning Representations*, 2021.

[150] Hofmann Professor Dr. Hans. Statlog (german credit data) data set., 2017.

[151] Lucas De Lara, Alberto González-Sanz, and Jean-Michel Loubes. A consistent extension of discrete optimal transport maps for machine learning applications. *arXiv preprint arXiv:2102.08644*, 2021.

[152] Emily Black, Samuel Yeom, and Matt Fredrikson. Fliptest: Fairness testing via optimal transport. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, FAT20, page 111–121, New York, NY, USA, 2020. Association for Computing Machinery.

[153] Cédric Villani et al. *Optimal transport: old and new*, volume 338. Springer, 2009.

[154] Hidde Fokkema, Rianne de Heide, and Tim van Erven. Attribution-based explanations that provide recourse cannot be robust. *arXiv preprint arXiv:2205.15834*, 2022.

[155] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4765–4774. Curran Associates, Inc., 2017.

[156] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International conference on machine learning*, pages 3319–3328. PMLR, 2017.

[157] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv e-prints*, pages arXiv–1706, 2017.

[158] Bevan I Smith. Evaluating counterfactual explanations using pearl's counterfactual method. *arXiv preprint arXiv:2301.02499*, 2023.

[159] Judea Pearl and Dana Mackenzie. *The book of why: the new science of cause and effect*. Basic books, 2018.

[160] Ramaravind K Mothilal, Amit Sharma, and Chenhao Tan. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pages 607–617, 2020.

[161] Alex Kulesza, Ben Taskar, et al. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 5(2–3):123–286, 2012.

[162] Lawrence Neal, Matthew Olson, Xiaoli Fern, Weng-Keen Wong, and Fuxin Li. Open set learning with counterfactual images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 613–628, September 2018.

[163] Amir Feder, Nadav Oved, Uri Shalit, and Roi Reichart. Causalm: Causal model explanation through counterfactual language models. *Computational Linguistics*, 47(2):333–386, 2021.

[164] Martin Pawelczyk, Chirag Agarwal, Shalmali Joshi, Sohini Upadhyay, and Himabindu Lakkaraju. Exploring counterfactual explanations through the lens of adversarial examples: A theoretical and empirical analysis. In *International Conference on Artificial Intelligence and Statistics*, pages 4574–4594. PMLR, 2022.

[165] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations (ICLR)*, 2018.

[166] Qianglong Chen, Feng Ji, Xiangji Zeng, Feng-Lin Li, Ji Zhang, Haiqing Chen, and Yin Zhang. KACE: Generating knowledge aware contrastive explanations for natural language inference. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2516–2527, Online, August 2021. Association for Computational Linguistics.

[167] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal, 2015. Association for Computational Linguistics.

[168] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, 2011. Association for Computational Linguistics.

[169] Tongshuang Wu, Marco Tulio Ribeiro, Jeffrey Heer, and Daniel Weld. Polyjuice: Generating counterfactuals for explaining, evaluating, and improving models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6707–6723, Online, August 2021. Association for Computational Linguistics.

[170] Tongshuang Wu, Marco Tulio Ribeiro, Jeffrey Heer, and Daniel Weld. Polyjuice repository on github.com. Online, 2021.

[171] Bing Wang, Yan Gao, Zhoujun Li, and Jian-Guang Lou. Know what i don't know: Handling ambiguous and unanswerable questions for text-to-sql. *arXiv preprint arXiv:2212.08902*, 2022.

[172] Joshua Vendrow, Saachi Jain, Logan Engstrom, and Aleksander Madry. Dataset interfaces: Diagnosing model failures using controllable counterfactual generation. *arXiv preprint arXiv:2302.07865*, 2023.

[173] Michael V. Mannino and Murlidhar V. Koushik. The cost-minimizing inverse classification problem: a genetic algorithm approach. *Decision Support Systems*, 29(3):283–300, 2000.

[174] Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, Xavier Renard, and Marcin Detyniecki. Comparison-based inverse classification for interpretability in machine learning. In *Information Processing and Management of Uncertainty in Knowledge-Based Systems. Theory and Foundations: 17th International Conference, IPMU 2018, Cádiz, Spain, June 11-15, 2018, Proceedings, Part I 17*, pages 100–111. Springer, 2018.

[175] Jonathan Moore, Nils Hammerla, and Chris Watkins. Cadex repository on github.com. Online.

[176] Raphael Mazzine Barbosa de Oliveira and David Martens. A framework and benchmarking study for counterfactual generating methods on tabular data. *Applied Sciences*, 11(16):7274, 2021.

[177] Sahil Verma, Varich Boonsanong, Minh Hoang, Keegan E. Hines, John P. Dickerson, and Chirag Shah. Counterfactual explanations and algorithmic recourses for machine learning: A review. *arXiv preprint arXiv:2012.12962*, 2020.

[178] Stratis Tsirtsis, Abir De, and Manuel Rodriguez. Counterfactual explanations in sequential decision making under uncertainty. *Advances in Neural Information Processing Systems*, 34:30127–30139, 2021.

[179] Sahil Verma, Varich Boonsanong, Minh Hoang, Keegan E Hines, John P Dickerson, and Chirag Shah. Counterfactual explanations and algorithmic recourses for machine learning: A review. *arXiv preprint*, 2020.

[180] Greta Warren, Mark T Keane, and Ruth M J Byrne. Features of explainability: How users understand counterfactual and causal explanations for categorical and continuous features in xai, 2022.

[181] Euclides. *Os elementos*. Editora Unesp, 1ª edition, 2009.

[182] Rudolf Rucker. *Geometry, Relativity and the Fourth Dimension*. Dover Publications, 2024. Dover Books on Mathematics, Kindle Edition.

[183] Claude Bragdon. *A Primer of Higher Space*. Manas Press, Rochester, NY, 1913.

[184] Michel Marie Deza and Elena Deza. *Encyclopedia of Distances*. Springer-Verlag Berlin Heidelberg, 2009.

[185] Maysam Abedi. Non-euclidean distance measures in spatial data decision analysis: investigations for mineral potential mapping. *Annals of Operations Research*, 303(1):29–50, 2021.

[186] Jasmine Irani, Nitin Pise, and Madhura Phatak. Clustering techniques and the similarity measures used in clustering: A survey. *International journal of computer applications*, 134(7):9–14.

[187] Anil Kumar Patidar, Jitendra Agrawal, and Nishchol Mishra. Analysis of different similarity measure functions and their impacts on shared nearest neighbor clustering approach. *International Journal of Computer Applications*, 40(16):1–5.

[188] Sally Elizabeth Howe. Estimating regions and clustering spatial data: Analysis and implementation of methods using the voronoi diagram. 1980.

[189] Wagner J Alvarenga, Felipe V Campos, Alexsander CAA Costa, Turibio T Salis, Eduardo Magalhães, Luiz CB Torres, and Antonio P Braga. Time domain graph-based anomaly detection approach applied to a real industrial problem. *Computers in Industry*, 142:103714, 2022.

[190] Wagner José de Alvarenga Júnior et al. Dominating sets of the gabriel graph: na approach for one-class and online learning classifiers gabriel graph. phd. thesis. 2022.

[191] LCB Torres, CL Castro, F Coelho, F Sill Torres, and AP Braga. Distance-based large margin classifier suitable for integrated circuit implementation. *Electronics Letters*, 51(24):1967–1969, 2015.

[192] Luiz CB Torres, Cristiano L Castro, Frederico Coelho, and Antônio P Braga. Large margin gaussian mixture classifier with a gabriel graph geometric representation of data set structure. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[193] L. C. B. Torres, Frederico Coelho, C. L. Castro, and A. P. Braga. A graph of gabriel approach for large margin classifiers. In *Proceedings of the Latin American Congress on Computational Intelligence (LA-CCI) Co-located with ARGENCON*, volume 1, pages 25–29, San Carlos de Bariloche, 2014.

[194] David W Matula and Robert R Sokal. Properties of gabriel graphs relevant to geographic variation research and the clustering of points in the plane. *Geographical analysis*, 12(3):205–222, 1980.

[195] Tomáš Kaiser, Maria Saumell, and Nico Van Cleemput. 10-gabriel graphs are hamiltonian. *Information Processing Letters*, 115(11):877–881, 2015.

[196] Ahmad Biniaz, Anil Maheshwari, and Michiel Smid. Bottleneck matchings and hamiltonian cycles in higher-order gabriel graphs. *Information Processing Letters*, 153:105869, 2020.

[197] Tung-Hsin Su and Ruei-Chuan Chang. The k-gabriel graphs and their applications. In Tetsuo Asano, Toshihide Ibaraki, Hiroshi Imai, and Takao Nishizeki, editors, *Algorithms*, pages 66–75, Berlin, Heidelberg, 1990. Springer Berlin Heidelberg.

[198] Boris Aronov, Muriel Dulieu, and Ferran Hurtado. Witness gabriel graphs. *Computational Geometry*, 46(7):894–908, 2013. EuroCG 2009.

[199] David G. Kirkpatrick and John D. Radke. A framework for computational morphology. In *Computational Geometry*, volume 2 of *Machine Intelligence and Pattern Recognition*, pages 217–248. North-Holland, Amsterdam, 1985.

[200] Prosenjit Bose, Jean Cardinal, Sébastien Collette, Erik D. Demaine, Belén Palop, Perouz Taslakian, and Norbert Zeh. Relaxed gabriel graphs. In *Canadian Conference on Computational Geometry*, 2009.

[201] Michaël Aupetit and Thibaud Catz. High-dimensional labeled data analysis with topology representing graphs. *Neurocomput.*, 63:139–169, January 2005.

[202] O'Rourke Joseph. *Computational Geometry in C.* Cambridge University Press, 1993.

[203] Menelaos I. Karavelas. A robust and efficient implementation for the segment voronoi diagram. In *Proc. 1 st Int. Symp. on Voronoi Diagrams in Science and Engineering*, pages 51–62, 2004.

[204] Mark de Berg, Mark Overmars, Mark van Krefeld, and Otfried Schwarzkopf. *Computational Geometry*. Springer, Berlin, Germany, 2 edition, 1998.

[205] Hans Martin Lipp. The importance of pattern recognition for general purpose adjustment systems. In Satosi Watanabe, editor, *Methodologies of Pattern Recognition*, pages 393–407. Academic Press, 1969.

[206] Franz Aurenhammer. Voronoi diagrams—a survey of a fundamental geometric data structure. *ACM Comput. Surv.*, 23(3):345–405, September 1991.

[207] Kevin Q Brown. Voronoi diagrams from convex hulls. *Information processing letters*, 9(5):223–228, 1979.

[208] Achiya Dax. The distance between two convex sets. *Linear Algebra and its Applications*, 416(1):184–213, 2006.

[209] Teresa W Haynes, Stephen Hedetniemi, and Peter Slater. *Fundamentals of domination in graphs*. CRC press, 1998.

[210] Richard Ernest Bellman and Rand Corporation. *Dynamic programming*. Princeton University Press, 1957.

[211] Richard Ernest Bellman. *Adaptive control processes: a guided tour.* Princeton University Press, 1961.

[212] Michael H. Peters. On the shrinking volume of the hypersphere. *The College Mathematics Journal*, 46(3):178–180, may 2015.

[213] Yihua Chen, Eric K Garcia, Maya R Gupta, Ali Rahimi, and Luca Cazzanti. Similarity-based classification: Concepts and algorithms. *Journal of Machine Learning Research*, 10(3), 2009.

[214] Arti Anuragi, Dilip Singh Sisodia, and Ram Bilas Pachori. Mitigating the curse of dimensionality using feature projection techniques on electroencephalography datasets: an empirical review. *Artificial Intelligence Review*, 57(3):75, 2024.

[215] Jordi Soria-Comas and Josep Domingo-Ferrer. Mitigating the curse of dimensionality in data anonymization. In *Modeling Decisions for Artificial Intelligence: 16th International Conference, MDAI 2019, Milan, Italy, September 4–6, 2019, Proceedings 16*, pages 346–355. Springer, 2019.

[216] Maximilian Stubbemann, Tobias Hille, and Tom Hanika. Selecting features by their resilience to the curse of dimensionality. *arXiv preprint arXiv:2304.02455*, 2023.

[217] Man-Wai Mak and Sun-Yuan Kung. A solution to the curse of dimensionality problem in pairwise scoring techniques. In *International Conference on Neural Information Processing*, pages 314–323. Springer, 2006.

[218] Tomaso Poggio and Maia Fraser. Compositional sparsity of learnable functions. *Bulletin of the American Mathematical Society*, 61(3):438–456, 2024.

[219] Paul F Evangelista, Mark J Embrechts, and Boleslaw K Szymanski. Taming the curse of dimensionality in kernels and novelty detection. In *Applied soft computing technologies: The challenge of complexity*, pages 425–438. Springer, 2006.

[220] Yoshua Bengio, Olivier Delalleau, and Nicolas Le Roux. The curse of dimensionality for local kernel machines. 2005.

[221] Rainer Storn and Kenneth Price. Differential evolution: A simple and efficient adaptive scheme for global optimization over continuous spaces. *Journal of Global Optimization*, 23, 01 1995.

[222] Dheeru Dua and Casey Graff. Uci machine learning repository, 2017.

[223] J. Alcalá-Fdez, Alberto Fernández, J. Luengo, J. Derrac, and S. García. Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. *J. Multiple Valued Log. Soft Comput.*, 17:255–287, 2011.

[224] K. Hess, K. Anderson, W. Symmans, V. Valero, N. Ibrahim, J. Mejia, D. Booser, R. Theriault, A. Buzdar, P. Dempsey, R. Rouzier, N. Sneige, J. Ross, T. Vidaurre, H. Gómez, G. Hortobagyi, and L. Pusztai. Pharmacogenomic predictor of sensitivity to preoperative chemotherapy with paclitaxel and fluorouracil, doxorubicin, and cyclophosphamide in breast cancer. *Journal of clinical oncology : official journal of the American Society of Clinical Oncology*, 24(26):4236–4244, 2006.

[225] David Ardia, Kris Boudt, Peter Carl, Katharine M. Mullen, and Brian G. Peterson. Differential Evolution with DEoptim: An application to non-convex portfolio optimization. *R Journal*, 3(1):27–34, 2011.

[226] Katharine Mullen, David Ardia, David Gil, Donald Windover, and James Cline. DEoptim: An R package for global optimization by Differential Evolution. *Journal of Statistical Software*, 40(6):1–26, 2011.

[227] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. 7:1–30, 12 2006.

[228] Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991.

[229] Michael I Jordan and Robert A Jacobs. Modular and hierarchical learning systems. In *The handbook of brain theory and neural networks*, pages 579–582. 1998.

[230] Simon Haykin. *Redes Neurais: Princípios e Prática*. Bookman, 2ª edition, jan 2000.

[231] John S Bridle. Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. In *Neurocomputing: Algorithms, architectures and applications*, pages 227–236. Springer.

[232] Todd R Golub, Donna K Slonim, Pablo Tamayo, et al. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286(5439):531–537, 1999.

[233] Kenneth R Hess, Kent Anderson, W Fraser Symmans, et al. Pharmacogenomic predictor of sensitivity to preoperative chemotherapy with paclitaxel and fluorouracil, doxorubicin, and cyclophosphamide in breast cancer. *Journal of Clinical Oncology*, 24(26):4236–4244, 2006.

[234] A. Van Looveren and J. Klaise. Interpretable counterfactual explanations guided by prototypes. *arXiv*, 2019. ABILIC.

[235] J. Moore, N. Hammerla, and C. Watkins. Explaining deep learning models with constrained adversarial examples. *arXiv*, 2019. CADEX.

[236] R. Guidotti, A. Monreale, F. Giannotti, D. Pedreschi, S. Ruggieri, and F. Turini. Factual and counterfactual explanations for black box decision making. *IEEE Intell. Syst.*, 34:14–23, 2019. LORE.

[237] A. Karimi, G. Barthe, B. Balle, and I. Valera. Model-agnostic counterfactual explanations for consequential decisions. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS 2020)*, volume 108, pages 895–905, 2020. MACE.

[238] I. Afonichkin. Explaining machine learning models by generating counterfactuals. `https://aaltodoc.aalto.fi/handle/123456789/39894`, 2021. ML-Explain, accessed on 4 August 2021.

[239] C. Fernández-Loría, F. Provost, and X. Han. Explaining data-driven decisions made by ai systems: The counterfactual approach. *arXiv*, 2020. SEDC.

[240] G. Ramakrishnan, Y. C. Lee, and A. Albarghouthi. Synthesizing action sequences for modifying model decisions. *arXiv*, 2019. SynAS.

[241] Sofie Goethals, Kenneth Sörensen, and David Martens. The privacy issue of counterfactual explanations: Explanation linkage attacks. 14(5), August 2023.

[242] Shuai An and Yang Cao. Counterfactual explanation at will, with zero privacy leakage. 2(3), may 2024.

[243] Huishi Luo, Fuzhen Zhuang, Ruobing Xie, Hengshu Zhu, Deqing Wang, Zhulin An, and Yongjun Xu. A survey on causal inference for recommendation. *The Innovation*, 2024.

[244] Zhuohang Dang, Minnan Luo, Chengyou Jia, Caixia Yan, Xiaojun Chang, and Qinghua Zheng. Counterfactual generation framework for few-shot learning. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(8):3747–3758, 2023.

[245] Tobiasa Gauss, Zaneb Perkins, and Thorstenc Tjardes. Current knowledge and availability of machine learning across the spectrum of trauma science. *Current Opinion in Critical Care*, 29(6):713–721, December 2023.

[246] Thomas Melistas, Nikos Spyrou, Nefeli Gkouti, Pedro Sanchez, Athanasios Vlontzos, Giorgos Papanastasiou, and Sotirios A Tsaftaris. Benchmarking counterfactual image generation. *arXiv preprint arXiv:2403.20287*, 2024.

[247] Raphael Mazzine Barbosa de Oliveira, Kenneth Sörensen, and David Martens. A model-agnostic and data-independent tabu search algorithm to generate counterfactuals for tabular, image, and text data. *European Journal of Operational Research*, 317(2):286–302, 2024.

[248] Lisa Grabinger, Florian Hauser, and Jürgen Mottok. On the perception of graph layouts. *Journal of Software: Evolution and Process*, 36(5):e2599, 2024.

[249] Stephen M Kosslyn. Understanding charts and graphs. *Applied cognitive psychology*, 3(3):185–225, 1989.

[250] Nadia Ali and David Peebles. The effect of gestalt laws of perceptual organization on the comprehension of three-variable bar and line graphs. *Human factors*, 55(1):183–203, 2013.

[251] Sahil Verma, Varich Boonsanong, Minh Hoang, Keegan Hines, John Dickerson, and Chirag Shah. Counterfactual explanations and algorithmic recourses for machine learning: A review. *ACM Computing Surveys*, 56(12):1–42, 2024.

[252] Oran Lang, Doron Yaya-Stupp, Ilana Traynis, Heather Cole-Lewis, Chloe R. Bennett, Courtney R. Lyles, Charles Lau, Michal Irani, Christopher Semturs, Dale R. Webster, Greg S. Corrado, Avinatan Hassidim, Yossi Matias, Yun Liu, Naama Hammel, and Boris Babenko. Using generative ai to investigate medical imagery models and datasets. *eBioMedicine*, 102:105075, 2024.

[253] Raphael Mazzine Barbosa de Oliveira and David Martens. A framework and benchmarking study for counterfactual generating methods on tabular data. *Applied Sciences*, 11(16):7274, 2021.

[254] Maryam Hashemi, Ali Darejeh, and Francisco Cruz. Understanding user preferences in explainable artificial intelligence: A survey and a mapping function proposal. *arXiv preprint arXiv:2302.03180*, 2023.

[255] Barry Smyth and Mark T. Keane. A few good counterfactuals: Generating interpretable, plausible and diverse counterfactual explanations. In Mark T. Keane and Nirmalie Wiratunga, editors, *Case-Based Reasoning Research and Development*, pages 18–32, Cham, 2022. Springer International Publishing.

[256] Dieter Brughmans, Lissa Melis, and David Martens. Disagreement amongst counterfactual explanations: how transparency can be misleading. *TOP*, 32(3):429–462, October 2024.

[257] Peter E. Hart. Progress on a computer based consultant. In *Proceedings of the 4th Int. Joint Conf. on Artificial Intelligence - Volume 1*, IJCAI'75, page 831–841, San Francisco, CA, USA, 1975. Morgan Kaufmann Publishers Inc.

[258] Randall Davis, Bruce Buchanan, and Edward Shortliffe. Production rules as a representation for a knowledge-based consultation program. *Artif. Intell.*, 8(1):15–45, feb 1977.

[259] Johanna D Moore and William R Swartout. Explanation in expert systems: A survey. Technical report, University of Southern California - Marina Del Rey Information Sciences Inst, 1988.

[260] Saito and Nakano. Medical diagnostic expert system based on pdp model. In *IEEE 1988 Int. Conf. on Neural Networks*, pages 255–262 vol.1, July 1988.

[261] Geoffrey E. Hinton. *Learning distributed representations of concepts*. Parallel distributed processing: Implications for psychology and neurobiology. Clarendon Press/Oxford University Press, 1989.

[262] R.J. Machado. Handling knowledge in high order neural networks: the combinatorial neural model. In *International 1989 Joint Conference on Neural Networks*, pages 582 vol.2–, 1989.

[263] Geoffrey G. Towell and Jude W. Shavlik. Extracting refined rules from knowledge-based neural networks. *Machine Learning*, 13(1):71–101, Oct 1993.

[264] C. L. Giles and C. W. Omlin. Rule refinement with recurrent neural networks. In *IEEE Int. Conf. on Neural Networks*, pages 801–806 vol.2, March 1993.

[265] Hongjun Lu, Rudy Setiono, and Huan Liu. Neurorule: A connectionist approach to data mining. In *Proceedings of the 21th Int. Conf. on Very Large Data Bases*, VLDB '95, page 478–489, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.

[266] Mark W. Craven and Jude W. Shavlik. Extracting comprehensible concept representations from trained neural networks. In *In: Working Notes on the IJCAI'95 Workshop on Comprehensibility in Machine Learning*, pages 61–75, 1995.

[267] Robert S. Baron. Knowledge extraction from neural networks:a survey. 1995.

[268] Mark W. Craven. *Extracting Comprehensible Models From Trained Neural Networks*. PhD thesis, University of Wisconsin, Madson, USA, 1996.

[269] Mark Craven and Jude Shavlik. Rule extraction: Where do we go from here? Technical Report 1, University of Wisconsin Machine - Learning Research Group, Madison, Wisconsin - USA, 7 1999.

[270] Ahamed Salihe Kuttiyil. Survey of rule extraction methods. Master's thesis, Wayne State University, Detroit, Michigan, 5 2004. An optional note.

[271] Henrik Jacobsson. Rule extraction from recurrent neural networks: A taxonomy and review. *Neural Comput.*, 17(6):1223–1263, June 2005.

[272] Sebastian Bader and Pascal Hitzler. *Dimensions of Neural-symbolic Integration - A Structured Survey*, volume 1, page 167–194. Int. Federation for Computational Logic, College Publications, London, UK, 2005.

[273] K. Kumar Sethi, D. Kumar Mishra, and B. Mishra. Extended Taxonomy of Rule Extraction Techniques and Assessment of KDRuleEx. *Int. Journal of Computer Applications*, 50(21):25–31, July 2012.

[274] M. G. Augasta and T. Kathirvalavakumar. Rule extraction from neural networks — a comparative study. In *Int. Conf. on Pattern Recognition, Informatics and Medical Engineering (PRIME-2012)*, pages 404–408, March 2012.

[275] Tameru Hailesilassie. Rule Extraction Algorithm for Deep Neural Networks: A Review . *(IJCSIS) Int. Journal of Computer Science and Information Security*, 14(7):376–381, July 2016.

[276] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. A survey of methods for explaining black box models. *ACM Comput. Surv.*, 51(5):93:1–93:42, August 2018.

[277] L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, and L. Kagal. Explaining explanations: An overview of interpretability of machine learning. In *2018 IEEE 5th Int. Conf. on Data Science and Advanced Analytics (DSAA)*, pages 80–89, Oct 2018.

[278] Qinglong Wang, Kaixuan Zhang, Alexander G. Ororbia, II, Xinyu Xing, Xue Liu, and C. Lee Giles. An empirical evaluation of rule extraction from recurrent neural networks. *Neural Comput.*, 30(9):2568–2591, September 2018.

[279] MatÄ›j Fanta. Rules extraction from deep neural networks - master thesis. Master's thesis, Czech Technical University In Prague - Faculty of Nuclear Sciences and Physical Engineering, Prague, 08 2019.

[280] J. Townsend, T. Chaton, and J. M. Monteiro. Extracting relational explanations from deep neural networks: A survey from a neural-symbolic perspective. *IEEE Trans. on Neural Networks and Learning Systems*, pages 1–15, 2019.

[281] Plamen P Angelov, Eduardo A Soares, Richard Jiang, Nicholas I Arnold, and Peter M Atkinson. Explainable artificial intelligence: an analytical review. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 11(5):e1424, 2021.

[282] Andreas Theissler, Francesco Spinnato, Udo Schlegel, and Riccardo Guidotti. Explainable ai for time series classification: A review, taxonomy and research directions. *IEEE Access*, 2022.

[283] Rami Ibrahim and M Omair Shafiq. Explainable convolutional neural networks: A taxonomy, review, and future directions. *ACM Computing Surveys*, 2022.

[284] Theo Araujo, Natali Helberger, Sanne Kruikemeier, and Claes H De Vreese. In ai we trust? perceptions about automated decision-making by artificial intelligence. *AI & SOCIETY*, pages 1–13, 2020.

[285] Erik Brynjolfsson and ANDREW Mcafee. The business of artificial intelligence:how ai fits into your data science team: What it can — and cannot — do for your organization. *Harvard Business Review*, pages 1–20, 2017.

[286] Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Proceedings of the 31st Int. Conf. on Neural Information Processing Systems*, NIPS'17, page 4768–4777, Red Hook, NY, USA, 2017. Curran Associates Inc.

[287] Ajay Chander, Ramya Srinivasan, Suhas Chelian, Jun Wang, and Kanji Uchino. Working with beliefs: Ai transparency in the enterprise. In *Fujitsu Laboratories of America Reports*, 03 2019.

[288] Imran Ahmed, Gwanggil Jeon, and Francesco Piccialli. From artificial intelligence to explainable artificial intelligence in industry 4.0: A survey on what, how, and where. *IEEE Trans. on Industrial Informatics*, 18(8):5031–5042, 2022.

[289] Jonas Paul Wanner. *Artificial Intelligence for Human Decision-Makers: Systematization, Perception, and Adoption of Intelligent Decision Support Systems in Industry 4.0.* doctoralthesis, Universität Würzburg, 2022.

[290] Arzam Kotriwala, Benjamin Klöpper, Marcel Dix, Gayathri Gopalakrishnan, Dawid Ziobro, and Andreas Potschka. Xai for operations in the process industry-applications, theses, and research directions. In *AAAI Spring Symposium: Combining Machine Learning with Knowledge Engineering*, 2021.

[291] Robert Andrews, Joachim Diederich, and Alan Tickle. Survey and critique of techniques for extracting rules from trained artificial neural networks. *Knowledge-Based Systems*, 6:373–389, 12 1995.

[292] Alan B Tickle, Robert Andrews, Mostefa Golea, and Joachim Diederich. The truth will come to light: Directions and challenges in extracting the knowledge embedded within trained artificial neural networks. *IEEE Transactions on Neural Networks*, 9(6):1057–1068, 1998.

[293] Finale Doshi Velez and Been Kim. Considerations for evaluation and generalization in interpretable machine learning. *Explainable and interpretable models in computer vision and machine learning*, pages 3–17, 2018.

[294] Kacper Sokol and Peter Flach. Explainability fact sheets: a framework for systematic assessment of explainable approaches. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 56–67, 2020.

[295] Giulia Vilone and Luca Longo. Notions of explainability and evaluation approaches for explainable artificial intelligence. *Information Fusion*, 76:89–106, 2021.

[296] Carlos Zednik. Solving the black box problem: A normative framework for explainable artificial intelligence. *Philosophy & technology*, 34(2):265–288, 2021.

[297] Heimo Müller, Michaela Kargl, Markus Plass, Bettina Kipperer, Luka Brcic, Peter Regitnig, Christian Geißler, Tobias Küster, Norman Zerbe, and Andreas Holzinger. Towards a taxonomy for explainable ai in computational pathology. *Humanity Driven AI: Productivity, Well-being, Sustainability and Partnership*, pages 311–330, 2022.

[298] Yiqiao Li, Jianlong Zhou, Sunny Verma, and Fang Chen. A survey of explainable graph neural networks: Taxonomy and evaluation metrics. *arXiv preprint arXiv:2207.12599*, 2022.

[299] Xuezhong Zhang, Libin Dai, Qingming Peng, Ruizhi Tang, and Xinwei Li. A survey of counterfactual explanations: Definition, evaluation, algorithms, and applications. In Ning Xiong, Maozhen Li, Kenli Li, Zheng Xiao, Longlong Liao, and Lipo Wang, editors, *Advances in Natural Computation, Fuzzy Systems and Knowledge Discovery*, pages 905–912, Cham, 2023. Springer International Publishing.

[300] Edward H. Shortliffe and Bruce G. Buchanan. A model of inexact reasoning in medicine. *Mathematical Biosciences*, 23(3):351 – 379, 1975.

[301] Rakesh Agrawal, Tomasz Imielinski, and Arun Swami. *Mining Association Rules Between Sets of Items in Large Databases, SIGMOD Conf.*, volume 22, pages 207–217. 06 1993.

[302] Petr Hájek and Tomás Havránek. *Mechanizing Hypothesis Formation: Mathematical Foundations for a General Theory*. Publisher: Publisher: Springer-Verlag Berlin and Heidelberg. New York., 1978.

[303] Jan Rauch. *Observational Calculi and Association Rules*. Publisher: Springer-Verlag Berlin and Heidelberg Gmbh, 01 2012.

[304] Lukasz A. Kurgan and Krzysztof J. Cios. Caim discretization algorithm. *IEEE Trans. on Knowl. and Data Eng.*, 16(2):145–153, February 2004.

[305] Michael Hahsler, Christian Buchta, Bettina Gruen, Kurt Hornik, Christian Borgelt, Ian Johnson, and Makhlouf Ledmi. *R Documentation - Package arules: Mining Association Rules and Frequent Itemsets*. Michael Hahsler - R Foundation for Statistical Computing - Powered by DataCamp, 2022.

[306] Y. Li, S. Zhang, Y. Yin, J. Zhang, and Z. Xu. Quantitative association rule mining for blast furnace production data. In *2017 Chinese Automation Congress (CAC)*, pages 2275–2280, 2017.

[307] Ze-kai Cheng, Qi Zhang, Qin Feng, and Li-qin Zhu. Analysis of blast furnace data based on association rules. *DEStech Trans. on Engineering and Technology Research*, 2:401–407, 2017.

[308] Bernard Grabot. Rule mining in maintenance: Analysing large knowledge bases. *Computers and Industrial Engineering*, 139:105501, 2020.

[309] Hsiao-Kang Lin, Cheng-Huan Hsieh, Nai-Chieh Wei, and Yi-Chun Peng. Association rules mining in r for product performance management in industry 4.0. *Procedia CIRP*, 83:699 – 704, 2019. 11th CIRP Conf. on Industrial Product-Service Systems.

[310] F. Wang. Application of multidimensional association rule techniques in manufacturing resource planning system. In *2010 Seventh Int. Conf. on Fuzzy Systems and Knowledge Discovery*, volume 3, pages 1433–1437, 2010.

[311] Bernard Kamsu-Foguem, Fabien Rigal, and Félix Mauget. Mining association rules for the quality improvement of the production process. *Expert Systems with Applications*, 40(4):1034 – 1045, 2013.

[312] Mohammed J. Zaki. Scalable algorithms for association mining. *IEEE Trans. on Knowl. and Data Eng.*, 12(3):372–390, May 2000.

[313] Jiawei Han, Jian Pei, and Yiwen Yin. Mining frequent patterns without candidate generation. *SIGMOD Rec.*, 29(2):1–12, May 2000.

[314] Ali Mohammed, Kamarudin Ambak, Ahmed Mosa, and Deprizon Syamsunur. Expert system in engineering transportation: A review. *Journal of Engineering Science and Technology*, 14:229–252, 01 2019.

[315] Tarun Gupta and Biman K Ghosh. A survey of expert systems in manufacturing and process planning. *Computers in Industry*, 11(2):195 – 204, 1989.

[316] Z. Z. Zhang, G. S. Hope, and O. P. Malik. Expert systems in electric power systems-a bibliographical survey. *IEEE Trans. on Power Systems*, 4(4):1355–1362, 1989.

[317] LA Miller, E Groundwater, JE Hayes, SM Mirsky, and L Beltracchi. Guidelines for verification and validation of expert systems in the nuclear industry. volume 2, survey and assessment of conventional software verification and validation methods. Technical report, Electric Power Research Inst., Palo Alto, CA (United States); Science ..., 1993.

[318] Spyros Tzafestas. *Expert Systems in Engineering Applications*, volume 1. Springer-Verlag Berlin Heidelberg, Berlin, Germany, 3 edition, 7 1993.

[319] Boon Toh Low, Chun Hung Cheng, Jaideep Motwani, and Manu S Madan. Expert systems in the service industry: a comprehensive survey and an application. *Int. Journal of Computer Applications in Technology*, 10(5-6):289–299, 1997.

[320] PI System. `https://www.osisoft.com/pi-system/`. Accessed: Month Day, Year.

[321] John von Neumann and Oskar Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, 1944.

[322] A. Messalas, Y. Kanellopoulos, and C. Makris. Model-agnostic interpretability with shapley values. In *2019 10th Int. Conf. on Information, Intelligence, Systems and Applications (IISA)*, pages 1–7, 2019.

[323] Scott M Lundberg, Bala Nair, Monica S Vavilala, Mayumi Horibe, Michael J Eisses, Trevor Adams, David E Liston, Daniel King-Wai Low, Shu-Fang Newman, Jerry Kim, et al. Explainable machine-learning predictions for the prevention of hypoxaemia during surgery. *Nature Biomedical Engineering*, 2(10):749, 2018.

[324] Scott M. Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M. Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. From local explanations to global understanding with explainable ai for trees. *Nature Machine Intelligence*, 2(1):2522–5839, 2020.

[325] Przemyslaw Biecek and Tomasz Burzykowski. *Explanatory Model Analysis*. Chapman and Hall/CRC, New York, 2021.

[326] Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and

Édouard Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[327] Marcelo Versiani Carneiro, Turibio Tanus Salis, Gustavo M. Almeida, and Antonio P. Braga. Prediction of mechanical properties of steel tubes using a machine learning approach. *Journal of Materials Engineering and Performance*, 30:434—-443, 2021.

[328] Ana Cristina Lima Gomes, Martín Gómez Ravetti, and Eduardo G. Carrano. Multi-objective matheuristic for minimization of total tardiness and energy costs in a steel industry heat treatment line. *Computers and Industrial Engineering*, 151:106929, 2021.

[329] Ayman Moawad, Ehsan Islam, Namdoo Kim, Ram Vijayagopal, Aymeric Rousseau, and Wei Biao Wu. Explainable ai for a no-teardown vehicle component cost estimation: A top-down approach. *arXiv e-prints*, pages arXiv–2006, 2020.

[330] Thomas Mckinley, Meghana Somwanshi, Devawrat Bhave, and Sandeep Verma. Identifying nox sensor failure for predictive maintenance of diesel engines using explainable ai. In *PHM Society European Conf.*, volume 5, 2020.

[331] Tuna Acisu, Soh Yee Lee, and Almut Scheerer. Interpretable ai for business applications. *Data Innovation Lab*, 1:1–31, 2020.

[332] Tiemo Thiess. *Designing Artificial Intelligence Systems for B2B Aftersales Decision Support*. PhD thesis, IT University of Copenhagen, Copenhagen, Denmark., 7 2020. An optional note.

[333] Leo Carlsson, Peter Samuelsson, and Pär Jönsson. Interpretable machine learning – tools to interpret the predictions of a machine learning model predicting the electrical energy consumption of an electric arc furnace. *steel research Int.*, 91:1–10, 11 2020.

[334] Dellainey Rosita Alcaçoas. Anomaly detection in ring rolling process using tree ensemble methods. Master's thesis, University Of Skövde, Skövde, Sweden, 4 2019.

[335] Ngoc Yen Nhi Vo. *Machine Learning Algorithms for Wealth Data Analytics*. PhD thesis, University Of Technology Sydney - Faculty Of Engineering And Information Technology, Sydney, Australia, 7 2020.

[336] Chris Aldrich. Process variable importance analysis by use of random forests in a shapley regression framework. *minerals*, 10(5), 2020.

[337] R. Galanti, B. Coma-Puig, M. d. Leoni, J. Carmona, and N. Navarin. Explainable predictive process monitoring. In *2020 2nd Int. Conf. on Process Mining (ICPM)*, pages 1–8, 2020.

[338] Bing Kong, Zhuoheng Chen, Shengnan Chen, and Tianjie Qin. Machine learning-assisted production data analysis in liquid-rich duvernay formation. *Journal of Petroleum Science and Engineering*, 200:108377, 2021.

[339] Karthik Sekaran and Shanmugam S. Interpreting the factors of employee attrition using explainable ai. In *2022 Int. Conf. on Decision Aid Sciences and Applications (DASA)*, pages 932–936, 2022.

[340] Lucas C Brito, Gian Antonio Susto, Jorge N Brito, and Marcus AV Duarte. An explainable artificial intelligence approach for unsupervised fault detection and diagnosis in rotating machinery. *Mechanical Systems and Signal Processing*, 163:108105, 2022.

[341] Luisa Macedo, Luis Miguel Matos, Paulo Cortez, André Domingues, Guilherme Moreira, and André Pilastri. A machine learning approach for spare parts lifetime estimation. In *14th Int. Conf. on Agents and Artificial Intelligence - ICAART*, volume 3, pages 765–772. SciTePress, 2022.

[342] Nijat Mehdiyev and Peter Fettke. Local post-hoc explanations for predictive process monitoring in manufacturing. *CoRR*, abs/2009.10513, 2020.

[343] Julian Senoner, Torbjørn Netland, and Stefan Feuerriegel. Using explainable artificial intelligence to improve process quality: Evidence from semiconductor manufacturing. *Management Science*, 2021.

[344] Ahmad Kamal Mohd Nor, Srinivasa Rao Pedapati, Masdi Muhammad, and Víctor Leiva. Abnormality detection and failure prediction using explainable bayesian deep learning: Methodology and case study with industrial data. *Mathematics*, 10(4):554, 2022.

[345] Jungsu Park, Woo Hyoung Lee, Keug Tae Kim, Cheol Young Park, Sanghun Lee, and Tae-Young Heo. Interpretation of ensemble learning to predict water quality using explainable artificial intelligence. *Science of The Total Environment*, 832:155070, 2022.

[346] Sarthak Manas Tripathy, Ashish Chouhan, Marcel Dix, Arzam Kotriwala, Benjamin Klöpper, and Ajinkya Prabhune. Explaining anomalies in industrial multivariate time-series data with the help of explainable ai. In *2022 IEEE Int. Conf. on Big Data and Smart Computing (BigComp)*, volume 1, pages 226–233, 2022.

[347] Thomas Rojat, Raphaël Puget, David Filliat, Javier Del Ser, Rodolphe Gelin, and Natalia Díaz Rodríguez. Explainable artificial intelligence (XAI) on timeseries data: A survey. *CoRR*, abs/2104.00950, 2021.

[348] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.

[349] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. *Parallel distributed processing: explorations in the microstructure of cognition*, 1:318–362, 1986.

[350] Berend Jan van der Zwaag, Kees Slump, Lambert Spaanenburg, and M Young. Extracting knowledge from neural networks in image processing. *University of Twente, Enschede, the Netherlands and Lund University, Lund, SweM den*, 1989.

[351] David A Handelman, Stephen H Lane, and JJ Gelfand. Integration of knowledge-based system and neural network techniques for autonomous learning machines. In *Proceedings of the International Joint Conference on Neural Networks IJCNN*, volume 89, pages 683–688, 1989.

[352] Luis E. Zárate, S. Mariano Dias, and M.A. Junho Song. Fcann: A new approach for extraction and representation of knowledge from ann trained via formal concept analysis. *Neurocomputing*, 71(13):2670–2684, 2008. Artificial Neural Networks (ICANN 2006) / Engineering of Intelligent Systems (ICEIS 2006).

[353] Jiongyu Guo, Defang Chen, and Can Wang. Alignahead: Online cross-layer knowledge extraction on graph neural networks. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2022.

[354] Sebastian Iglesias. A trepan (neural network inductive learning algorithm) implementation in java. github, 10 2018.

[355] Mark Craven. An implementation of the trepan algorithm in python. trepan extracts a decision tree from an ann using a sampling method. Professor Mark Craven's website at University Of Wisconsin–Madison. Biostatistics and Medical Informatics - School of Medicine and Public Health, 07 2000.

[356] Aneesh Barthakur. An implementation of the trepan algorithm in python. trepan extracts a decision tree from an ann using a sampling method. github, 05 2019.

[357] Christoph Molnar. *R Documentation - TreeSurrogate: Decision tree surrogate model*. Christoph Molnar - R Foundation for Statistical Computing - Powered by DataCamp, 2022.

[358] Claudia Milaré, Andre de Carvalho, and Maria-Carolina Monard. Extracting knowledge from artificial neural networks: An empirical comparison of trepan and symbolic learning algorithms. pages 272–281, 04 2002.

[359] Tanusree De, Prasenjit Giri, Ahmeduvesh Mevawala, Ramyasri Nemani, and Arati Deo. Explainable ai: A hybrid approach to generate human-interpretable explanation for deep learning prediction. *Procedia Computer Science*, 168:40–48, 2020.

[360] R. Confalonieri, T. Weyde, T. R. Besold, and F. Moscoso del Prado Martín. Trepan reloaded: A knowledge-driven approach to explaining artificial neural networks. In *24th European Conf. on Artificial Intelligence*, volume 325, pages 2457–2464. IOS Press, 2020.

[361] Roberto Confalonieri, Guendalina Righetti, Pietro Galliani, Nicolas Toquard, Oliver Kutz, and Daniele Porello. Towards knowledge-driven distillation and explanation of black-box models. In *Proc. of the Workshop on Data meets Applied Ontologies in Explainable AI*. September 2021.

[362] M. Sato and H. Tsukimoto. Rule extraction from neural networks via decision tree induction. In *IJCNN'01. Int. Joint Conf. on Neural Networks. Proceedings (Cat. No.01CH37222)*, volume 3, pages 1870–1875 vol.3, July 2001.

[363] J. Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[364] Jan Ruben Zilke. Extracting rules from deep neural networks. Master's thesis, Technische Universität Darmstadt, Darmstadt, Germany, 2016.

[365] J. Shotton, Sebastian Nowozin, Toby Sharp, John Winn, P. Kohli, and Antonio Criminisi. Decision jungles: Compact and rich models for classification. *Advances in Neural Information Processing Systems*, 01 2013.

[366] Joost de Nijs. Decision dags – a new approach. In *Computer Science - Brown University Master Theses*, 2011.

[367] Andrejs Bondarenko. *Development Of Knowledge Extraction Methodology From Trained Artificial Neural Networks*. PhD thesis, Riga Technical University. Faculty of Computer Science and Information Technology - Institute of Information Technology, Riga, Latvia. 1 Setas Street, Room 202., 05 2020.

[368] Andrey Bondarenko and Ludmila Aleksejeva. Workflow for knowledge extraction from neural network classifiers. pages 1–6, 10 2018.

[369] Andrey Bondarenko. Knowledge extraction from trained artificial neural networks (mlp's) in the form of binary classification decision tree. contains sensitivity based pruning (with retraining). https://bitbucket.org/, 05 2020.

[370] Andrey Bondarenko. Knowledge extraction from trained artificial neural networks - pyhton (nnkx-py). https://bitbucket.org/, 05 2020.

[371] M. R. A. Iqbal. Eclectic rule extraction from neural networks using aggregated decision trees. In *2012 7th Int. Conf. on Electrical and Computer Engineering*, pages 129–132, Dec 2012.

[372] G Weckman, W Young, S Hernández, M Rangwala, and V Ghai. Extracting knowledge from carbon dioxide corrosion inhibition with artificial neural networks. *Int J Indust Eng*, 17(1):69–79, 2010.

[373] Deepika Bhalla, Raj Kumar Bansal, and Hari Om Gupta. Function analysis based rule extraction from artificial neural networks for transformer incipient fault diagnosis. *Int. Journal of Electrical Power & Energy Systems*, 43(1):1196–1203, 2012.

[374] Luis E. Zárate and Sérgio M. Dias. Qualitative behavior rules for the cold rolling process extracted from trained ann via the fcann method. 22(4–5):718–731, June 2009.

[375] Senhui Wang, Haifeng Li, Yongjie Zhang, and Zongshu Zou. An integrated methodology for rule extraction from elm-based vacuum tank degasser multiclassifier for decision-making. *Energies*, 12(18), 2019.

[376] GR Weckman, DF Millie, C Ganduri, M Rangwala, W Young, M Rinder, and GL Fahnenstiel. Knowledge extraction from the neural black box in ecological monitoring. 3(1):38–55, 2009.

[377] Ichiro Shigaki and Hiroshi Narazaki. A machine-learning approach for a sintering process using a neural network. *Production Planning & Control*, 10(8):727–734, 1999.

[378] Shaohan Chen and Chuanhou Gao. Linear priors mined and integrated for transparency of blast furnace black-box svm model. *IEEE Trans. on Industrial Informatics*, 16(6):3862–3870, 2020.

[379] Qiaomu Shen, Yanhong Wu, Yuzhe Jiang, Wei Zeng, Alexis K H LAU, Anna Vianova, and Huamin Qu. Visual interpretation of recurrent neural network on multi-dimensional time-series forecast. In *2020 IEEE Pacific Visualization Symposium (PacificVis)*, pages 61–70, 2020.

[380] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[381] Jason Lines, Sarah Taylor, and Anthony Bagnall. Hive-cote: The hierarchical vote collective of transformation-based ensembles for time series classification. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 1041–1046, 2016.

[382] Diego Cabrera, Fernando Sancho, Mariela Cerrada, René-Vinicio Sánchez, and Chuan Li. Knowledge extraction from deep convolutional neural networks applied to cyclo-stationary time-series classification. *Information Sciences*, 524:1–14, 2020.

[383] Sebastian D Goodfellow, Andrew Goodwin, Robert Greer, Peter C Laussen, Mjaye Mazwi, and Danny Eytan. Towards understanding ecg rhythm classification using convolutional neural networks and attention mappings. In *Machine learning for healthcare conference*, pages 83–101. PMLR, 2018.

[384] Roy Assaf and Anika Schumann. Explainable deep neural networks for multivariate time series predictions. In *IJCAI*, pages 6488–6490, 2019.

[385] Jingyuan Wang, Ze Wang, Jianfeng Li, and Junjie Wu. Multilevel wavelet decomposition network for interpretable time series analysis. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2437–2446, 2018.

[386] En-Yu Hsu, Chien-Liang Liu, and Vincent S. Tseng. Multivariate time series early classification with interpretability using deep learning and attention mechanism. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 541–553. Springer, 2019.

[387] Udo Schlegel, Arnout Hiba, Mennatallah El-Assady, Daniela Oelke, and Daniel A Keim. Towards a rigorous evaluation of xai methods on time series. In *2019 IEEE/CVF Int. Conf. on Computer Vision Workshop (ICCVW)*, pages 4197–4201. IEEE, 2019.

[388] Ioana Giurgiu and Anika Schumann. Explainable failure predictions with rnn classifiers based on time series data. *arXiv preprint arXiv:1901.08554*, 2019.

[389] Kevin Fauvel, Elisa Fromont, Véronique Masson, Philippe Faverdin, and Alexandre Termier. Xem: An explainable-by-design ensemble method for multivariate time series classification. *Data Mining and Knowledge Discovery*, 36(3):917–957, 2022.

[390] Michael C. Thrun, Alfred Ultsch, and Lutz Breuer. Explainable ai framework for multivariate hydrochemical time series. *Machine Learning and Knowledge Extraction*, 3(1):170–204, 2021.

[391] Mohsin Munir, Shoaib Siddiqui, Ferdinand Küsters, Dominique Mercier, Andreas Dengel, and Sheraz Ahmed. Tsxplain: Demystification of dnn decisions for time-series using natural language and statistical features. In *Conf.: Int. Conf. on Artificial Neural Networks*, pages 426–439, 09 2019.

[392] S. A. Siddiqui, D. Mercier, M. Munir, A. Dengel, and S. Ahmed. Tsviz: Demystification of deep learning models for time-series analysis. *IEEE Access*, 7:67027–67040, 2019.

Appendix A

XAI Brief History

A.1 Explanation in artificial intelligence

The knowledge extraction and explanation is a subject that has been investigated since the rise of classical AI in the sixties and seventies([257–259]) following the great expansion of the subject Artificial Neural Networks (ANN)in the 1980s [260–262] until the mid-1990s [263–266]. From then on, the theme reached a vast and diverse amount of work, as can be seen in the *survey* and comparisons publications: [4, 16, 267–282]. Notably, in the past ten years, AI has presented a new boost, primarily due to the development of deep networks in a field called Deep Learning (DL). DL also has received attention of the Exaplanibility enthusiasts [283].

In the present days, IA is already widespread in people's daily lives, including automatically making decisions that affect their lives [13, 284]. Given this fact, providing accounts of the behavior of the models becomes relevant and the claims for explainability emerged from different entities.

A.2 AI dissemination and the claims for accountability: XAI

Artificial Intelligence, in particular Machine Learning, already figures as the leading General Purpose Technology (GPT) of the present days [285]. Given this context, accountability for the ML models' behavior becomes relevant. Comprehending why models make a particular prediction can be as crucial as their prediction accuracy [286]. Understanding model decision-making reasoning becomes necessary when choosing an AI tool [287].

It is worth mentioning that the explainability subject is also involved in social, ethical [19], and legal issues [20, 21]. Regulations regarding AI models' decision justification already have drawn the attention of geopolitical leadership centers legislation, for instance, the General Data Protection Regulation (GDPR) from European Union [22]; the Code of Federal Regulations, and the specific executive order on AI in the USA [23, 24]; and The Personal Data Protection Law in Brazil [25]. Such regulations impact the activities of machine learning professionals who must meet the requirements that now have law power. While these laws can impose challenges, they also provide opportunities for ML researchers to take the lead in designing algorithms that supply explanation features [26].

Regarding the industrial environment, the justification and the models' decision-making process comprehension are crucial to guarantee trust in the results. Many professionals, including engineers and operators, must trust their methods and tools [10]. The experience in the industrial shop floor shows that if people do not trust a system, they usually stop using it and start to adopt parallel solutions. Explainability appears as a way to enhance user trust in the models [288].

Besides trusting in the models, and their decision process, there is also a technological requirement regarding connectivity and fluidity in human-AI interaction [13, 289, 290]. AI systems must communicate their results and intentions effectively, giving justifications for their conclusions to the people who share the informatics ecosystem [13].Such fluency in human-AI interaction becomes relevant in a high-connected network. For instance, in an Industry 4.0 (I4.0) environment, where machines, systems, and people work collaboratively, producing and sharing information, seeking to optimize several tasks. Furthermore, the model's result explainability is a prerequisite to guarantee scientific value to the investigation results [16].

Such demand for "AI-explanation ability" has produced a particular study field denominated Explainable Artificial Intelligence (XAI). XAI area has received growing attention, as one can see based on the quantity of works and promoted events involving the topic. Nowadays, XAI researches aim to meet demands from different research fields.

A.3 XAI main objectives

Currently,XAI researchers seek to meet different demands from different audiences [4, 15, 16]. The main needs involves:

- Justifications seeking to support the results (reliability, acceptance).
- Models Training improvement (identify avoid biases, errors).
- Elucidate findings and formulate hypotheses, causality (automatic knowledge discovery).
- Law, ethics, transparency and justice (justify outputs and propose feasible actions to change them).
- Interactivity targeting to facilitate human-AI mutual understanding. (Human-in-the-loop good fluidity).
- Simulate different scenarios ("possible worlds").

When using AI to solve problems, there is often a trade-off between getting accurate predictions and understanding how the AI model arrived at its conclusion. This means that sometimes, in order to get the most accurate results, we have to sacrifice our ability to explain exactly how the AI arrived at those results.

When working with complex data and functions, it is often assumed that a more complicated model will result in more accurate predictions, especially when dealing with time series and images. However, the behavior of these complex models can be complicated for people to understand. Another XAI objective is helping to make these models more interpretable, assisting, this way, in the Accuracy Vs interpretability trade-off.

Figure A.1 shows some popular models and their position within the Accuracy-Interpretability trade-off.

Such investigations concerning XAI usually deal with some commonalities. It is important to know these aspects for a better understanding of the topics discussed later.

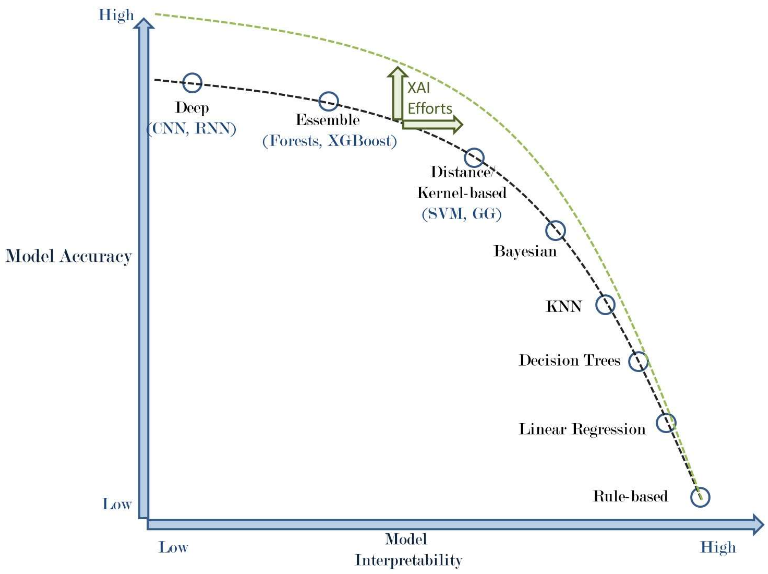


Figure A.1: Accuracy x Interpretability trade-off and XAI's researchers objectives [4]

A.4 XAI essential concepts

As AI and XAI fields continuing to expand, it accumulate different concepts, ideas, and terms associated with them. To help organize the vast and complex landscape regarding XAI, various taxonomies have been proposed to organize and classify the main elements [1, 4, 126, 271, 273, 276, 277, 281–283, 291–299]

Analyzing these different works, it is possible to identify some essential concepts when investigating methods of AI implicit knowledge explanation. In general, it is about the kind of model, how to manipulate the models' acquired knowledge, how to expose the knowledge explanation to users, the kind of explanation (which question to answer), and finally, how to verify whether that whole explanation process had performed correctly (explanation outputs quality).

Appendix B

AI Explaination: Methods and Experimentation

The present Appendix section describes the main XAI methods' operating principles and provides a practical demonstration of their application utilizing real-world databases. Additionally, some others actual applications are listed. For organizational purposes, the addressed methods are separated into two large distinct groups: (1) Natural Explainability Models (Transparent Models) and (2) Post-Hoc Explainability Methods. Due to the importance of time series to industry, engineering and biotech areas, the work also presents a section devoted to reviewing time series-based XAI methods.

B.1 Transparent Models - Rule-based models

Observational Calculi - Association rules mining

Pioneering works related to the automatic generation of hypotheses by computational means date back to the 1960s and involved causal relationships between binary categorical variables [8]. In the seventies, researchers implemented a method to calculate probabilistic metrics of propositional rules. Such a method was applied to support an automatic diagnostic system [300]. Such approaches reached their biggest icon in the nineties, with the Apriori algorithm [301]. Apriori is typically adopted in association rule mining in sales transaction analysis (*market basket analysis*).

For databases containing binary categorical variables, there is even a mathematical formalism called *Observacional Calculi* that guides the process of extracting hypotheses [302, 303]. Such work has proposed several metrics to quantify the statistical relevance of the discovered propositional rules [303]. The classic association rules mining methods, such as Apriori, adopt binary attributes that flag the existence or not of each item (sold goods) in the shopping basket (sale transaction). The coexistence pattern of items considering diverse transactions allows the analyst to understand the customers' purchase behavior.

However, industrial processes are more complex, commonly presenting some non-linear relationships between the variables. The industrial systems' databases include continuous variables (real attributes) and categorical information that must be part of the discovered association rules.

To handle continuous variables, for instance, it is necessary to adopt some discretization method seeking to transform the continuous attributes into discrete ones [304]. The most straightforward approach is segmenting continuous attributes by adopting fixed intervals. Each one of the ranges of each variable will appear as a new binary attribute (or a category), thus indicating that a given variable is within a specific range, assigning that category to the corresponding transaction (row, sample).

Another relevant question is: when performing shopping basket analysis, it is possible to include any item as an antecedent or consequent of the extracted rules. On the other hand, in the case of industrial process analysis, some variables are seen exclusively as antecedents while others are necessarily consequent of the rules. Operational parameters such as machine settings and actual process conditions are seen as antecedents, while quality indexes and operational performance indicators are seen exclusively as consequences.

Some libraries have already handled those issues. For instance, the R package called *arules* [305].Arules implementation allows tuning filters involving items names, items' position in the rule (antecedent or consequent), and including rules' probabilistic metrics.

The bibliographic exploration of the association rule mining theme shows works applied to industrial databases. Table B.1 summarizes some of them.

Bib	Area	Algorithm	Summary
[306]	Metallurgy - Blast Furnace	Apriori	It seeks to analyze the relationship between operational parameters and the corresponding production indicators. It comments on the discretization of continuous variables at intervals defined by process specialists. The generated labels, corresponding to each variable interval, are formed by concatenating the variable id and the range (interval) id.
[307]	Metallurgy - Blast Furnace	Apriori	It seeks to determine the qualitative relationship between the various measurement points along the refractory surface of the Blast Furnace. Each measurement point appears as an "item". The flag 1 for certain item indicates a positive variation in the instant of temperature measurement.
[308]	Maintenance Engineering	Apriori	ERP - SAP ECC Maintenance module data. Extract rules from maintenance services tickets huge database. It comments some ways of visualizing association rules.
[309]	Semiconductor Assembly	Apriori	Associating process conditions with quality issues to perform predictive analysis.
[310]	Sales Department	Multidimensional Association Rules	Sales analysis. It uses multidimensional data from ERP system. It determines fixed points for continuous variables discretization.
[311]	Oil and gas	Apriori [301], Eclat [312] , FP-Growth [313]	Algorithms comparison. It uses data from manufacturing processes. It formats the data to meet the specific needs of each algorithm by grouping and labeling sequence of events.

Table B.1: Examples of association rules mining from industrial system databases.

B.1.1 Experiment 4 - Mining association rules from an industrial process data base

The present experiment used the Apriori algorithm to investigate the association rules extraction from an industrial process database. The tests adopted an Apriori implementation named arules [305, 309] from the R platform. The industrial process database belongs to a steel plant and involves information from cylindrical bars continuous casting and seamless tubes rolling mill process.

The experiment's interest data are some process variables sampled from a continuous casting machine and the corresponding quality results assessed after the rolling mill execution. Each transaction (record, row) on the dataset contains the process variables followed by the corresponding quality result. Here, the quality result works as flags, indicating the occurrence of particular defects.

The investigation sought to mine rules treating them as hypotheses about the process realms. The experiment has generated hypotheses to explain how continuous casting conditions would influence the occurrence of the defect called husky (shelly) raw material fold in the rolled pipes.

In order to restrict the analysis, the test filter a specific period, steel grade (LC46), and bars gauge (310 mm). Furthermore, to reduce the number of discovered rules, arules's filtering resource was tuned to return only rules that present DMC-flag as the consequence of the rules. Table B.2 shows the apriori algorithm return. It is possible to see the discovered rules.

id	lhs	rhs	support	confidence	coverage	lift	count
[1]	VAZAO_AGUA_MIN=[2.16e+03,2.22e+03) =>	TEVE_DMC_RUIDO=1	0.06818182	1	0.06818182	4.3	66
[2]	VAZAO_AGUA_MAX=[2.21e+03,2.25e+03) =>	TEVE_DMC_RUIDO=1	0.07128099	1	0.07128099	4.3	69
[3]	NIVEL_ACO_MAX=[98,125] =>	TEVE_DMC_RUIDO=1	0.05268595	1	0.05268595	4.3	51
[4]	LCI_POS_NO_SEQUENCIAL=[4,10] =>	TEVE_DMC_RUIDO=1	0.06095041	1	0.06095041	4.3	59
[5]	PESO_BARRA=[6.73e+03,7.65e+03] =>	TEVE_DMC_RUIDO=1	0.08884298	1	0.08884298	4.3	86

Table B.2: Main rules discovered by arules, the Apriori algorithm implementation in R platform.

The arules implementation returned a large number of rules. In order to highlight the most significant ones, those who presented the higher antecedent-consequent influence factor, the algorithm sorted the list according to the metric named *lift*. Then, these discovered rules were presented to the process specialists, who pointed out three of them: [1], [2], and [3].

Rules [1] and [2] has prioritized by the specialists. Rule [3] has became an instigating hypothesis requiring more detailed analysis. Rule [4] gained minor attention, but not discharged. The process variable LCI_POS_NO_SEQUENCIAL (heat's sequential position) has not prior connection with DMC defect. Rule [5] was not prioritized because, according to the experts, there are no previously known phenomenological reasons for the antecedent variable PESO_BARRA ("WEIGHT_OF_THE_BAR") to be associated with the DMC defect.

It is worth mentioning that hypotheses [1] and [2] refer to the same tag (measured variable's signal coming from the plant). The process variable is the cooling water flow. The antecedent variable exposed in [1] consists of the minimum water flow value assumed in the period. In contrast, the antecedent variable in [2] reflects maximum water flow value. According to the process experts, large fluctuations or inadequate water flow rates can, in certain situations, cause heterogeneous cooling of the material and irregular solidification of the steel bar, thus leading to DMC formation. The problem can be intensified during the subsequent rolling and heat treatment process.

The rules discovered by the Apriori algorithm proved to be consistent hypotheses for process issue investigations. For instance, in the present study, more detailed historical analyses of the antecedent variable in the rules [1] and [2] have revealed intermittent failures in the water supply in the period in question. In order to mitigate the occurrence of this same problem on new occasions, some warnings (alarms) were implemented in the continuous casting machine's SCADA system.

B.1.2 Expert Systems

The topic of explainability has been a subject of interest even in the domain of expert systems within the classical AI paradigm. The literature reveals several works dating back to the 1970s, as is evident from the bibliographic survey presented in Moore's (1988) study [259]. Davis' and his colleagues (1977) work [258] emphasizes the significance of explainability for the specialist community and presents a rules-based system with its own grammar. Their system is equipped with a textual front end (a kind of chat) that accepts various explanation requests. The researchers successfully developed a system that can generate textual explanations in high-level language for queries such as "how did you decide that sample A might be classified as Class B? From the paper:*"...how did you decide that organism-1 might be an e.coli?..."* [258]. The generated response is based on the propositional rules chain that were employed in the system (model) inference process, which is associated with the respective relevance metrics. This exemplifies a typical instance of explainability, where the user gains an understanding of the reasoning behind the model's behavior.

Systems that rely solely on crisp rules are an older paradigm and have been followed by systems that incorporate fuzzy rules and other advanced methodologies [314]. In industrial environments, expert systems have become increasingly prevalent on the factory floor. They automatically compute sets of propositional rules and generate outputs based on the current values obtained from the process variables. In addition, expert systems frequently assume the responsibility of prescribing the sequence, timing, and execution parameters of activities, especially in the context of process planning [315].

Different industrial applications for rule-based expert systems can be found in reviews written over time: electrical power systems [316], nuclear [317], robotics, modeling and simulation, energy planning and control, production [318], transportation [314], in addition to other areas that do not directly involve the factory floor [319].

The following topic describes an experiment that aims to implement fault diagnosis explanation via a rule-based system.

B.1.3 Experiment 5 - Fault diagnosis explanation via a rule-based system

This study examines applying a rule-based expert system to diagnose equipment faults. The experiment involved levying propositional rules for equipment operation on a seamless steel tube rolling line. These rules were collected with the participation of process specialists, such as operators, responsible engineers, and electrical and mechanical maintainers.

Subsequently, a graphic panel was created to provide a visual representation of the actual equipment and sub-parts actual situation based on the recorded rules. Resources from an already installed Plant Information Management System (PIMS) were used to accomplish that purpose. The specific PIMS used was the OSI Soft PI platform [320].

This proof-of-concept's primary objective was to create a quite simple graphic display that identifies the equipment and its sub-components that may be experiencing a failure. The equipment and locations in question were organized into a seven-level hierarchical tree, with the rules focusing mainly on the deeper levels of the tree (sensors with smaller granularity).

Initially, the hierarchical structure of the equipment, or installation locations, was imported from the Enterprise Resource Planning (ERP) system SAP R3 to the PIMS system. Once the hierarchical structure was included, the rules reported by the experts were registered and the output status defined by each rule was labeled using a color code.

The graphic representation follows the hierarchical structure of the registered equipment, allowing users to drill down to the most elementary hierarchical level and provide a detailed explanation of the alert reasons level by level. Such a graphic interface aimed to represent the mental model of the equipment hierarchy.

Figure B.1 - a provides an example of the developed panel, indicating the state of the Cone Type Piercer equipment (CTP Roller) at a given moment.

Figure B.1 - b exemplifies some registered rules involving tags or process variables. The subdivisions on the heat map reflect the hierarchical structure, indicating the constituent parts of the visualized equipment/device. The colors indicate the different situations in which the equipment can be found.

The various variables or tags used in the rules are read from the field every second. The rules are then processed sequentially, and the graphical panel is automatically updated. The graphical panel shows the diagnosis, obtained from the set of rules, and its explanation showing "why" a particular equipment is "red" or "orange".

The logical rules' persistence and automatic calculation ensure that the knowledge related to the process in question is maintained, shared, and standardized. The graphical representation, a heat map chart, facilitates interpretation and displays which rules are being met and which are not.

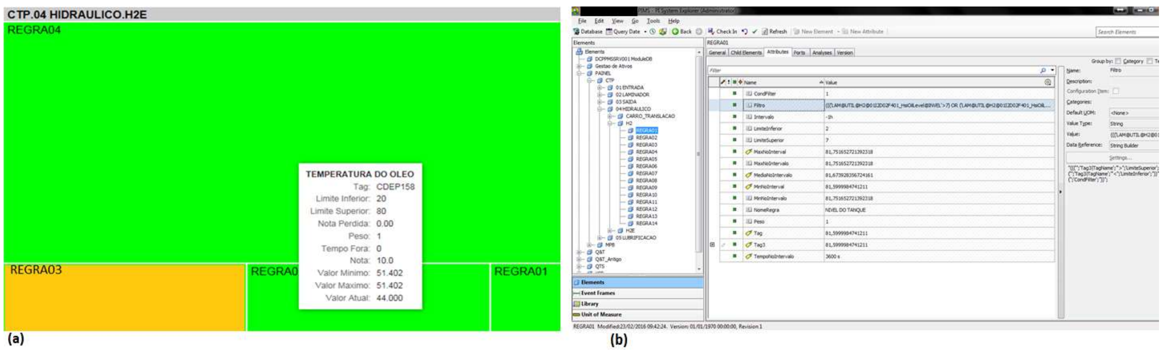


Figure B.1: Expert system based on crisp rules. Diagnosis and its explanation.

B.2 Post-Hoc Explainability

B.2.1 Shapley Values - Features Importance

Those techniques seek to describe opaque model responses by classifying or measuring the influence (relevance or importance) that each attribute (feature, characteristic) has in the model's final output [4]. The main idea is to quantify the participation of individual datasets' attributes in the model's final result.

In order to quantify the participation of individual attributes the most popular approaches adopt concepts described in the Game Theory [321], mainly the method proposed by Lloyd Shapley [87].

From the Shapley Values perspective, each feature appears as a "participant" in a Game Theory coalition game [322]. Then, a numerical value called *Shapley Value* is assigned to each characteristic, demonstrating its individual contribution to the model's final result.

Currently, one of the most widespread computational tools regarding explainability through Shapley Values is the so-called Shap, from *SHapley Additive exPlanations* [286] with implementations available in R and Python languages [155, 323–325].

Nowadays, available implementations have graphical resources that display the attributes' Shapley values, both for each particular sample and generalizing for the entire dataset.

Shap tool performs explanation by following the steps that led to the model's decision. It is possible to demonstrate, using charts, how the attributes' values, one by one, conducted the model to this or that result. The values of each attribute, for example, guide the model to classify the samples in this or that class. The subsection Computational Experiments will address that issue in more detail.

the subsequent section takes care about the practical demonstration of the use of Shapley Values as a means of explaining the response of opaque models facing two different situations.

The explanation of models' output through Shapley Values was experimented with SHAP implementation [286, 324]. Two tests were carried out. The first one involves a classification problem, while the second addresses a regression situation.

Opaque models explanation through Shapley Values - Demonstration 1: Classification Task

The current demonstration adopted a Random Forest classifier implemented by a Sklearn package [326]. The purpose was to train that opaque classifier with an actual dataset collected from a steelmaking continuous casting process. Given some process variables, the classifier learned how to predict the defect presence flag (TEVE_DMC_RUIDO). DMC is a type of steel pieces defect.

The training process generated a model AUC = 0.84. The method itself and its tuning parameters are not relevant here. The investigation is concerned with the models' responses explanation method, not with the model training itself.

The charts in Figure B.2 show the classification decision process in two opposite situations. It uses pictorial resources to show the features importance on the models' output. It tries do explain the models' output via features forces dynamics.

The upper plot shows a classification prediction corresponding to a sample that indicates a DMC defect. The lower plot shows a defect-free prediction situation. That kind of plot intends to demonstrate how the features' values take the model's output from its base status and conduct it until its final response. The base value consists of the average model output value defined over the training dataset. Roughly speaking, the plot represents a "forces chart" showing the behavior of the forces' dynamics and how it generated the model's output.

Shap implementation brings a varied plots arsenal that helps in the interpretation of attributes' Shapley Values behavior [323, 324]. For instance, Figure B.3 shows the force plot for an entire dataset. To do so, it is a matter of generating several individual explanations, like the two shown in Figure , rotate them 90 degrees, and line them horizontally from left to right.

Figure B.4 shows the beeswarm and waterfall plots that can help explain the model's classification. Figure B.5 shows two different cases of the dependence plot. In the first one it is possible to see the simple dependence plot, while the second shows a Multi featured example.

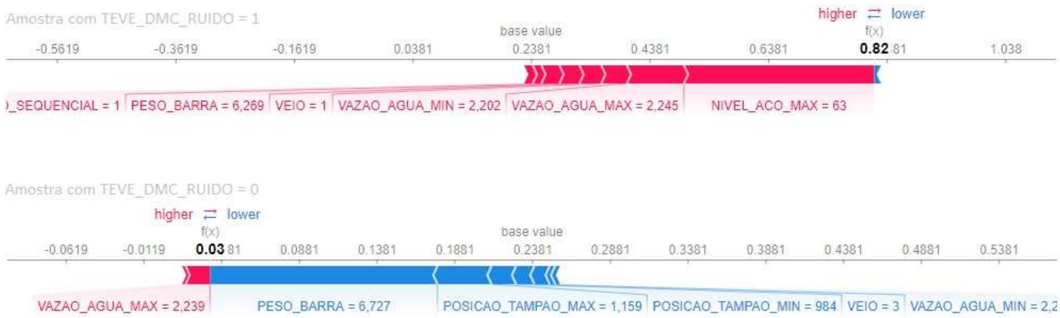


Figure B.2: Plots showing the interaction between the attributes' values forces and how they displaced the model's output from its resting state to its final value.

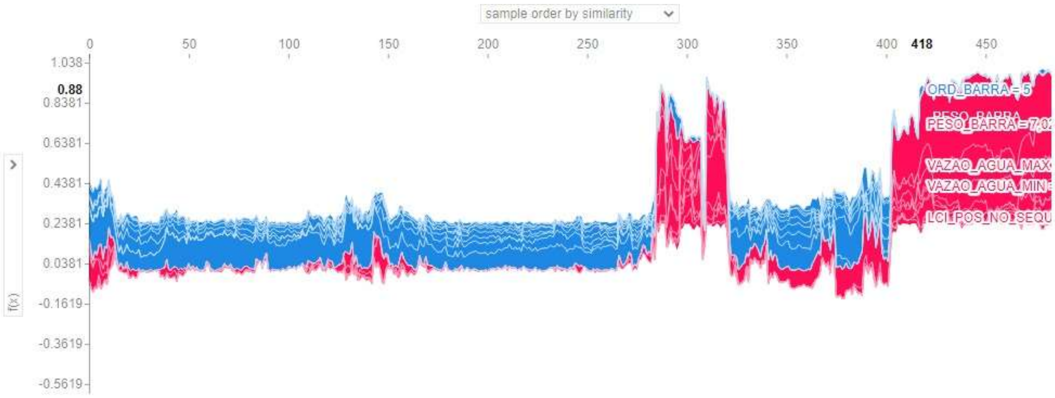
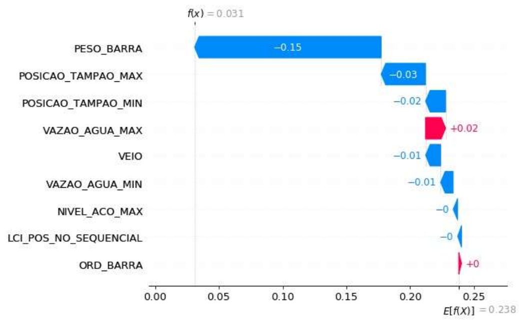


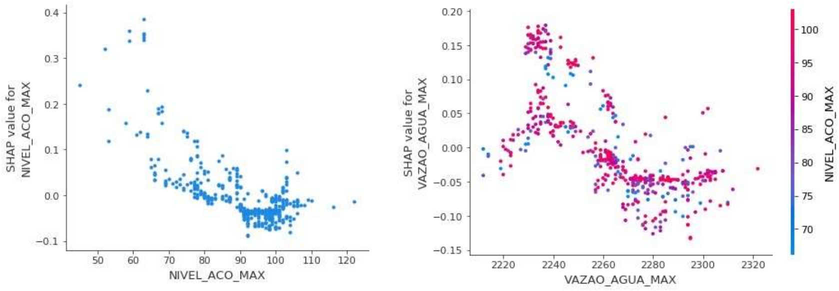
Figure B.3: General model behavior. Entire dataset Force Plot.

(a) Summary - beeswarm plot



(b) Waterfall plot

Figure B.4: Shap Explanation Plots - beeswarm summary view and waterfall plots examples.



(a) Dependence plot **(b)** Dependence plot - Multi featured

Figure B.5: Shap Explanation Plots - dependence plots examples

Opaque models explanation through Shapley Values - Demonstration 2: Regression Task

The present demonstration involves a regression task. The problem in question is to predict some mechanical properties of seamless steel tubes after their heat treatment process. The heat treatment is a metallurgical process that seeks to impart determined mechanical properties to metal alloys to meet the specific needs of their application. In the case of steel tubes, these properties mainly depend directly on the tube geometry, the steel alloy chemical composition, and the heat treatment process conditions to which the material was submitted during its manufacturing process [327, 328].

The adopted regressor model is a Gradient Boosting Regressor from the Cat Boost Regressor implementation available in the Sklearn library.

The dataset comprises several variables collected by the heat treatment area's Material Tracking System (level 2 system) during the entire process execution. The contained information embraces the tubes dimensional data, heating and cooling process data of each tube individually, the metal alloy's actual chemical composition, and the respective mechanical properties values achieved after the heat treatment process. Each row in the database represents an individual produced tube with its process execution conditions and the corresponding mechanical properties results.

The model input data consists of the dimensional variables and heat treatment process data. The output to be predicted by the regressor consists of the mechanical property called Yield Strength (The yield strength property seeks to measure the material's resistance to its plastic deformation [327]).

After training the Gradient Boosting Regressor model, the Explainner object of the Shap implementation was applied in order to test the generated explanation. The regressor explanation adopts the Shapley values to expose the answer explanation. Figure B.6 shows a Force Graph for a specific sample. The output of the regressor is explained by the resultant of the forces shown in the graph. Figure B.7 shows two waterfall plots explaining the output

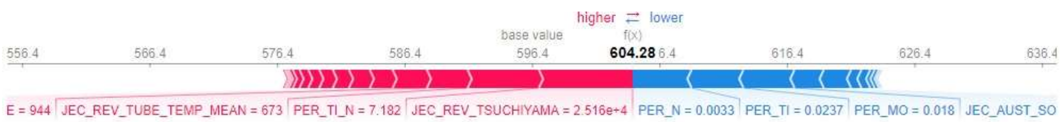


Figure B.6: Force plot explaining the model's output referring to one single sample.

promoted by two different samples. Note that the individual attributes' values generate different Shapley values, producing different force magnitudes and leading the regressor output to different outputs (final result).

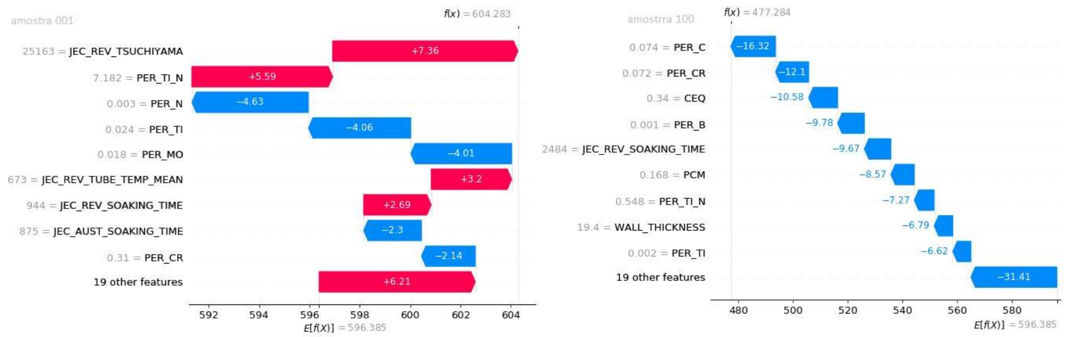


Figure B.7: Waterfall explaining the outputs from sample number 001 and number 100.

Another way to explain the model output is through Decision plots. Figure B.8 shows an example of that kind of plot. Note that it is possible to check the dynamics oscillation from the average value (resting point) until the output given by the regressor.

Notice the different driving promoted by attributes' distinct values. It is possible to see how each attribute influenced the model's output individually. This fact can be helpful, even in identifying biases caused by unbalanced or unrepresentative data sets.

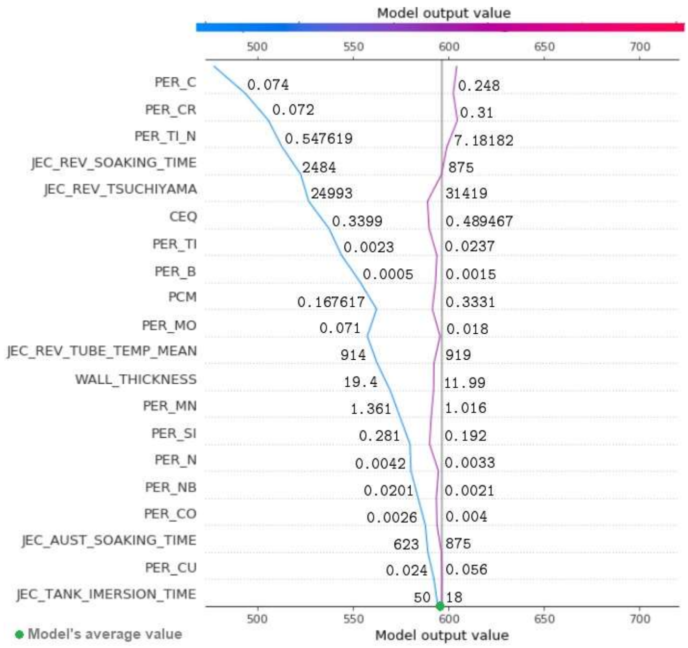
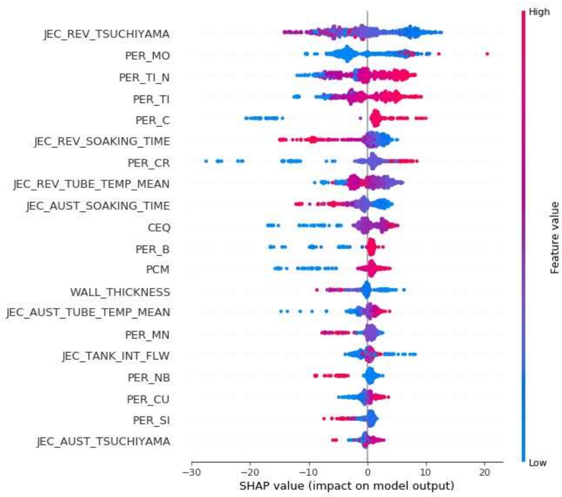
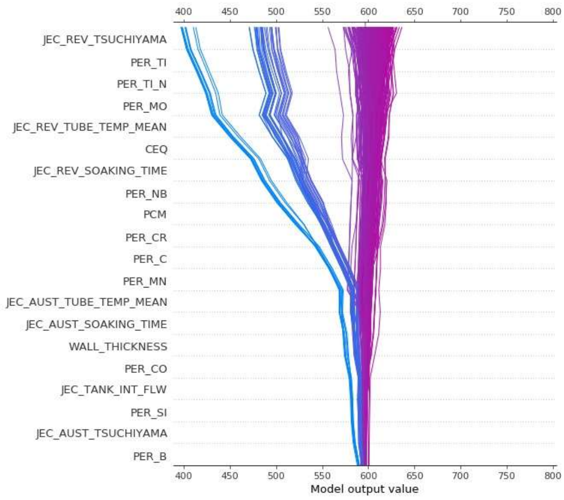


Figure B.8: DECISION PLOTS EXPLAINING THE MODEL'S OUTPUT VALUES DYNAMICS FOR TWO DIFFERENT SAMPLES.

It is also possible to use other plot resources to understand how the variables' different values influence the model's response. Figures B.9 and B.10 show examples of those functionality. Note that it is possible to perform individual and combined analyses. The plot permits investigating which attributes and values influence the model's output most.



(a) Summary - beeswarm plot



(b) Decision Summary

Figure B.9: Additional explainner plots

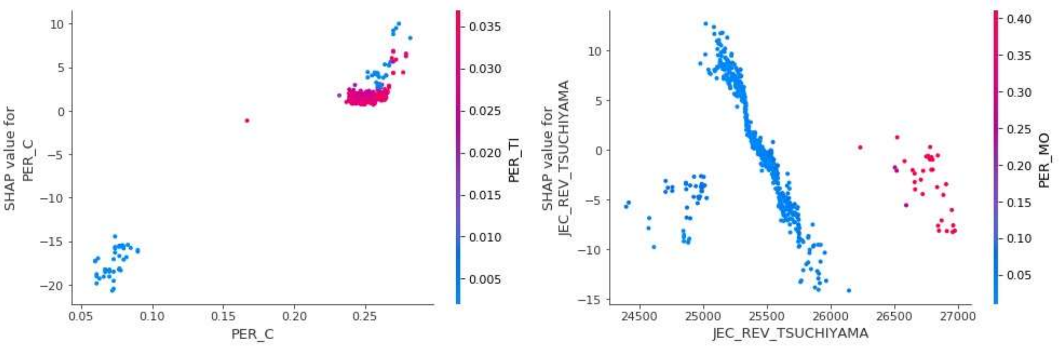


Figure B.10: Scatter plots showing the relationship between two variables and the magnitude of influence on the model response.

The next section describes some works that adopt the use of Shapley Values as a way to explain models' outputs.

Applications of Shapley values in model interpretation

It is possible to find several works describing employment of Shapley Values as a way to understand models' output. Table B.3 summarizes some of that.

Bib	Area	Model	Implement	Summary
[329]	Automotive industry	Catboost	R e Python (Shap)	Modeling prices of vehicles and their components leveraging a combination of machine learning and game theory concepts.
[330]	Automotive industry	XGBoost	Python (Shap)	Prediction of failures in nitrogen oxide sensors of urban bus diesel engines.
[331]	Industrial Instrumentation	MLP	Phyton	Prediction of events that lead to the occurrence of operational stoppages (downtime) of an industrial Nitrogen compressor.
[332]	Relationship Management	LightGBM	Phyton (Shap)	CRM and B2B. Determining pricing strategies in after-sales customer relationships.
[333]	Siderurgy, Melpshop, FEA	ANN	Phyton (Shap)	Electricity consumption forecast in an FEA (Electric Arc Furnace)
[334]	Siderurgy, Rolling Mill	XGBoost	Phyton (Shap)	Seeks to detect anomaly patterns in the steel ring rolling process, recognizing situations that lead to the risk of deformation of the parts during the manufacturing process.
[335]	Asset Management	SOM	Phyton (Shap)	Customer profiling and segmentation.
[336]	Mining	Random Forests	Shap Regression	Leaching reagent consumption prediction in a gold ore processing circuit
[337]	Administration	LSTM	Python (Shap)	Predictive monitoring of business processes. Detect specific instances of process execution to flag in advance those that require special attention (those that can generate unfavorable conditions, excessive execution time, promote high cost, quality risk). It includes KPIs forecasting case study.
[338]	Oil and Gas	XGBoost	Python (Shap)	Modeling the deposit formation process in shale oil exploration wells.
[339]	People Analytics	LightGBM	Python (Shap)	To trace out and understand the reason for employee attrition.

[340]	Maintenance Engineering	DNN	Python (Shap)	Fault detection and diagnosis in rotating machinery.
[341]	Maintenance Engineering	Random Forest	Python (Shap)	Spare parts lifetime estimation applied to an automotive multimedia assembly company.
[342]	Business Process Management	DNN	R	Implementing a predictive process analytics by integrating the top-floor and shop-floor data obtained from several information systems, a deep learning model to predict the process outcomes and Shapley values to explain it.
[343]	Quality Management	DNN ensembles	Python (Shap)	Mapping the relationship between a set of production parameters and the manufacturing process quality.
[344]	Maintenance Engineering	LSTM	Python (Shap)	Gas turbine failure prediction.
[345]	Utilities	XGBoost ensemble	Python (Shap)	Prediction of chlorophyll-a concentration in a drinking water supply system.
[346]	Maintenance	Auto-encoder	Discrete Time For SHAP	Detect and explain anomalies in industrial multivariate time-series.

Table B.3: Examples of industrial applications involving Shapley values-based XAI

More examples of Shapley values industrial-related applications can be found on the surveys works [288, 347].

B.2.2 Explanation via Knowledge Extraction

The main idea behind AI Models' Knowledge Extraction consists of attempts to understand the trained models' behavior through symbolic rules. That use to be a recurrent subject, mainly regarding to *knowledge extraction from neural networks*.

Actually the relationship between symbolic rules and connectionist AI models has existed since the dawn of the field, as can be seen in the historic landmark established by the work of McCulloch and Pitts in 1943 entitled *A logical calculus of the ideas immanent in nervous activity*

"...Because of the "all-or-none" character of nervous activity, neural events and the relations among them can be treated by means of propositional logic. It is found that the behavior of every net can be described in these terms, with the addition of more complicated logical means for nets containing circles; and that for any logical expression satisfying certain conditions, one can find a net behaving in the fashion it describes..." [348].

The authors aimed to demonstrate that, how nerves can present the both "on" or "off" state, it is possible to use logic to understand how they work together. From that principle, is it possible to describe how groups of nerves work collectively using logical symbolic expressions, and it is also possible to find a group of nerves that behave in the way a logical expression describes.

The subsequent researches, however, pointed the explanation matter in symbolic pure approaches, mainly in specialist system such as [257, 258].So, in the mid-80's, the other historical milestone, *Learning Internal Representations by Error Propagation* of Rumelhart, Hilton and Williams, demonstrated all the potential of the connectionist approach, pushing a wide application of the Artificial Neural Networks.

"...Thus, if we have the right connections from the input units to a large enough set of hidden units, we can always find a representation that will perform any mapping from input to output through these hidden units..." [349].

The work showed the powerful effect of using hidden layers and continuous weight adjustments through the generalized delta rule. However, due to the very nature of the problem addressed, the issue of symbolic representation and interpretability of the input-output mapping was not including the investigation.

From that moment on, the widespread use of neural networks led researchers to interest in opening the black-box of the neural networks. The idea relied in "*...provide an insight into the neural network's inner "reasoning...*" [350].

One can found researches during all remaining 80's, [260, 261, 350, 351], passing by the 90's, with the widely commented works [263–266]. Since then, the topic has produced a diversified endeavors. Including various shallow models [267, 269, 352], deep structures [275, 279, 283] and also graph networks [353].

The large number of works produced stimulated the development of reviews (surveys) throughout this historical period until recent years. Here, one can see a list of surveys in chronological order [4, 16, 267–280, 291, 353].

B.2.3 Description of the principals knowledge extraction methods

The present section exposes the principal concepts of Knowledge Extraction from opaque models. The main approaches are organized based on the translucency property from the ADT [291] taxonomy.

Translucency concerns the relationship between the extracted knowledge and the model's internal structure. How much the extracted knowledge is linked to the model's internal parts. In other words, it takes care of the level of detail the knowledge extraction method sees the model internally.

Such an attribute categorizes the knowledge extraction methods into one of three classes. (1) *Decompositional*: takes into account the internal model's structure during the human-friendly knowledge representation forming process. For instance, a decompositional method looks at layers and neurons individually. (2) *Pedagogical*: it means model-agnostic. See the trained models as a "black box."Feed the model input with the examples, obtain the corresponding outputs, and build the symbolic knowledge through some heuristics. (3) *Ecletic*: corresponding to methods that combine elements of the previous two.

Pedagogical methods

Pedagogical approaches see the models as "black boxes". Models' internal parts do not matter. There is an agent playing the role of an "oracle", asking successive questions to the model seeking to extract information about its behavior. It is a attempt of understand the models' behavior through their answers. An iconic example is the TREPAN method (Trees Paroting Artificial Neural Networks) proposed by Mark Craven [268] which builds decision trees equivalent to trained neural networks's implicit knowledge.

TREPAN produces a decision tree that approximates the function learned by the trained neural network. To do so, it uses the trained network itself and the data set.

The generated decision tree promotes clearer understanding of the network's decision-making process. The generated symbolic rules in each tree node can be IF-THEN or M-of-N types. The disadvantage of this method is that depending on the database dimensionality and the learned function, the generated tree may contain numerous rules, making the model's behavior difficult to understand.

There are implementations and available source codes for general use of the TREPAN algorithm in the Java [354], C [355] Python [356] and R programming language [357].

Works inspired by TREPAN have been developed over time [358], including a hybrid version for deep models [359], and reaching recent years with the TREPAN RELOADED [360,361], an extended version that optimizes the decision tree creation process by determining which features are more understandable for a particular user. To do so, the proposed method adopts ontologies (set of formulae) that model specific domain's knowledge.

Decompositional methods

Decompositional methods see the models' internal structure. In the case of feedforward neural networks, such methods use synaptic weights and neuronal activation values to generate symbolic rules.

Usually, such methods can present the following steps: (1) Model training: Model training itself. It may include previous knowledge insertion procedures [263]. (2) Size adjustment. It refers to pruning strategies, summarization, discretization, and clustering of neuron weights and outputs. That step addresses reducing the network, fostering the generation of more straightforward rules. (3) Knowledge Extraction. It usually involves symbolic rules formulation or decision tree induction. (4) Merging strategies aspiring rules simplification. (5) Exhibition. It deals with the knowledge displaying strategies. For instance, how to show the composed symbolic rules or induced decision trees.

A decompositional method commonly cited in the literature is the Neurorule [265]. This method extracts "IF-Then" classification rules from trained Multilayer Perceptron (MLP) neural networks. NeuroRule and other methods derived from it have a workflow similar to that mentioned above. Its first step involves clustering and discretizing the hidden layer activation signals generated by the correctly classified samples. Then, it maps the hidden layer activation values to the network's outputs through IF-THEN rules. After that, it generates rules to explain the hidden layer activation clusters as a function of the original attributes that stimulated the network entry. Finally, it combines the rules generated in the previous steps. This way, the formulated set of rules can explain the network's final output values as a function of the attributes' values presented at the network input.

Other relevant decompositional rule extraction algorithm is denominated *Continuous/describe Rules Extractor via decision tree Induction* (CRED) [362]. Roughly, CRED decomposes the neural network building decision trees based on the hidden-output and input-hidden activation patterns. The idea is to induce decision trees considering the hidden neuron activation values as attributes (characteristics) and the output neuron values as the response variable. This process converts each output layer neuron into a decision tree. The nodes of this tree represent logical tests using hidden layer activation patterns (vectors), and the leaves represent the corresponding class in the output layer.

Then the same idea is applied to the previous layer. The input activation patterns work as characteristics (attributes), and the hidden neuron activation pattern (vector) plays as the response variable. Some determined aggregation criteria are adopted to simplify the generated rules. Figure B.11 shows that method.

Figure B.11-a represents a hidden layer activation patterns vector. Figure B.11-b shows a spatial partitioning of the hidden layer's activation patterns. The plot also exhibits the classification label corresponding to the output layer's response. Figure B.11-c shows the decision tree corresponding to the output space partition. CRED performs its decision tree induction through the C4.5 algorithm [363].

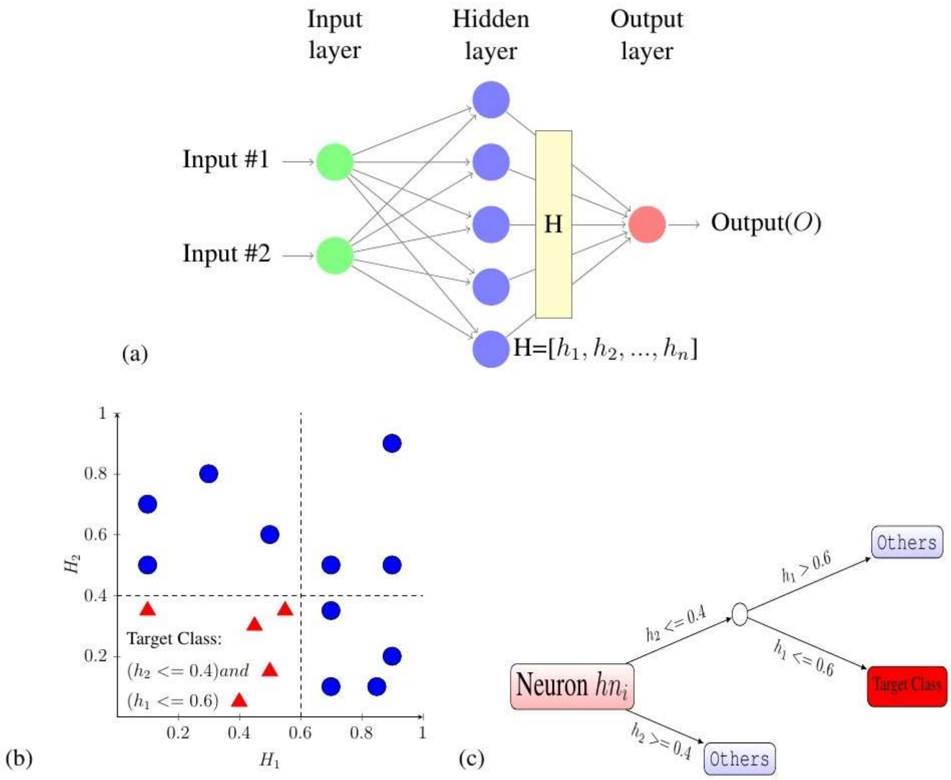


Figure B.11: (a)-Activation pattern vector H; (b)-H space partitioning and output's labeling; (c)-Induced decision tree

The CRED initial version works exclusively with one-hidden-layer networks, but several researchers have aimed to adjust CRED's approach to other types of neural networks. There is a version for Deep MLPs (DeepRed) [364]. Besides adapting CRED to work with any quantity of layers, DeepRed also brought improvements in explaining network knowledge. While CRED adopts decision trees, DeepRed uses Decision Directed Acyclic Graphs (DDAG) that have a more remarkable ability to synthesize the extracted rules [365, 366]. There is a version also for Convolutional Neural Networks [279].

Andrejs Bondarenko has developed different methods for extracting rules from feed-forward neural networks [367, 368]. He presented a method for extracting IF-THEN rules decision trees through linear Piecewise approximation. Another approach presented in his work was the elliptic rules extraction from RBF Neural Networks. The researcher also contributed with implementations of the proposed methods in the languages Lua [369] and Python [370].

Eclectic Methods

Eclectic, or Hybrid, Methods combine pedagogical and decompositional elements. An example of this category is the Hierarchical and Eclectic Rule Extraction via Tree Induction and Combination (HERETIC) algorithm [371]. HERETIC supports different network architectures, allows any number of layers, and accepts partial connections. HERETIC uses decision tree algorithms to generate, for each neuron, a set of rules that approximates their learned function. The method converts the generated decision trees into symbolic rules and removes the redundancies to form the most succinct set of rules as the final output.

Industrial Application of Knowledge Extraction Methods

The bibliography reports industrial applications of knowledge extraction methods. Table B.4 summarizes some of them.

Bib	Area	Method	Type	Summary
[372]	Oil and Gas	TREPAN	P	Rule extraction for a better understanding of the carbon dioxide corrosion process.
[373]	Power System	REFANN	P	The application involves transformer diagnosis and fault detection via dissolved gas analysis. The method trains an ANN using transformer oil dissolved gases information and then uses TREPAN to extract the implicit knowledge.
[352, 374]	Metallurgy - Rolling Mill	FCANN	P	Understand the relationship between process parameters, different conditions of rolling load, and the final thickness of the material.
[375]	Siderurgy - Melt Shop	Own	P	Extract rules from a tank vacuum degasser SCADA system to support operators's decision making. Rules are induced from an ELM fed with continuous and discrete variables.
[376]	Environmental/Utilities	TREPAN	P	Extract rules from a phytoplankton biomass predictor using several physical and chemical indicators .
[377]	Sintering	Own	D	Rules extraction about the sintering process to aid operators set the process parameters in order to obtain a product (sinter) within the specified quality. An MPL neural network is trained and then the symbolic rules are extracted from it.
[378]	Blast Furnace	pTsm-SVM	D	Integration of a priori knowledge to an SVM
[379]	Environmental	MultiRNNExplorer	D	Environment pollution forecasting via RNN. Implements a system dedicated to visually and exploring the model's behavior for both global and individual levels of the RNN network nodes.

Table B.4: Some examples of industrial applications involving knowledge extraction methods.

Experiment 3 - Opaque Model Surrogate

The present experiment investigated the symbolic rules extraction from a trained black-box model employing TREPAN [268] method implementation called TreeSurrogate [357] for the R platform.

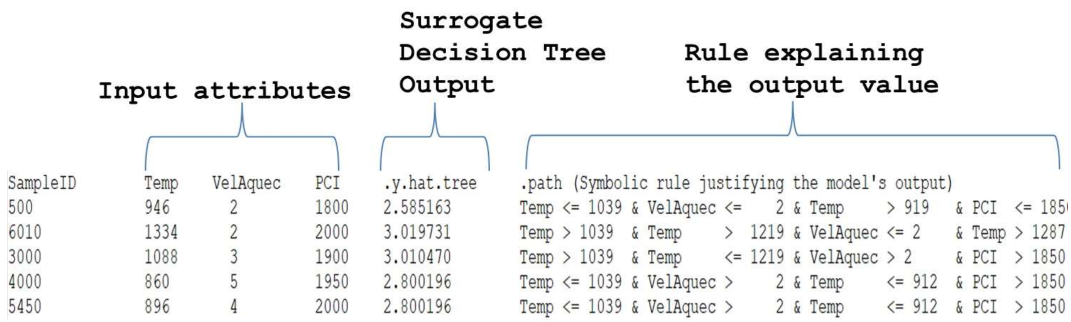
The learning problem involved tuning the Air-Gas ratio's setpoint of steel ingots' reheating furnaces. Three other variables determine the Air-Gas ratio's setpoint: Furnace front temperature (ºC); current heating speed (º/min); and the gas' lower calorific potential (kcal/Nm3). These three input variables and the Air-Gas ratio setpoint (variable to be predicted) are continuous.

The employed AI model consisted of a Multilayer Perceptron (MLP) neural network from the R platform's caret package. The dataset was sampled from the furnaces' SCADA system and counted with 6645 observations (rows in the dataset) of the four variables.

The network's training achieved an RMSE equal to 0.061. After the training assessment, the next step consisted of instantiating a TreeSurrogate object from the trained neural network. The TreeSurrogate object seeks to extract the neural network acquired knowledge and explain it graphically through a decision tree or textually via symbolic rules.

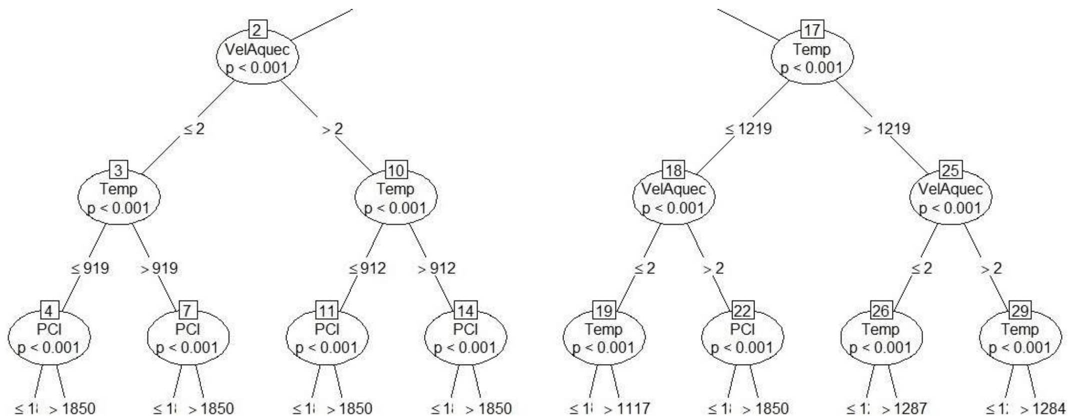
Figure B.12-(a) exhibits symbolic rules explaining (justifying) the surrogate decision tree, and by consequence the network's, outputs for five different samples. It is possible to verify why the neural network gave a particular response value in front each situation.

Figure B.12-(b) shows a piece of the decision tree (DT) induced from the trained neural network's implicit knowledge. The induced decision tree seeks to parrot the MLP trained previously.



SampleID	Temp	VelAquec	PCI	.y.hat.tree	.path (Symbolic rule justifying the model's output)
500	946	2	1800	2.585163	Temp <= 1039 & VelAquec <= 2 & Temp > 919 & PCI <= 185
6010	1334	2	2000	3.019731	Temp > 1039 & Temp > 1219 & VelAquec <= 2 & Temp > 1287
3000	1088	3	1900	3.010470	Temp > 1039 & Temp <= 1219 & VelAquec > 2 & PCI > 1850
4000	860	5	1950	2.800196	Temp <= 1039 & VelAquec > 2 & Temp <= 912 & PCI > 1850
5450	896	4	2000	2.800196	Temp <= 1039 & VelAquec > 2 & Temp <= 912 & PCI > 1850

(a) Symbolic rules justifying the models' outputs regarding five distinct samples



(b) Part of the parroting decision tree (induced from the MLP)

Figure B.12: Example of two features offered by "iml" library of the R platform. TreeSurrogate function implements the TREPAN algorithm.

B.3 Explainable Artificial Intelligence Applied to Time Series

Industrial plants usually dispose of systems devoted to sampling, persisting, and managing signals produced by their process. PIMS systems, for instance, seek to keep historic of time-based variables (tags) gathered from whole the plant. It can include dozens of thousands of tags. PIMS is widely adopted by Chemical, Oil and Gas, Mining, and Metallurgical plants. Supervisory (SCADA) systems are another kind of system that also deals with signals.

Supervisory systems seek to monitoring and act in the process. Usually they take care of one particular process or equipment. Depending on the required sample rate it is also employed high-speed PDA systems that sample data directly from the controllers (PLC) via optical fibers. The leading players of such systems offer real-time data exchange features, allowing, for instance, to transmit signals to math models and receive some information as a response.

The systems usually format the signals adopting time series structures. The time series contents can provide information about the current conditions of the process or equipment that has produced them. Some implicit information in the time series' behavior allows maintainers and operators to detect anomalies in their processes, predict events and take corrective actions.

Industrial processes' dynamics usually produce complex time series. Machine learning problems that deal with such signals fall into the "interpretability vs. accuracy" trade-off. For example, the most performative methods in time series classification tasks are ResNet (Deep model [380]) and HIVE-COTE (Hierarchical Vote Collective of Transformation-Based Ensembles [381]). Both are considerably opaque. Understanding their behavior and extracting their implicit knowledge will become more challenging.

Therefore, the best performance of complex models treating time series leads to rising demand for XAI methods, especially on the industry shop floor. Comprehending why the model has made some decision or presented some response is enjoyable for time series-based tasks such as classification, similarity analysis, clustering, prediction, and knowledge extraction.

B.3.1 Applications of time series models explanation

Although the available literature on interpreting opaque models for time series is relatively scant compared to other data structures, such as tabular data, text, and images, it is possible to find works addressing that issue. Table B.5 summarizes some of recent works.

Bib	Input	Summary
[382]	CNN	Fault diagnosis from a bearing signal. Fault severity assessment. Presents algorithm that can extract an informative structure from the deep-layer model. The architecture includes one additional explainable layer to the usual deep learning model.
[383]	CNN	Adopts class activation maps to highlight the most important time series aspects that contributed to the model output (decision). Offers a means to handle time series features importance.

[384]	CNN	Comments the forecast of a photo-voltaic power plant energy consumption. To highlight the most important features of the input they employed Grad-Class Activation Maps.
[385]	CNN	Brings a wavelet decomposition network. It execute perturbations in both high and low frequency of the the time series' sub-sequences through various network layers. The approach makes it possible to assign a relevance to each sub-sequence per layer by observing the model's output.
[386]	mvTS	Adopts a sliding window to sample sub-sequences of the signal. The sampled windows can be sent to a classifier. The classifier makes a prediction for each individual sampled sub-sequence. The results consists of a probability distribution over the all input sampled windows. The probability distribution can be used to assign a relevance to each sub-sequence. It fits multivariate time series.
[387]	LSTMs and ResNets	Performs XAI methods (LIME, LRP,DeepLIFT,Saliency Maps and SHAP) comparison. Applies XAI methods previously designed for text and images to time series. It indicates that SHAP method works robustly for all models, but DeepLIFT, LRP, and Saliency Maps work better for some specific architectures.
[346]	mvTS/autoencoders	Conceives anomaly detection methods to multivariate time-series. It brings seven techniques that aims to explain the root cause of the anomalies. Methods based mainly on feature attribution, explanation by example, and decision trees. It deals with multivariate time series.
[388]	TS	Adopts RNN to extract the anomalous spikes from the time series. Treats the spikes as events and then builds an RNN classifier with attention mechanisms to insert the frequency and irregularity of the extracted events (spikes). Uses a LIME version for time series to interpret how the spikes contribute to the predictions.
[389]	mvTS	Conceives an Explainable-by-design ensemble method called XEM to perform multivariate time series classification. It combines an explicit "boosting-bagging" approach to manage the "bias-variance" trade-off and an implicit divide-and-conquer heuristic to deal with classifier errors on different parts of the training dataset (adopts a decision tree model to learn different parts of the training data). XEM method tests indicates that it surpass the state-of-the-art multivariate time series classifiers in front of the public UEA datasets.

[390]	mvTS	Proposes distance-based data structures XAI (DDS-XAI). It consists of a modular and flexible framework designed to provide explanations about multivariate time series. The proposed framework has three main steps. 1-Structures identification and Visualization, 2-Cluster analysis, and 3-Providing relevant explanations. Each of these steps has several modules. The method provides the possibility to explore unknown implicit structures within the time series and can deliver pertinent explanations because it does not depend on prior domain knowledge. The process is based only on the time-series data. The work supply links to open-source code in R language for all modules of the DDS-XAI framework.
[391, 392]	TS	The authors present a framework designed to generate, in natural language, explanations about the decisions made by a CNN when detecting abnormalities in time series. Called Time-series Visualization framework (TSViz). The method adopts clustering through Dynamic Time Warping (DTW) similarity metric and information about features influence and some time-series statistics. These set of processed information are sent to a module called Textual Explanation Generator that builds the output texts in natural language. Explanations are generated so that the anomalous points in each channel of the multivariate time series can be highlighted.

Table B.5: Examples of time series-based XAI methods.

Appendix C

Tables

C.1 Main Distances' Formulas grouped by Families

Distance Measure Family	Formula
Lp Minkowski family: Euclidean L_2	$d_{Eul}(A,B) = \sqrt[p]{\sum_{j=1}^{m} \lvert a_j - b_j \rvert^p},\ p = 2$
Lp Minkowski family: City block L_1	$d_{CB}(A,B) = \sum_{j=1}^{m} \lvert a_j - b_j \rvert,\ p = 1$
Lp Minkowski family: Chebyshev L_∞	$d_{Cheb}(A,B) = \max_j \lvert a_j - b_j \rvert,\ p = \infty$
L1 family: Sørensen	$d_{Sr}(A,B) = \sum_{j=1}^{m} \frac{\lvert a_j - b_j \rvert}{a_j + b_j}$
L1 family: Soergel	$d_{Sg}(A,B) = \sum_{j=1}^{m} \frac{\lvert a_j - b_j \rvert}{\sum_{j=1}^{m} \max(a_j,b_j)}$
L1 family: Kulczynski	$d_{Kul}(A,B) = \sum_{j=1}^{m} \frac{\lvert a_j - b_j \rvert}{\min(a_j,b_j)}$
L1 family: Canberra	$d_{Can}(A,B) = \sum_{j=1}^{m} \frac{\lvert a_j - b_j \rvert}{a_j + b_j}$
L1 family: Lorentzian	$d_{Lor}(A,B) = \sum_{j=1}^{m} \ln\left(1 + \lvert a_j - b_j \rvert\right)$
Intersection family: Wave hedges	$d_{WH}(A,B) = \sum_{j=1}^{m} \left(1 - \frac{\min(a_j,b_j)}{\max(a_j,b_j)}\right)$
Intersection family: Motyka	$d_{Mot}(A,B) = \frac{\sum_{j=1}^{m}(a_j + b_j)}{\sum_{j=1}^{m} \max(a_j,b_j)}$
Inner product family: Cosine	$d_{Cos}(A,B) = \frac{1}{\pi}\cos^{-1}\left(\frac{\sum_{j=1}^{m} a_j b_j}{\sqrt{\sum_{j=1}^{m} a_j^2}\sqrt{\sum_{j=1}^{m} b_j^2}}\right)$
Inner product family: Jaccard	$d_{Jac}(A,B) = \frac{\sum_{j=1}^{m}(a_j - b_j)^2}{\sum_{j=1}^{m} a_j^2 + \sum_{j=1}^{m} b_j^2 - \sum_{j=1}^{m} a_j b_j}$
Inner product family: Dice	$d_{Dice}(A,B) = \frac{\sum_{j=1}^{m}(a_j - b_j)^2}{\sum_{j=1}^{m} a_j^2 + \sum_{j=1}^{m} b_j^2}$
Fidelity family: Matsuita	$d_{Mat}(A,B) = \sqrt[2]{\sum_{j=1}^{m}\left(\sqrt{a_j} - \sqrt{b_j}\right)^2}$
Fidelity family: Squared-chord	$d_{Sch}(A,B) = \sum_{j=1}^{m}\left(\sqrt{a_j} - \sqrt{b_j}\right)^2$

Table C.1: Distance metrics families.

Organization and table originaly presented in [185].