U.PORTO

FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Trustable Intelligent Decision Support for Enhancing Industrial Digital Twins

Flávia Georgina da Silva Pires

Supervisor: Paulo Leitão, Ph.D., Polytechnic Institute of Bragança

Co-Supervisor: António Paulo Moreira, Ph.D., University of Porto

Programa Doutoral em Engenharia Electrotécnica e de Computadores

June 6, 2024

Faculdade de Engenharia da Universidade do Porto

Trustable Intelligent Decision Support for Enhancing Industrial Digital Twins

Flávia Georgina da Silva Pires

Programa Doutoral em Engenharia Eletrótecnica e de Computadores

Approved by:

President: Prof. Dr. Ana Cristina Costa Aguiar, Universidade do Porto
Referee: Prof. Dr. Maria Leonilde Rocha Varela, Universidade do Minho
Referee: Prof. Dr. Carlos Fernando da Silva Ramos, Instituto Politécnico do Porto
Referee: Prof. Dr. Luís Paulo Gonçalves dos Reis, Universidade do Porto
Referee: Prof. Dr. Paulo José Cerqueira Gomes da Costa, Universidade do Porto
Advisor: Prof. Dr. Paulo Jorge Pinto Leitão, Instituto Politécnico de Bragança

Assinado por: **Paulo Jorge Pinto Leitão**
Num. de Identificação: 08446321
Data: 2024.06.20 17:23:20+01'00'

June 6, 2024

I would like to dedicate this thesis to my husband, my family and friends.

Abstract

The manufacturing industry has faced a significant challenge in decision-making due to the introduction of Industry 4.0 technologies. The traditional decision-making strategies that rely on the decision-makers analysis and empirical knowledge is time-consuming and leads to inaccurate decisions, particularly for inexperienced decision-makers with limited historical knowledge when faced with vast amounts of data.

Digital Twin technology offers a platform to develop an intelligent, automated, and computerised solution for decision support, providing faster, more accurate, flexible, and intelligent decision support. However, although this technology enables real-time monitoring, data analysis, and what-if simulation, the analysis of all this information still depends on the decision-maker. By integrating recommendation systems, it is possible to have cooperative decision support systems that provide recommendations and enable the decision-maker to have their input in the final decision. However, these systems suffer from problems such as cold-start and data sparsity, making performing recommendations challenging.

Therefore, a Digital Twin architecture for decision support based on an innovative recommendation system approach called *SimQL* is proposed. The proposed architecture has six layers *Physical Layer*, *Communication Layer*, *Data Analysis Layer*, *Simulation Layer*, *Decision Support Layer*, and the *Human Trust Layer*, however this work focus mostly on the last three. The proposed recommendation approach integrates trust and similarity measures, what-if simulation, and an AI-algorithm to minimise the effects of the cold-start and data sparsity problems when supporting the decision-making for new users or items. This also includes different forms of calculating the predicted trust rating to improve the predicting rating calculation accuracy.

The proposed approach was experimentally validated using a case study based on a battery pack assembly line called Integrated Manufacturing & Logistics (IML) at Warwick Manufacturing Group (WMG). Each part of the *SimQL* approach was validated individually through the performance of preliminary experiments. Following this, the proposed recommendation approach was compared with state-of-the-art approaches, from which was possible to conclude that the *SimQL* approach outperforms the approaches with which it was compared. A sensitivity analysis of the *SimQL* approach using a Fuzzy Logic approach was performed, it was possible to extract information about the parameters that most influence the recommendation quality of this model.

Keywords: Digital Twin Architecture. Trust-Based Recommendation Systems. Similarity and Trust Measures. Cold-Start and Data Sparsity Problems.

Resumo

A indústria da manufatura está a enfrentar um desafio significativo na tomada de decisões devido à introdução de tecnologias relacionadas com a Indústria 4.0. As estratégias tradicionais de tomada de decisões, que se baseiam na análise e no conhecimento empírico dos decisores, consomem muito tempo e conduzem a decisões inadequadas, especialmente para decisores inexperientes com conhecimentos históricos limitados, quando confrontados com grandes quantidades de dados.

A tecnologia de *Digital Twin* oferece uma plataforma para desenvolver uma solução inteligente, automatizada e informatizada de apoio à decisão, proporcionando um apoio à decisão mais rápido, preciso, flexível e inteligente. No entanto, apesar de esta tecnologia permitir a monitorização em tempo real, a análise de dados e a simulação de cenários *what-if*, a análise de toda esta informação continua a depender do decisor. Ao integrar sistemas de recomendação, é possível ter sistemas cooperativos de apoio à decisão que fornecem recomendações e permitem que o decisor tenha o seu contributo na decisão final. No entanto, estes sistemas sofrem de problemas como o *cold-start* e a *data sparsity*, o que pode dificultar a formulação de recomendações.

Por esse motivo, é proposta uma arquitetura Digital Twin para apoio à decisão baseada numa abordagem inovadora de sistema de recomendação denominada *SimQL*. A arquitetura proposta tem seis camadas *Camada Física*, *Camada de Comunicação*, *Camada de Análise de Dados*, *Camada de Simulação*, *Camada de Apoio à Decisão* e a *Camada de Confiança Humana*, mas este trabalho centra-se sobretudo nas últimas três camadas. A abordagem de recomendação proposta integra medidas de confiança e de similaridade, simulação *what-if* e um algoritmo basedo em inteligência artificial, de forma a minimizar os efeitos dos problemas de *cold-start* e de *data sparsity* ao apoiar a tomada de decisões para novos decisores ou itens. Este sistema inclui também diferentes formas de cálculo da previsão do feedbak de confiança para melhorar a precisão do cálculo da previsão do feedbak.

A abordagem proposta foi validada experimentalmente utilizando um caso de estudo baseado numa linha de montagem de baterias denominada Integrated Manufacturing & Logistics (IML) que se encontra no Warwick Manufacturing Group (WMG). Cada parte da abordagem *SimQL* foi validada individualmente através da realização de experiências preliminares. De seguida, a abordagem de recomendação proposta foi comparada com abordagens do estado da arte, tendo sido possível concluir que a abordagem *SimQL* supera as abordagens com as quais foi comparada. Foi também realizada uma análise de sensibilidade da abordagem *SimQL* utilizando uma abordagem de Lógica Fuzzy, tendo sido possível extrair informação sobre os parâmetros que mais influenciam

a qualidade de recomendação deste modelo.

Palavras-Chave: Arquitetura Digital Twin. Sistemas de Recomendação Baseados em Confiança. Medidas de Similariadade e de Confiança. Problema de *Cold-Start* e de *Data Sparsity*.

Acknowledgments

I am humbled and grateful to all those who have supported me in completing this PhD work, both directly and indirectly.

First and foremost, I would like to express my genuine gratitude to my supervisor, Paulo Leitão, and co-supervisor, António Paulo Moreira, whose guidance and support have been invaluable throughout this journey. Their expertise, knowledge, encouragement, and patience have been instrumental in shaping this work. I also would like to thank Bilal Ahmad and Professor Robert Harrison for receiving me at the University of Warwick in the United Kingdom and for the vital role that they played in the work developed.

I would like to thank the Faculty of Engineering of the University of Porto for allowing me to develop my research work. Additionally, I would like to thank the Polytechnic Institute of Bragança, where I have carried out my entire academic career so far, particularly to the Research Centre in Digitalisation and Intelligent Robotics (CeDRI), who welcomed me before I began my PhD and who encouraged me to enter this adventure.

I would like to thank for the financial support to the Fundação para a Ciência e Tecnologia within the scope of the PhD Grant SFRH/BD/14243/2019.

From CeDRI to the colleagues who have become friends, I would like to thank ALL of them, the ones who keep working there, to those who have recently entered, and to the ones who have already left on new adventures, especially Barbosa (Primo, the one that helped me and encouraged me to the point of being annoying), Nelson Ricardo, Jonas, Adriano, Margarida (Guida, who already was my close friend way before we worked together), Piardi, Marcelo, Victória, Gustavo, Alexandre, Braun, Rebecca, and so many others.

To my family, to the ones who are still here and to the ones who left during this journey, I would like to thank you for all your patience and understanding. To my parents, Zeza and Carlos; to my brother and his wife, Roberto and Sandra; to my grandmother Adelina, my godfather, godson and aunt, Nuno, Luís and Sónia; to all a special thank you. As the Portuguese saying says, last but not least, to my boyfriend-turned-husband during this journey, Hugo, who endured me all the way, I know that it was not easy. I would also like to thank my husband's family, who are now also my family.

Thank you to all those who have played a part, big or small, in the completion of this PhD.

Flávia Georgina da Silva Pires

This work was supported by FCT - Fundação para a Ciência e Tecnologia within the scope of the PhD Grant SFRH/BD/143243/2019 (period from 01-April-2020 - to 12-May-2024).

Publications

This chapter lists the most important scientific contributions developed during the development of the work for this dissertation. The presented publications relate to international, IEEE-sponsored peer-review conferences and journals indexed either on Scopus or Web of Science.

Journal Publications:

- **F. Pires**, P. Leitão, A.P. Moreira, and B. Ahmad **Reinforcement learning based trustworthy recommendation model for digital twin-driven decision-support in manufacturing systems**. *Computers in Industry*, vol. 148, pp.103884, 2023, doi: 10.1016/j.compind.2023.103884.

- P. Leitão, **F. Pires**, S. Karnouskos and A. W. Colombo. **Quo Vadis Industry 4.0? Position, Trends, and Challenges**. *IEEE Open Journal of the Industrial Electronics Society* vol. 1, pp.298–310, 2020, doi: 10.1109/OJIES.2020.3031660.

Conference Publications:

- **F. Pires**, A.P. Moreira, and P. Leitão. **Cold-Start and Data Sparsity Problems in a Digital Twin based Recommendation System**. In *2024 29th IEEE IES International Conference on Emerging Technologies and Factory Automation (ETFA)*.2024, pp.1-8, (Accepted to be presented at the Conference).

- **F. Pires**, V. Melo, J. Queiroz, A.P. Moreira, F. de la Prieta, E. Estevez and P. Leitão. **Positioning Cyber-Physical Systems and Digital Twins in Industry 4.0**. In *2024 IEEE 7th International Conference on Industrial Cyber-Physical Systems (ICPS)*.2024, pp.1-6, (Presented at the Conference).

- **F. Pires**, B.Ahmad, A.P. Moreira, and P. Leitão. **Recommendation System using Reinforcement Learning for What-if Simulation in Digital Twin**. In *2021 IEEE International Conference on Industrial Informatics (INDIN)*.2021, pp.1-6, doi: 10.1109/INDIN45523.2021.9557372.

- **F. Pires**, B. Ahmad, A. P. Moreira and P. Leitao. **Digital Twin based What-if Simulation for Energy Management**. In *2021 4th IEEE International Conference on Industrial Cyber-Physical Systems (ICPS)*. 2021, pp.309–314, doi: 10.1109/ICPS49255.2021.9468224.

- **F. Pires**, V. Melo, J. Almeida and P. Leitao. **Digital Twin Experiments Focusing Virtualisation, Connectivity and Real-time Monitoring**. In *2020 IEEE Conference on Industrial Cyberphysical Systems (ICPS)*. 2020, pp.309–314, doi: 10.1109/ICPS48405.2020.9274739.

- **F. Pires**, A. Cachada, J. Barbosa, A. P. Moreira and P. Leitao. **Digital Twin in Industry 4.0: Technologies, Applications and Challenges**. In *2019 IEEE 17th International Conference on Industrial Informatics (INDIN)*. 2019, pp.721–726, doi: 10.1109/INDIN41052.2019.8972134.

Book Chapter Publications:

- **F. Pires**, A.P. Moreira and P. Leitão. **Sensitivity Analysis of the SimQL Trustworthy Recommendation System**. In Theodor Borangiu, Damien Trentesaux, Paulo Leitão, Lamia Berrah, Jose-Fernando Jimenez (eds.). *Service Oriented, Holonic and Multi-Agent Manufacturing Systems for Industry of the Future*. vol. 1336, Springer International Publishing, 2023, doi: 10.1007/978-3-031-53445-4_28.

- **F. Pires**, A.P. Moreira and P. Leitão. **Trust Model for Digital Twin Recommendation System**. In Theodor Borangiu, Damien Trentesaux, Paulo Leitão, Olivier Cardin and Laurent Joblot (eds.). *Service Oriented, Holonic and Multi-Agent Manufacturing Systems for Industry of the Future*. vol 1034, Springer International Publishing, 2022, doi: 10.1007/978-3-030-99108-1_11.

- **F. Pires**, M. Souza, B. Ahmad and P. Leitão. **Decision Support Based on Digital Twin Simulation: A Case Study**. In Theodor Borangiu, Damien Trentesaux, Paulo Leitão, Olivier Cardin and Samir Lamouri (eds.). *Service Oriented, Holonic and Multi-Agent Manufacturing Systems for Industry of the Future*. vol. 952, Springer International Publishing, 2021, doi: 10.1007/978-3-030-69373-2_6.

Publications Awards:

- *Best Paper Award of the 19th IEEE International Conference on Industrial Informatics (INDIN21)* regarding the paper **Recommendation System using Reinforcement Learning for What-if Simulation in Digital Twin**, INDIN2021.

- *Best Presentation In Session Award*, in appreciation to the paper **Digital Twin based What-if Simulation for Energy Management**, ICPS2021.

"If we knew what we were doing, it would not be called research, would it?"

Albert Einstein

Contents

List of Figures

List of Tables

Acronyms

3D Tridimensional.

4IR Fourth Industrial Revolution.

ACOS Adjusted Cosine Similarity.

AGV Automated Guided Vehicle.

AI Artificial Intelligence.

C4ISR Command, Control, Communications, Intelligence, Surveillance, and Reconnaissance.

CA Context-Aware.

CBF Content-Based Filtering.

CF Collaborative Filtering.

CNN Convolutional Neural Networks.

COS Cosine Similarity.

CPS Cyber-Physical Systems.

CR Conversation Rate.

CTR Click-Through Rate.

CWP Context-based Performance.

DES Discrete Event Simulation.

DF Demographic Filtering.

DoF Degrees of Freedom.

DSR Design Science Research.

DSS Decision Support Systems.

ED Euclidean Distance.

EIF European Interoperability Framework.

GAN Generative Adversarial Networks.

GCN Graph Convolutional Networks.

GDP Gross Domestic Product.

GNN Graph Neural Networks.

HBF Hybrid-Based Filtering.

ICT Information and Communication Technologies.

IDARTS Intelligent Data Analysis and Real-Time Supervision.

IML Integrated Manufacturing & Logistics.

IoE Internet-of-Everything.

IoP Internet-of-People.

IoT Internet-of-Things.

IT Information Technologies.

JD Jaccard Similarity.

KBA Knowledge-Based Approach.

KPI Key Performance Indicator.

MaaS Manufacturing as a Service.

MAE Mean Absolute Error.

MDP Markov Decision Process.

MES Manufacturing Execution System.

MF Matrix Factorisation.

ML Machine Learning.

MSD Mean Square Distance.

NASA National Aeronautics and Space Administration.

NN Neural Networks.

OPC-UA Open Platform Communications Unified Architecture.

PaaS Product as a Service.

PCC Pearson Correlation Coefficient.

PIP Proximity-Impact-Popularity.

PLCs Programmable Logic Controllers.

PLM Product Life-cycle Management.

PSS Proximity-Significance-Singularity.

RL Reinforcement Learning.

RMSE Root Mean Square Error.

RNN Recurrent Neural Networks.

RS Recommendation System.

SAMBA Self-Aware health Monitoring and Bio-inspired coordination for distributed Automation Systems.

SVD Singular Value Decomposition.

SVM Support Vector Machine.

TBR Trust-Based Recommendation.

TF-IDF Term Frequency-Inverse Document Frequency.

TRL Technology Readiness Level.

UML Unified Modelling Language.

VSS Vector Space Similarity.

WMG Warwick Manufacturing Group.

List of Symbols

u_n	User
i_t	Items (e.g., what-if scenarios)
$U\ (u)$	Set of all the users ($U, u \in U$)
$I\ (i)$	Set of all the possible items that can be recommended ($I, i \in I$)
S_m	Set of what-if scenarios generated by the what-if engine
$Model_n$	Virtual model of the physical scenario
DoF_o^n	Degree of freedom for $Model_n$, $DoF_o^n = \{x \in \mathbb{R} : Min \leq x < Max\}$
NS_m	Number of what-if scenarios generated by the what-if engine
$NDoF_j^n$	Number of degree of freedoms established for $Model_n$
$D_{j,j}^n$	Number of dependencies of the established DoF_j^n
UT_S	User trust in the recommendation system
UT_R	User trust in the given recommendation i
E_R	Expected user u rating for scenario i
A_R	Actual user u rating for scenario i
U_{Acc}	User acceptability is the intention to accept the recommendation i to be applied in the physical system
$Q(s_t, a_t)$	Q-values of the Q-Learning algorithm
s_t	Trust state of the user ($s \in S$)
a_t	Actions which represent all the possible scenarios to be recommended ($a \in A$)
r_t	Reward given by the reward function R
α	Learning rate, , $0 \leq \alpha \leq 1$, establishes the learning pace of the algorithm
γ	Discount factor, $0 \leq \gamma \leq 1$, determining the importance of future rewards compared to future rewards
R_{Value}	Value based on which the recommendation will be performed
$Trust_P$	Average value for positive trust states
$Trust_N$	Average value for negative trust states
$sim(u, v)$	Similarity between user u and user v
$sim(a_t, a_j)$	Similarity between scenario a_t and scenario a_j
C	Shrinking term [1, 10]
$rep(u)$	Reputation of user u
$Trust_{u,v}$	Trust of user u by user v
C_{UT}	Current user trust of the user in the recommendation system
$DTrust_{u,v}$	Direct trust between user u and v
$ITrust_{u,v}$	Indirect trust between user u and v

Chapter 1

Introduction

The ongoing and widespread implementation of Industry 4.0, also known as the Fourth Industrial Revolution (4IR), is characterised by the integration of advanced and intelligent technologies and concepts, such as Cyber-Physical Systems (CPS), Internet-of-Things (IoT), Artificial Intelligence (AI), Big Data Analytics, Cloud Computing, and Digital Twin, into industrial environments (Bauer et al., 2015; Lu, 2017; Pires et al., 2018; Leitao et al., 2020). The integration of these technologies has brought about a new digital transformation era, revolutionising production processes by making them more efficient, flexible, data-driven, and responsive to changes in market demands. However, integrating these digital technologies presents a significant challenge for traditional Decision Support Systems (DSS).

The manufacturing industry of today faces a significant challenge in the field of decision-making since most of the traditional decision-making strategies are based on the decision-makers analysis and empirical knowledge, which can be time-consuming and inaccurate due to the high quantity of data, especially for new decision-makers, who have limited or no historical knowledge about a possible problem (Franke et al., 2022). The traditional DSS are limited to static decision support techniques, which, with the emergence of the Digital Twin technology offering a platform that enables the development of an intelligent, automated, and computerised solution for decision support, capable of providing faster, more accurate, flexible and intelligent decision support (Bisantz and Seong, 2000).

Although the Digital Twin bases its decision support on real-time monitoring, data analysis and what-if simulation, the analysis of all this information would still depend on the decision-maker. Through the integration of a Recommendation System (RS), it will be possible to have cooperative DSS enabling the active performance of decision support by providing recommendations and enabling the decision-maker to have their input in the final decision. Despite the potential benefits that the integration of a RS can have in enhancing decision support, these still suffer from problems such as cold-start, which can be defined as when a system can not work properly due to the lack of information on new users or items, and data sparsity, which is the lack of sufficient rating data for the system to perform accurate recommendations (Guo, 2012; Fletcher, 2017; Nanthini and Pradeep Mohan Kumar, 2023). These two problems make it challenging for RS to perform

recommendations, especially for new users or items with limited or no historical data, posing a significant challenge for decision support.

Given the importance of decision support in Industry 4.0-compliant manufacturing environments, the objective of this thesis is to propose a Digital Twin architecture for decision support based on an innovative RS approach, capable of handling the cold-start and data sparsity challenges in highly dynamic, flexible and complex environments. This work integrates what-if simulation, intelligence, similarity and trust measures to mitigate the mentioned challenges.

The rest of this chapter is structured as follows:

- Section 1.1: provides context for developing a Digital Twin-based DSS integrated with an intelligent RS using similarity and trust measures.
- Section 1.2: outlines the research problem, the main objective, and the research questions that are the focus of the work.
- Section 1.3: elaborates on the research methodology employed in the development of this work.
- Section 1.4: presents an overview of the chapters in this document, which delineate the research, design, development, and experiments conducted throughout this work.

1.1 Context and Motivation

In 2011, the German Federal Government introduced the Industry 4.0 paradigm to develop and stimulate the economy. Since then, it has played a leading role in the global transformation of the manufacturing industry (Kagerman et al., 2013; Leitao et al., 2020; Lu, 2017). Today's manufacturing sector is directly connected to a country's economic development, technological progress and global interconnection. In 2012, globally, manufacturing accounted for approximately 16% of Gross Domestic Product (GDP) and 14% of employment (Manyika et al., 2012). In 2023, manufacturing reached 17.5% of the global GDP, even though the world is still dealing with the repercussions of the pandemic (Thomas, 2023). Manufacturing systems are responsible for creating all the products used worldwide, making them an essential part of everyday life. Implementing the Industry 4.0 paradigm requires the adoption of Information and Communication Technologies (ICT), resulting in real-time responsiveness, reconfigurability, flexibility, and decentralisation. To remain competitive, manufacturers must increase efficiency and agility. Hence, they require intelligent DSS capable of making efficient, accurate, adaptable, and reactive decisions during manufacturing processes (Rosin et al., 2022).

The digitalisation of manufacturing systems through Industry 4.0 technologies, such as IoT and Big Data, has led to the generation of vast amounts of data, which poses a challenge for traditional decision-making processes (Ahuett-Garza and Kurfess, 2018; Li et al., 2022; Khan et al., 2017; Nouinou et al., 2023). Decision-making entails evaluating situations or problems, considering various scenarios or possible solutions, contemplating the most suitable solutions from the

available alternatives, and implementing the most suitable actions. With significant amounts of data, decision-making can be time-consuming since it requires a large quantity of data to be analysed within real-time constraints (Nouinou et al., 2023). Integrating Industry 4.0 technologies, such as Digital Twin and AI, with DSS could improve the decision-making process, reduce the amount of data for analysis, and improve productivity in flexible manufacturing systems. In traditional manufacturing systems, decision-making is often manual and dependent on the decision-makers experience and empirical knowledge, which can limit the speed and accuracy of the decisions (Kunath and Winkler, 2018). This can be incredibly challenging for new decision-makers or when the manufacturing system requires flexibility and faster decision-making (Franke et al., 2022). The integration of AI-based algorithms into DSS can help improve the decision-making process, ensuring compliance with the defined requirements.

Since the 1970s, DSS have been developed to support complex decision-making and problem-solving (Felsberger et al., 2016). With Industry 4.0 technologies, the application of intelligent DSS in manufacturing systems has significantly evolved, integrating real-time, AI, and predictive analytics. Intelligent DSS can learn, provide tailored suggestions, and offer automated support, transforming traditional systems into intelligent systems (Liang, 2008; Simeone et al., 2021).

Integrating decision support methods, such as RS, can simplify decision-making processes. RS is a subclass of information filtering systems that are particularly good at providing recommendations based on the decision-makers trust level and supporting decision-making with large amounts of data (Selmi et al., 2016; Ricci et al., 2015). This enables more comprehensive support of decision-makers throughout the manufacturing process's life cycle.

Automated and computerised DSS in manufacturing is becoming more prevalent. However, it is important to consider the issue of trust, as it can affect the reliability and dependability of the system (Bisantz and Seong, 2000). Trust can be viewed from different perspectives regarding decision support, such as social trust, which can act as the basis for recommendations, or the trust decision-makers place in the DSS. This is a crucial factor in designing a DSS, as it can influence whether or not the provided support and the system are accepted or rejected (Madhavan and Wiegmann, 2007). Unfortunately, decision-makers often underestimate the decisions provided by a DSS, but by incorporating the concept of trust, the acceptance of the system and its decisions can be improved. Low levels of trust in a DSS can lead to non-use of the system, while high levels of trust can lead to DSS-induced complacency, so finding the right balance is key (Seong et al., 2006). Measuring trust in a DSS is also an issue, as no standard evaluation method exists. In the case of RS, some implementations already utilise the trust concept of decision-makers as a basis for their support (O'Donovan and Smyth, 2005; Victor et al., 2011; Selmi et al., 2016).

RS have proven to be helpful decision support tools in various fields, including e-commerce, transportation, entertainment, e-health, and agriculture (Fayyaz et al., 2020). In the manufacturing industry, the implementation of RS offers a powerful solution to the complex challenges faced by this sector. These challenges include data complexity, quality, availability, real-time processing, the cold-start problem, and model interpretability (Fayyaz et al., 2020; Meyer, 2012). The cold-start problem is a common limitation when the RS lacks sufficient information to function opti-

mally (Son, 2016). This can happen when new products, processes, equipment, or decision-makers are introduced, leading to a lack of data to provide accurate recommendations. Fortunately, several techniques can be employed to overcome this issue, including analysing recommended item features, utilising similarity measures between items and/or decision-makers, identifying correlations, utilising social relationships between decision-makers, combining multiple recommendation techniques to overcome limitations, and integrating human expertise in the recommendation process (Chen et al., 2013; Jain and Mahara, 2019; Guo, 2013). A combination of an AI-based algorithm with trust and similarity measures could effectively address the challenges the cold-start problem poses.

Intelligent manufacturing relies on the integration and interconnection between the physical and digital worlds (Li et al., 2022). This is where the concept of Digital Twin comes into play, as it is the key to making effective decisions within the Industry 4.0 framework (Yokogawa, 2019). Although Michael Grieves introduced the Digital Twin concept in 2002 (Grieves and Vickers, 2017), it only recently became popularised with Industry 4.0 (Neto et al., 2021). In simple terms, the Digital Twin concept refers to a virtual replica of a physical system (such as equipment, a process or an entire plant) that is connected to sensors and embedded devices collecting data (including engineering, operational and behavioural data) in real-time to a simulation model (Li et al., 2022; Pires et al., 2019). This model can help perform various simulation analyses, such as what-if analysis and predictive analysis (Pires et al., 2019; Kunath and Winkler, 2018). In addition, the collected data helps to monitor the system's behaviour and predict its performance. Integrating RS with Digital Twin in manufacturing systems has proven beneficial in cost reduction, service improvement, personalised process optimisation, and real-time decision-making (Kunath and Winkler, 2018; Jones et al., 2020). However, analysing and correlating all the data provided by different modules can be challenging for the decision-maker. Nevertheless, the fully integrated Digital Twin makes data analysis and simulation the key decision support tools (Kunath and Winkler, 2018).

The traditional decision-support approaches used in Industry 4.0-based manufacturing systems fall short of meeting the requirements of the new business model that emphasises flexibility, responsiveness, reconfigurability, decentralisation, and real-time demands. These approaches rely on batch processing and rigid models, which makes it difficult to adapt quickly and limits integration. As a result, data silos are created, preventing a comprehensive view of the entire system and limiting learning and predictive capabilities. This highlights several research opportunities for decision support in manufacturing systems, mainly focused on the Industry 4.0 paradigm and the evolving requirements of manufacturing systems.

1.2 Research Problem & Objectives

Given the circumstances of traditional decision support in manufacturing systems detailed in the preceding section, there is a rising need for an intelligent DSS capable of responding to the problems posed by the new business model based on Industry 4.0 manufacturing environments. Considering the aspects identified in the previous section, it was possible to design a schema, illustrated

in Figure 1.1, of the identified research problem with traditional DSS in Industry 4.0 compliant-manufacturing systems.

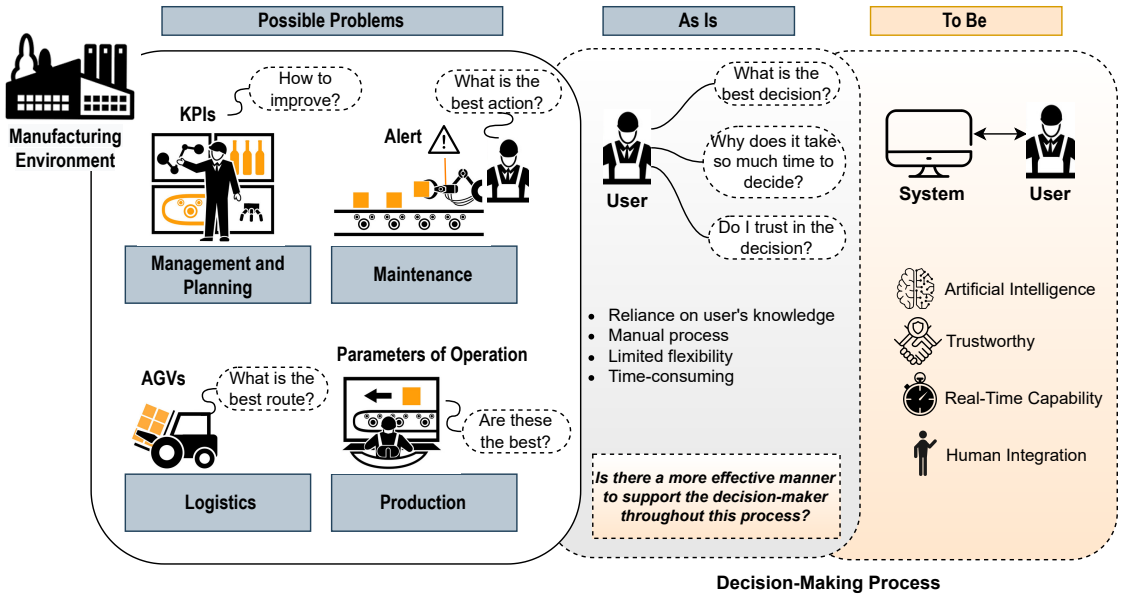

Figure 1.1: Systematisation of the research problem.

The manufacturing environment is prone to various kinds of problems, such as management and planning, improving the Key Performance Indicator (KPI)s, sudden alerts in maintenance, determining the best logistics route or number of Automated Guided Vehicle (AGV)s for the production line, and setting optimal operation parameters in production. These problems require quick solutions, but traditional decision-making processes are time-consuming and limited in flexibility and number of possibilities that can be analysed in a timely manner. It is a manual process that relies heavily on the decision-maker's knowledge, making it difficult for new decision-makers to make accurate and efficient decisions. *Is there a more effective manner to support the decision-maker throughout this process?*

The integration of Industry 4.0 technologies, as AI and Digital Twin, into a DSS enables the mitigation of some of the identified challenges of the traditional decision-making process, enabling the system to handle large amounts of data that can come from the digitalised manufacturing environment, and the support for new decision-makers without any previous knowledge. Also, allowing human integration in the decision-making cycle with the intelligent system promotes user confidence in the system and its recommendations. Taking into consideration these circumstances and the identified research problem, the main challenge that will be the focus of this work is to develop an intelligent and trust-based DSS capable of providing support to the decision-maker in its decision-making cycle in a timely manner in response to the fast-changing manufacturing environment.

Based on the provided context, motivation and research problem, this work aims to achieve the following thesis statement:

The Digital Twin-based architecture integrated with recommendation system can provide decision support to the decision-makers enabling personalised, interactive, trust-based and intelligent recommendations, and mitigate the cold-start and data sparsity problem.

Considering the proposed thesis statement, this thesis intends to propose a Digital Twin-based architecture for decision support and develop an intelligent and trust-based recommendation model capable of providing support to the decision-maker in its decision-making cycle in a timely manner in response to complex and fast-changing manufacturing environments, particularly focusing on the challenges of data sparsity and cold-start problems.

In order to address the research problem and thesis statement, the following research questions were established for this dissertation:

- **RQ 1:** *In which way the integration of AI-based algorithms and trust model in the Digital Twin-based architecture can enhance the RS to provide personalised recommendations for flexible and dynamic manufacturing environments?*

The proposed DSS is based on a Digital Twin-based architecture, which will integrate RS and what-if simulation module to enable faster and personalised decision support. Although the implementation of traditional RS could improve the decision-making cycle in the manufacturing domain, the integration of AI-based algorithms can improve the analysis capabilities of vast amounts of data, better customisation of the recommendations, faster and more informed decisions, and learning capabilities. Integrating a trust model can ensure the recommendations' reliability, accuracy, quality, and transparency and implement mitigation strategies as similarity measures for the cold-start problem.

- **RQ 2:** *In which way the similarity measures, focusing both on items and decision-makers, can accelerate the learning process in cold-start environments?*

Apart from the reliability and time requirements that the decision support architecture will have to ensure, it will also have to be able to minimise the problems caused by cold-start problems. The definition of mitigation measures (e.g., similarity measures) that can improve recommendation quality and speed is an important development for AI and trust-based RS in a manufacturing sector with time requirements.

1.3 Research Methodology

One of the first steps in performing research is choosing the research methodology. A research methodology can be considered the overall strategy to achieve the main goal and objectives of the research (Sutrisna, 2009). Two main factors to consider when choosing a research methodology are the alignment with the research objectives and the nature of the research problem. In this case, the research developed in this dissertation follows a classical Design Science Research (DSR) methodology based on an iterative process. The chosen research methodology was proposed by

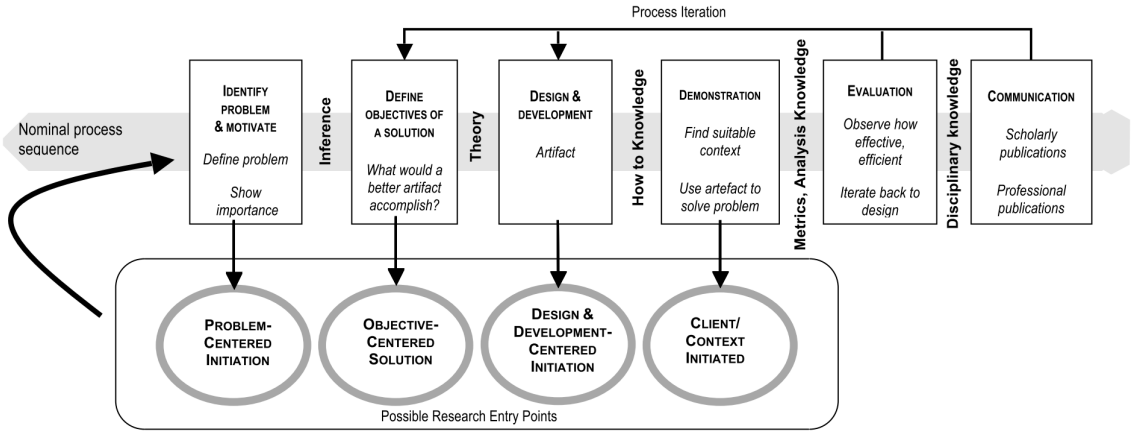

Figure 1.2: The iterative process of the design science research methodology (Peffers et al., 2007).

Peffers et al. (2007), which follows the DSR methodology for information systems. Figure 1.2 illustrates the DSR methodology.

The DSR methodology consists of a six-step nominal sequential iterative process, meaning that at any step of the methodology, it is possible to evaluate the outputs and go back to previous steps to improve or redefine parameters. The steps defining this methodology are briefly described, indicating how these were applied in developing this research work.

The first step in the research methodology is the *Problem identification and motivation*, which translates to defining a specific research problem and why it is essential to its resolution. Usually, this step comprises a literature review which offers the context of the domain. Regarding this work, Sections 1.1 and 1.2 provide the context, motivation and the proposed research problem, which are supported by the literature review performed in Chapter 2. Step two is related to *Define Objectives for a Solution* that infers the objectives and requirements for a solution for the research problem by considering what is possible and feasible. Considering this work, the main objectives and requirements are presented in Section 1.2. The following step is the *Design and Development* of certain artefacts (i.e. models, methods or instantiations), defining its functionality and architecture, ending up by creating the actual artefact. This work is presented in Chapter 3 providing the definition of a general Digital Twin architecture and modules for decision support and its implementation, respectively. The fourth step is the *Demonstration*, where the artefact is applied to solve one or more scenarios or case studies from the problem, which is defined in Chapter 4. The fifth step *Evaluation* is where the artefact is evaluated in how well it supports the solution of the problem. The evaluation is performed by comparing the expected objectives with the actual results from the application of the artefact. These last two steps are presented and discussed in Chapter 5, for which the proposed artefact is demonstrated and evaluated in a case study from an academic and real-world perspective. Finally, the last step in this research methodology is *Communication*. This step is closely related to disseminating the attained results and the transfer of knowledge and academic results (Peffers et al., 2007). In this case, this is performed by this report and by the

scientific publications published in conferences and journals throughout the development of the work.

Although the research methodology is postulated to be performed in sequential order, there is no obligation to do so. The process can be at any stage and then work backwards to ensure the rigour applied to the design process (Peffers et al., 2007). The DSR methodology applied to develop the work proposed in this dissertation implies refining the proposed research questions and thesis statement throughout the research process.

1.4 Dissertation Organisation

The organisation of this dissertation is described in this subsection. This document is divided into six chapters, starting with the present chapter that contextualises the research work, including the research questions and hypothesis, the main objectives and contributions of this dissertation.

Chapter 2, entitled *"Related Work"*, provides a contextualisation and literature review of concepts, technologies and challenges in the development of a Digital Twin-based architecture for decision support. Considering that the proposed architecture integrates RS an overview of the state-of-the-art recommendation approaches and the new enablers of RS, as trust, intelligence and similarity. Characterisation of the cold-start and data sparsity problems will be provided, focusing on the state-of-the-art approaches that tackle these challenges.

Chapter 3, entitled *"SimQL Trust-based Recommendation Model"*, presents the Digital Twin-based architecture, describing each layer, focusing more on *Simulation Layer*, *Decision Support Layer*, and *Human Trust Layer*. Presents the formalisation of the *SimQL* trust-based recommendation model, focusing on the mitigation strategies based on trust and similarity measures for the cold-start and data sparsity challenges, and the definition of different recommendation strategies for the presented recommendation environment.

Chapter 4, entitled *"Case Study and Evaluation Measures"*, describes the proposed case study of a battery pack assembly line, focusing on the description of the case study and the problem statement. Lastly, the performance measurement procedure and metrics are defined to validate the different aspects of the proposed approach.

Chapter 5, entitled *"Experimental Validation and Results"*, presents the preliminary experiments validating each component of the approach, the what-if simulation, the RL algorithm and the similarity and trust measures, and it also presents the comparison of proposed *SimQL* trust-based recommendation model with state-of-the-art approaches. All the experiments are based on the previously defined case study.

Finally, the dissertation is round-up with Chapter 6, entitled *"Conclusions and Future Work"*, which describes the significant conclusions reached during the development process and the main contributions of this thesis aligned them to the research questions and hypothesis presented in Chapter 1. The chapter is finalised by outlining future research trends and guidelines.

Chapter 2

Related Work

The rise of digitalisation in manufacturing has resulted in a greater reliance on ICT and data generation within industrial settings. However, the current market demands have placed significant pressure on manufacturers to respond quickly, making traditional decision-making during production processes (e.g., maintenance, logistics and operations) more challenging. To address this, manufacturers require an advanced and intelligent DSS capable of providing timely and valuable insights, suggestions, and recommendations based on vast amounts of data or without enough data. These systems can enable manufacturers to make more informed and timely decisions, reduce downtime, minimise waste, enhance productivity, and ultimately gain a competitive advantage in the industry.

In the given context, the Digital Twin concept offers a decision support platform by creating a virtual replica of a physical system, process or entity, such as a manufacturing environment or production process. This virtual counterpart allows for real-time monitoring, data analysis, and simulation, enabling decision-makers to infer valuable insights, which support the decision-maker by presenting a large amount of data from which value and knowledge should be extracted based on previous knowledge. By integrating a RS into the Digital Twin, it will be possible to amplify the benefits of both technologies, providing a more personalised, adaptive and efficient approach to decision support in dynamic and complex environments. However, one of the major problems in having a highly dynamic, flexible and complex environment is the lack of historical and initial data to perform recommendations, also known as the cold-start and data sparsity problem. To mitigate this, the integration of an AI-based algorithm as a recommendation algorithm that enables active learning encouraging decision-maker interaction and feedback, with similarity measures, which help to identify patterns and similarities between users or items, and trust-based mechanisms, considering user preferences and preferences of trusted users, can be a powerful solution for mitigating the cold-start problem.

This chapter thoroughly analyses existing literature to establish an up-to-date theoretical background and supporting concepts and technologies related to the Digital Twin for decision-support purposes integrated with RS to handle the cold-start and data sparsity problems. This chapter will cover the following topics:

- Section 2.1: presents an overview of the main aspects of the 4IR paradigm, contextualising and focusing on the Digital Twin technology. Clarifying the definition, technologies, application domains and key functionalities of Digital Twin, mainly focusing on decision support.
- Section 2.2: presents an overview of RS for decision support focusing on the formal description of the recommendation process, the properties and requirements, the state-of-the-art approaches and the key challenges.
- Section 2.3: discusses the main aspects regarding the cold-start and data sparsity RS challenges, presenting a bibliometric study assessing, in general, the scientific landscape, and a characterisation study presenting the approaches that focus on these problems, including the enabling technologies and application domains.
- Section 2.4: explores the enabling methods identified for handling the cold-start and data sparsity problems, such as trust, intelligence and similarity.
- Section 2.5: presents the previous sections' summary highlighting this chapter's main takeaways.

2.1 Digital Twin in the Era of 4th Industrial Revolution

The 4IR has significantly changed various industries, including manufacturing, healthcare, energy, transportation, and construction. The core of 4IR lies in automation, communication, and cyber-physical integration, enabled by the use of advances ICT, such as IoT, Cloud Computing, Big Data, CPS, and AI.

2.1.1 Industry 4.0 - the 4th Industrial Revolution

The manufacturing world has been evolving through what has been commonly called "industrial revolutions". To this day, four time periods have been identified that can be considered industrial revolutions, as illustrated in Figure 2.1.

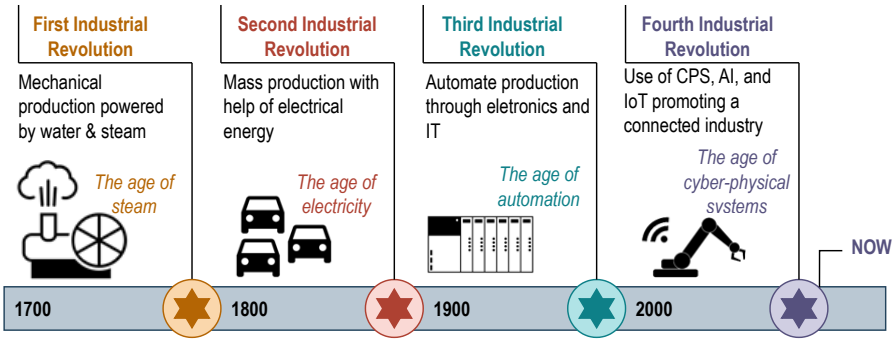

Figure 2.1: Industrial revolutions throughout time.

The first industrial revolution occurred towards the end of the 18^{th} century, integrating mechanical production with water and steam. The second industrial revolution started around the beginning of the 20^{th} century, introducing the conveyor belt and mass production. The third industrial revolution began around the 70s with the rise of electronics, with two significant inventions, namely Programmable Logic Controllers (PLCs) and robots, transforming existing production into automated production. Nowadays, the 4IR emerged and uses CPS as the backbone, the confluence between the physical and virtual parts of a system, and IoT and AI as main digital technologies (Bloem et al., 2014).

The first time that the 4IR was brought to the world was in 2011, at the Hanover Fair, Germany (Xu et al., 2018). The German government launched an initiative called *Industrie 4.0*, with the sole purpose of driving digital manufacturing forward. According to Kagerman et al. (2013), Industry 4.0 can be defined as a change in the manufacturing paradigm through a new business model focusing on product customisation and digitalisation throughout its life cycle by embedding ICT technologies. Assessing the changes at a higher level between the current manufacturing environment and an Industry 4.0-based manufacturing environment, the most significant changes will be the flexibility, modularity, and reconfigurability of the systems to allow the production of customised individual products, the integration of information end-to-end, and the connection between the real and virtual worlds.

The integration of systems within the Industry 4.0 paradigm can be divided into three types of integration:

- *Horizontal Integration*, which represents the connection, coordination and integration of Information Technologies (IT) technologies between the overall value chain of a company (e.g., suppliers, materials, logistics) to transform it into a value network, connecting with different companies, to increase the efficiency in productivity;

- *Vertical Integration*, which describes the integration of IT within the hierarchical levels (e.g., actuator and sensor level, manufacturing level, production management level) of a value creation module in manufacturing systems through cells, lines and factories, also integrating activities as marketing, sales and technology development, delivering an end-to-end solution;

- *End-to-end Integration*, which comprehends the integration throughout the entire product life cycle throughout the engineering process, integrating the virtual and real-world across the product value chain and network (Peres et al., 2018; Xu et al., 2018; Stock and Seliger, 2016).

2.1.1.1 Design Principles

The Industry 4.0 paradigm has unique characteristics, represented by six design principles serving as guidelines for its implementation. The six design principles are *Interoperability*, *Virtualisation*,

Real-time capability, *Decentralisation*, *Service orientation*, and *Modularity* (Hermann et al., 2015; Habib and Chimsom I., 2019; Ghobakhloo, 2018).

Interoperability allows different systems and devices to communicate and share data through technologies such as IoT, Internet-of-People (IoP), Internet-of-Everything (IoE), and CPS. Frameworks like Command, Control, Communications, Intelligence, Surveillance, and Reconnaissance (C4ISR) (Sowell, 2006), ATHENA (Berre et al., 2007), and European Interoperability Framework (EIF) (Commission, 2017) integrate these systems at four levels: operational, systematic, technical, and semantic. At each level, specific concepts are defined, guiding principles established, and tools integrated to support the technologies. *Virtualisation* merges the physical system data with simulated models for optimised processes. Digital Twin technology creates a virtual replica, modelling, testing and validating throughout its life cycle. Virtualisation offers benefits such as designing and testing models and prototypes, workforce training, and customer involvement in product design, and has *real-time capabilities*. *Real-time capability* is a critical design principle that involves responsiveness, reliability, fault tolerance, availability, maintainability, and functional safety. It requires real-time data analysis, decision-making and support, and cybersecurity attack detection. Two frameworks, Intelligent Data Analysis and Real-Time Supervision (IDARTS) (Peres et al., 2018) and Self-Aware health Monitoring and Bio-inspired coordination for distributed Automation Systems (SAMBA) (Siafara et al., 2017), can help achieve this capability. Industry 4.0 technologies are increasing the demand for customisation in manufacturing. *Decentralisation* allows each system component to work independently and autonomously, making real-time decisions. This approach is facilitated by quality assurance, traceability, self-regulation, and intelligent control systems, enabling horizontal and vertical integration of processes and decisions. *Service Orientation* is the design principle of offering a system's functionalities and products as services. It includes Manufacturing as a Service (MaaS) and Product as a Service (PaaS). MaaS involves collaborative manufacturing through different companies offering manufacturing services, while PaaS promotes products as services or virtual experiences. *Modularity* is a key design principle that enables easy system reconfiguration, promotes flexibility, and facilitates plug-and-play capabilities. It allows for agile, adaptable, and flexible manufacturing systems and enables new modules and systems to be integrated seamlessly without disrupting the initial infrastructure. This principle is supported by personalised product manufacturing at all levels.

The conjugation of all six design principles represents an Industry 4.0 environment with flexibility, modularity, and interconnection between the different systems and plains (physical and virtual) through virtualisation, with implementation and data collection and data analysis in real-time, enabling decision support for increasing the system's efficiency.

2.1.1.2 Enabling Technologies

In the beginning, Industry 4.0 referred to a set of technologies that enable industrial automation, such as CPS, IoT, AI, Big Data, and Cloud Computing, among others, which acted as enablers in implementing the Industry 4.0 paradigm. Since the introduction in 2011 of the strategic initiative

of the German government *Industrie 4.0*, Industry 4.0 has evolved significantly, with many new and promising technologies being developed in the last decade.

A bibliometric study was conducted to identify the latest technologies that facilitate the implementation of the Industry 4.0 paradigm. A well-established bibliometric research methodology was followed to achieve this, and quantitative techniques were applied to bibliometric data. The methodology used for bibliometric analysis enables the exploration and analysis of vast amounts of scientific data, making it possible to identify emerging areas or technologies and evolutionary nuances in a particular field (Donthu et al., 2021). Figure 2.2 illustrates the details of this methodology.

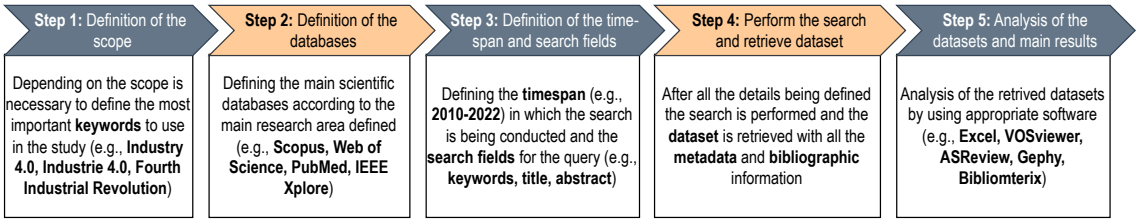

Figure 2.2: General bibliometric methodology.

The bibliometric methodology is divided into five steps, being *Step 1*, the definition of the scope, in this case, it was defined that the scope of the analysis was the Industry 4.0 paradigm, and the most important keywords were used to construct a query as follows,

```
TITLE-ABS-KEY("industry 4.0" OR "industrie 4.0" OR "fourth industrial revolution") AND PUBYEAR > 2009 AND PUBYEAR < 2024
```

In *Step 2*, where the definition of the databases is conducted, in this study, it was considered only one database, Scopus, since it is one of the most comprehensive scientific databases with over 90 million records, including 7000 publishers (Sco, 2023). The timespan and the search fields are defined in *Step 3* to build the query. For this case, it was taken into consideration the publications made between 2010 and 2023, considering the publications that have the selected keywords in one of these fields of the publications, *"Title", "Abstract" or "Keywords"*, it was identified 36.623 publications. The final dataset was limited to publications written in *English* and in the final publication stage, resulting in a dataset of 34.470 publications. After this, the dataset was retrieved (*Step 4*), and an analysis was conducted, *Step 5*, using the VOSviewer[1] and Excel software packages. Figure 2.3 presents the author's keywords co-occurrence network map of the literature on Industry 4.0, allowing the identification of the enabling technologies associated with this paradigm.

In this type of network, the nodes' size indicates the keywords' frequency, and the links between two nodes indicate the strength of the co-occurrence between keywords. It is also important to consider the proximity of the keywords, representing the set of keywords that are mostly used together. The network is also divided into clusters represented by different colours, in this case, 4 clusters (*Red, Yellow, Green, Blue*), each denoting distinct thematic areas. The most frequent

[1]Software used for and visualising bibliometric networks.

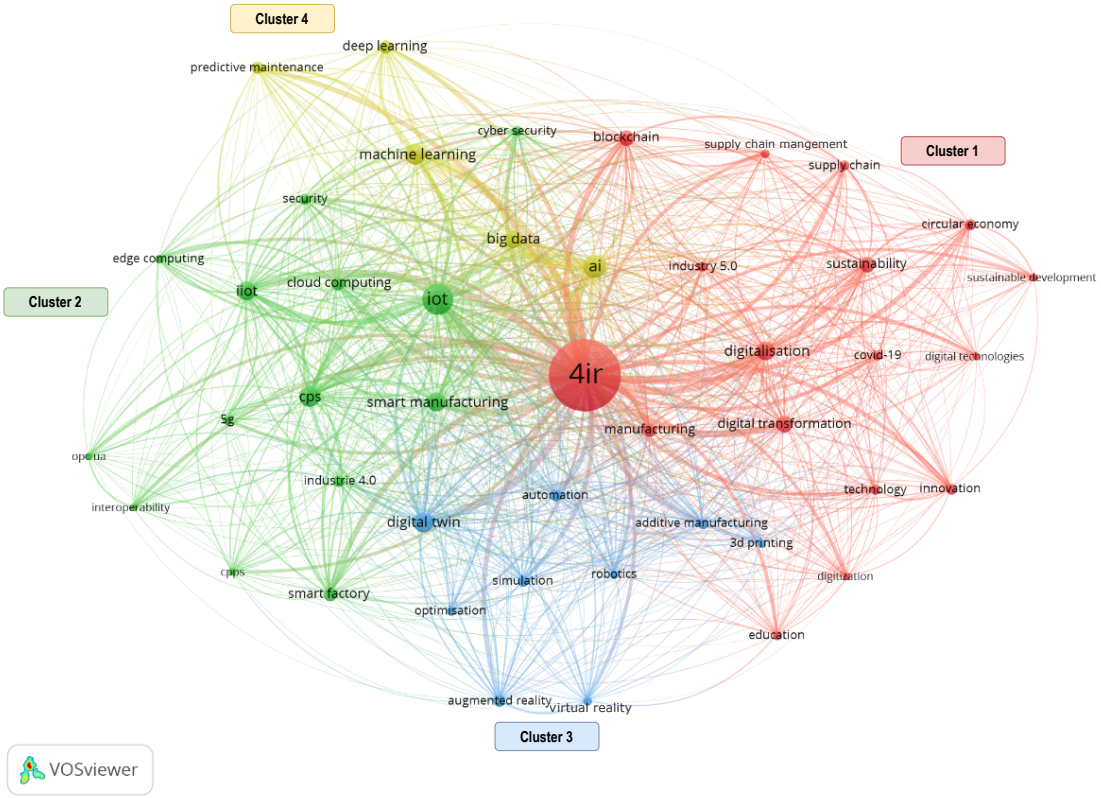

Figure 2.3: Authors keywords co-occurrence network map of the literature on Industry 4.0 (Timeframe 2010-2023; *n*= 52066 keywords; threshold of 200 occurrences per keyword, display 45 keywords), with four clusters.

keywords in this dataset are *Industry 4.0* related labelled as "4ir", *Internet-of-things* labelled as "iot", *Cyber-Physical System* labelled as "cps", *Artificial Intelligence* labelled as "ai", *Machine Learning*, *Digital Twin*, *Big Data*, *Cloud Computing*, *Blockchain*, *Smart Manufacturing*, and *Digitalisation*. According to Rüßmann et al. (2015); Vaidya et al. (2018); Forcina and Falcone (2021); Moeuf et al. (2018), there are at least nine groups of technologies that were the enablers of Industry 4.0, such as CPS, IoT, *Cloud Computing*, *Big Data*, *Cyber-Security*, *Additive Manufacturing* or *3D Printing*, *Augmented Reality*, *Robotics*, and *Simulation*. Several of these pillars of technological advancement can be identified in the authors' keywords network. Additionally, emerging technologies like *Blockchain*, *Edge Computing*, *Virtual Reality*, *5G*, and *Digital Twin* have been added along the way. The Digital Twin technology, in this group of keywords, can be found in Cluster 3 (*Blue Cluster*), which can be labelled as a "3D and Simulation" cluster, being composed of keywords as *additive manufacturing, 3d printing, augmented reality, digital twin, simulation, virtual reality, optimisation, robotics, automation*. Although it was not initially defined as an enabler technology, the Digital Twin technology is becoming a key enabler for implementing an Industry 4.0 environment (Corallo et al., 2021; Madni et al., 2019; Pang et al., 2021; Leng et al., 2021; Tao et al., 2019a).

2.1.2 Digital Twin - The Key Enabler Technology

Digital Twin has become a crucial element of smart manufacturing through the Industry 4.0 paradigm, with attention from academia and industry alike (VanDerHorn and Mahadevan, 2021). The results from the bibliometric study on Industry 4.0 identified that the Digital Twin is a key and evolving technology. Based on this, a new bibliometric search was performed in Scopus, considering the time frame of 2010 to 2023 on the topic *"Digital Twin"* occurring in the title, abstract or keyword of a publication (`TITLE-ABS-KEY("digital twin") AND PUBYEAR > 2009 AND PUBYEAR < 2024`), having been identified as a total of 17.500 publications. The final dataset includes only English-written and final-stage publications, resulting in 16.068 publications. In order to assess the evolution of the concept in terms of research, a graph was created based on the results, including the number of publications throughout time and the type of publication (e.g., conference paper, article, others). Figure 2.4 presents the results regarding the number of scientific publications about the Digital Twin publications within the specified time frame.

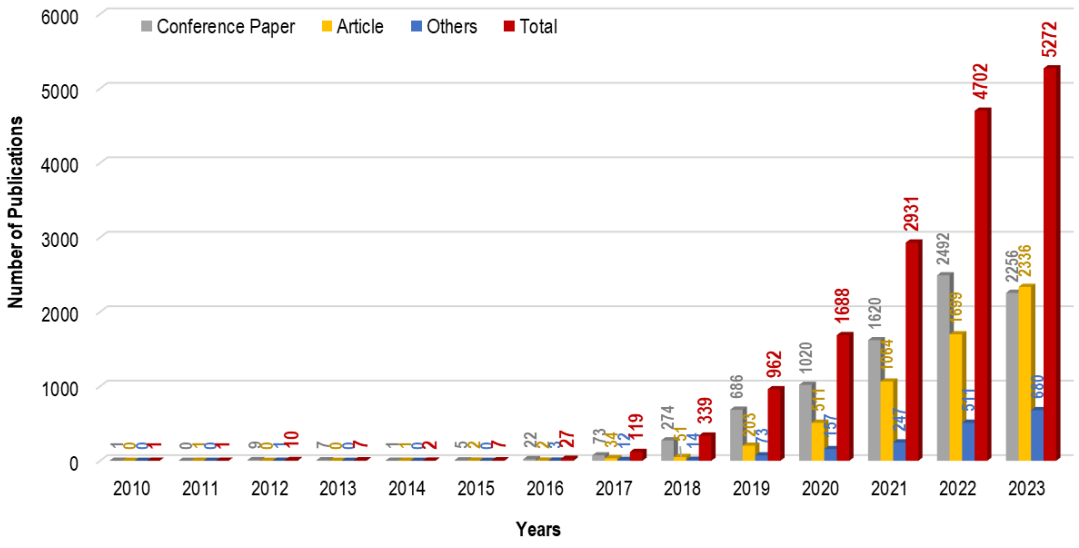

Figure 2.4: Number of publications about the Digital Twin by publication type.

From 2010 until the end of 2023, there has been a significant exponential growth in the number of publications about Digital Twin. After 2015, Digital Twin publications gained more prominence, with a surge in conference and journal papers. The 'Others' category comprises publications classified as reviews, short surveys, books, book chapters, editorials, erratum, letters, data papers, and notes. The number of these publications is less expressive than the conference and journal publication numbers. By evaluating the evolution in the number of journal publications concerning the Digital Twin topic, it is possible to infer that the topic has matured in terms of research being conducted since it is expected that journal publications are the result of more established research. The rise of Industry 4.0 and the emphasis on the connection between physical and

virtual spaces has driven the growing academic interest in Digital Twin technology. This technology enables cyber-physical integration, which bridges the physical and virtual worlds, enabling smart production and manufacturing associated with Industry 4.0.

First introduced in a presentation on Product Life-cycle Management (PLM) in 2002 by Michael Grieves, the Digital Twin concept consisted of a real space, a virtual space and a link for data flow between the two spaces, similar to the concept of predictive control established in the 80s (Grieves and Vickers, 2017; Peterka, 1984). In 2011, National Aeronautics and Space Administration (NASA) adopted the Digital Twin concept to integrate a high-fidelity simulation model with the health management system and historical data of aircraft for improved safety and reliability (Glaessgen and Stargel, 2012). With the emergence of Industry 4.0 technologies like, e.g., IoT, AI, Big Data and Cloud Computing, the Digital Twin concept has rapidly evolved, especially in the manufacturing industry (Madni et al., 2019). According to Lee et al. (2013), incorporating these emerging technologies into manufacturing systems is essential to enhance efficiency and productivity, with Digital Twin technologies playing a crucial role in the industry's future. One of the earliest industrial applications of Digital Twin in manufacturing was optimising shop-floor production (Tao and Zhang, 2017).

Understanding the definition and evolution of the Digital Twin is crucial since this is a key enabler of technology for smart manufacturing. The definition of Digital Twin has evolved from static to dynamic, reflecting this technology's growing understanding and development. Table 2.1 reflects some of the different definitions of Digital Twin in the literature throughout time.

Originally, "Digital Twin" referred to creating a virtual replica of an aircraft's structure in the aerospace industry (Shafto et al., 2010; Tuegel et al., 2011; Gockel et al., 2012). However, the concept was later adopted by Lee et al. (2013) to the predictive health management industry outside the aerospace field. Since the definition of Industry 4.0 in 2015, the term "Digital Twin" has expanded to include replicating products, systems, and processes (Rosen et al., 2015; Grieves and Vickers, 2017). Initially, the virtual replica was only focused on simulation, however, with the growing implementation of the IoT, it evolved into the integration of physical and virtual systems and the use of real-time data for monitoring and optimisation since 2017 (Boschert and Rosen, 2016; Stark et al., 2017; Negri et al., 2017). The definition of Digital Twin has since further developed to encompass a virtual model of a physical asset or system that can be used for simulation, optimisation, monitoring, and decision support. This integration involves real-time data and advanced technologies such as AI, Big Data, IoT, and Machine Learning (ML) (Liu et al., 2018; Macchi et al., 2018; Tao et al., 2019b; Pires et al., 2019; Rasheed et al., 2020; Botín-Sanabria et al., 2022). According to the Digital Twin Consortium, a Digital Twin is *"a virtual representation of real-world entities and processes, synchronised at a specific frequency and fidelity"* (DTc, 2020). This definition is also reinforced by the 23247 ISO standards (ISO 23247, 2021).

This evolution has led to a paradigm shift in manufacturing, enabling predictive maintenance, performance optimisation, and enhanced decision-support. The modern definition of a Digital Twin in manufacturing encompasses a holistic and interconnected ecosystem, fostering unprece-

Table 2.1: Definitions of Digital Twin.

Reference	Definition Digital Twin
Shafto et al. (2010)	*"an integrated multi-physics, multi-scale, probabilistic simulation of a vehicle or system that uses the best available physical models, sensor updates, fleet history, etc., to mirror the life of its flying twin. It is ultra-realistic and may consider one or more important and interdependent vehicle systems"*
Tuegel et al. (2011)	*"is ultrarealistic in geometric detail, including manufacturing anomalies, and in material detail, including the statistical microstructure level, specific to this aircraft tail number."*
Gockel et al. (2012)	*"ultra-realistic, cradle-to-grave computer model of an aircraft structure that is used to assess the aircraft's ability to meet mission requirements"*
Lee et al. (2013)	*"the coupled model of the real machine that operates in the cloud platform and simulates the health condition with an integrated knowledge from both data driven analytical algorithms as well as other available physical knowledge"*
Rosen et al. (2015)	*"Very realistic models of the process current state and its behavior in interaction with the environment in the real world"*
Grieves and Vickers (2017)	*"a set of virtual information constructs that fully describes a potential or actual physical manufactured product from the micro atomic level to the macro geometrical level. At its optimum, any information that could be obtained from inspecting a physical manufactured product can be obtained from its Digital Twin."*
Negri et al. (2017)	*"as the virtual and computerized counterpart of a physical system that can be used to simulate it for various purposes, exploiting a real-time synchronization of the sensed data coming from the field; such a synchronization is possible thanks to the enabling technologies of Industry 4.0"*
Liu et al. (2018)	*"a living model of the physical asset or system, which continually adapts to operational changes based on the collected online data and information, and can forecast the future of the corresponding physical counterpart."*
Pires et al. (2019)	*"as the digital copy of a physical object or system, that is connected and shares functional and/or operational data. The real-time and historical data will be used for assessment of the conditions of the physical asset, to perform optimisation and prediction through the use of machine learning algorithms and simulation techniques."*
Rasheed et al. (2020)	*"a virtual representation of a physical asset enabled through data and simulators for real-time prediction, optimization, monitoring, controlling, and improved decision making"*
Botín-Sanabria et al. (2022)	*"is a virtual representation of a physical object or process capable of collecting information from the real environment to represent, validate and simulate the physical twin's present and future behavior"*

dented levels of efficiency, adaptability, and innovation, playing a crucial role in shaping the future of smart manufacturing.

Based on the definitions proposed in the literature, a conceptual architecture was designed to understand the definition and functionality of the Digital Twin concept (Parrott and Lane, 2017). Figure 2.5 comprises the proposed conceptual architecture with a six-phase approach that includes phases as *Modelling*, *Communication*, *Aggregation*, *Analysis*, *Knowledge*, and *Actuation*.

The Digital Twin concept involves two worlds: the physical world and the virtual world. The *Modelling* phase implements IoT sensing technologies to create the virtual model (e.g., 3D model, equation, simulation model) of the physical asset and its environment. Followed by the *Communication* phase, in which are established the communication protocols (e.g., Modbus, Open Platform Communications Unified Architecture (OPC-UA)) for sending information to the virtual model in the virtual world. The transferred data can be executed at three speeds: near real-time, real-

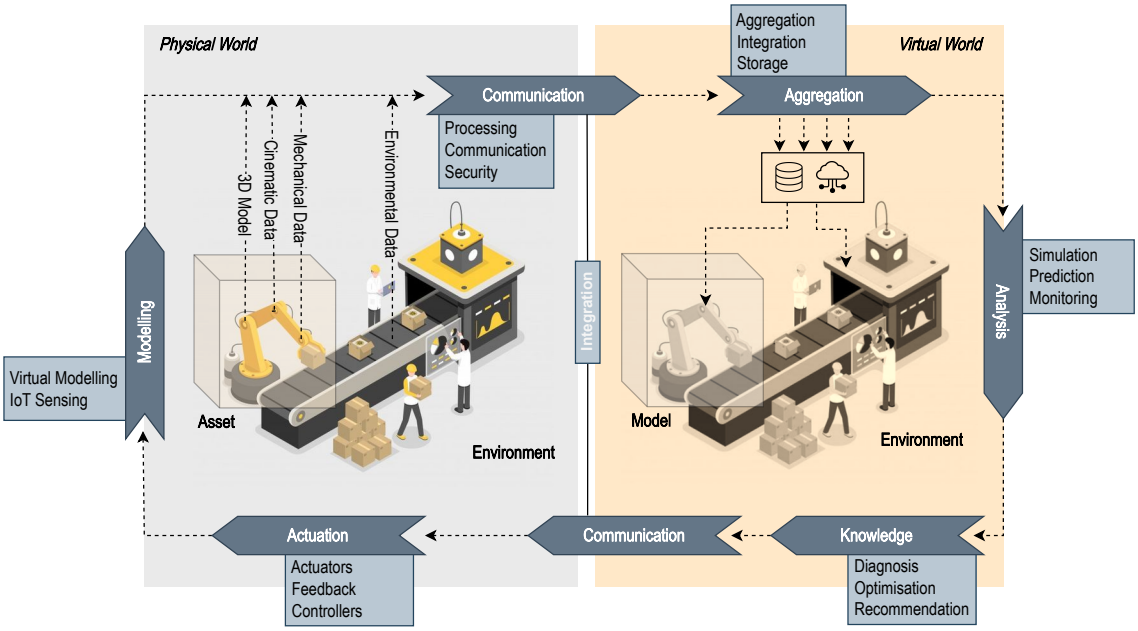

Figure 2.5: Conceptual architecture for Digital Twin (Based on Parrott and Lane (2017)).

time, or faster, depending on the application of the Digital Twin. After collecting data from the physical world, it enters the integration phase known as *Aggregation*. This phase consists of three steps: aggregation, integration, and storage. The processed data is then saved into a repository (e.g., SQLite or MongoDB), and the virtual model is updated accordingly based on the collected data. Following this phase is the *Analysis* phase, where the stored data is used to perform data monitoring and analysis to enable simulations and predictions. Based on the results obtained in the previous phase, valuable knowledge and insights can be extracted in the next phase, known as the *Knowledge* phase. This knowledge can diagnose problems, optimise processes, and provide recommendations. In the *Actuation* phase, this knowledge is transmitted to the physical asset, which can use actuators and controllers to implement the recommended feedback. Ideally, the ultimate Digital Twin implementation does not require human intervention, but this requires the human already has much confidence in the system. Therefore, in this last phase, one of the main functionalities of the Digital Twin is to provide decision support (Parrott and Lane, 2017).

Digital Twins provide multiple advantages, as highlighted in several studies (Mashaly, 2021; Rasheed et al., 2020; Oracle, 2017). One of the key benefits is the ability to perform *Real-time Monitoring, Control and Data Acquisition*, where continuous updates between physical and digital systems enable dynamic monitoring of changes. This facilitates informed decision-making and remote system control. The emphasis on *Business Continuity through Remote Access* underscores the system's accessibility, fostering collaboration among team members and enabling autonomy for enhanced productivity. Additionally, *Increased Efficiency* is achieved through rigorous scenario testing, offering a platform to optimise solutions, increase autonomy levels, and perform tasks remotely. The integration of AI and ML enables *Predictive Maintenance and Optimised Scheduling*, leveraging real-time data analysis to predict machine states and schedule maintenance

effectively.

Furthermore, *Enhanced Risk Assessment* is accomplished by conducting virtual scenario testing without impacting the physical system, contributing to proactive risk management. Lastly, the Digital Twin plays a key role in *Supporting Decision-Making*, providing a unified platform for real-time analytics and data-driven decision-making, ultimately enhancing organisational efficiency (Mashaly, 2021; Rasheed et al., 2020; VanDerHorn and Mahadevan, 2021). Deploying Digital Twins offers a range of benefits described above that promise to transform industries by improving operations, minimising risks and enabling data-driven decisions.

Based on the dataset attained earlier from the bibliometric search on the Digital Twin topic (with 16.068 publications), an analysis of the co-occurrence network generated using author keywords can provide a comprehensive understanding of the landscape surrounding the Digital Twin. The VOSviewer co-occurrence network of the author keywords is illustrated in Figure 2.6.

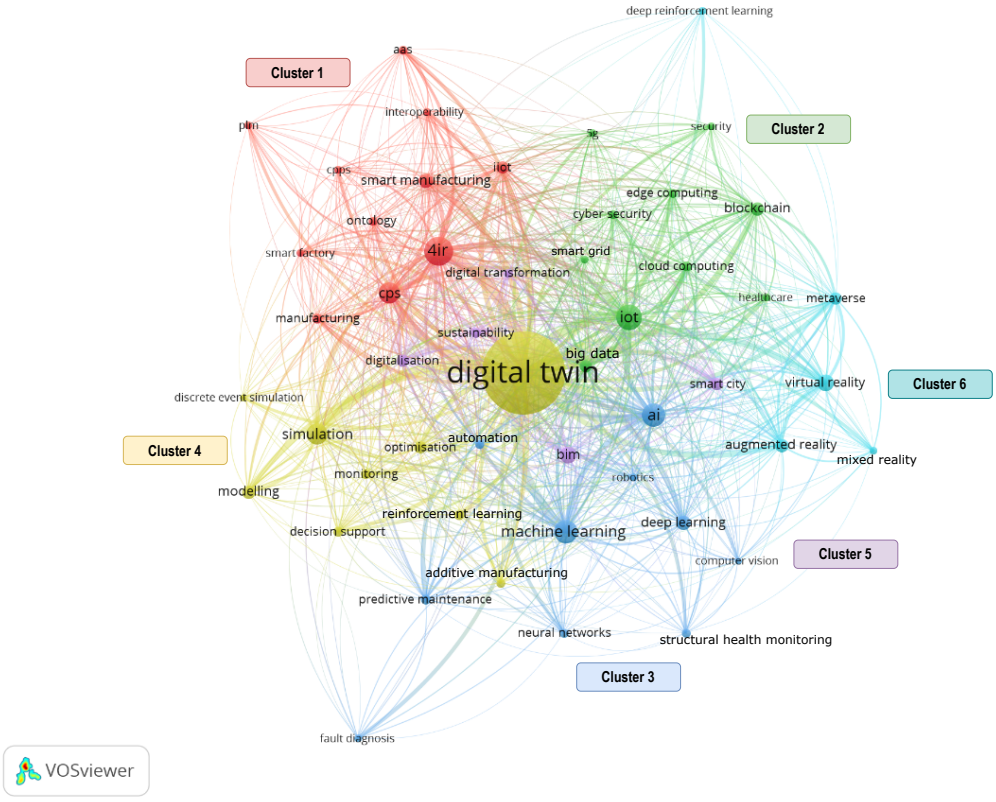

Figure 2.6: Author keywords co-occurrence network map of the literature on Digital Twin (Timeframe 2010-2023; *n*= 27.847 keywords; threshold of 70 occurrences per keyword, display 50 keywords), with six clusters.

Within the network are presented six distinct clusters that represent a variety of technologies, application domains, and functionalities. In Cluster 4 (Yellow), the Digital Twin concept holds the central position concerning all the clusters, being the most common term, considering the authors' keywords, in the publications dataset. This particular cluster is focused on the main functionalities

of the Digital Twin technology, which encompasses *optimisation*, *monitoring*, *modelling*, *simulation*, and *decision support*. One of the most prominent application domains for Digital Twin technology is currently within the manufacturing industry (*smart manufacturing*, *smart factory*, and *manufacturing*). Upon more profound analysis of the network, it becomes clear that the technology is inherently linked to several other key concepts, including *4IR*, *CPS*, *simulation*, *IoT*, *AI*, and *ML*. These keywords represent the most significant connections to the Digital Twin concept. While manufacturing remains the primary application domain, other areas, such as *predictive maintenance*, *smart city*, *smart grid*, and *healthcare*, are also experiencing growth.

The main purpose of the Digital Twin of a manufacturing system is to facilitate the decision-making process and to enable decision automation through simulation (Kunath and Winkler, 2018). Considering the research problem identified in the previous chapter, which aims to enhance the effectiveness of decision-making, the central objective of this project is to develop and implement decision-support capabilities within the Digital Twin system. This will enable decision-makers to make informed and accurate decisions based on data-driven and simulation insights.

2.1.3 Decision Support using Digital Twin

The Digital Twin technology is one of the new technologies supporting digital transformation and enabling decision support. Once the Digital Twin is fully integrated with the manufacturing system, it is a central tool for decision support (Kunath and Winkler, 2018; VanDerHorn and Mahadevan, 2021). The Digital Twin for decision support can be employed following three approaches: diagnosis, monitoring, and prognosis. The main functionality to be explored during this work is the decision support based on Digital Twin.

Although it is a well-known and straightforward term *"decision support"* is always very loosely defined, depending on the context. Without any question *"decision support"* is a part of the decision-making processes. In which a *decision* is defined as a choice of one among several alternatives, and the *decision-making* refers to the whole process of making the choice. The term *"decision support"* includes the word *support*, which translates to the action of supporting people in making decisions, concerning mainly human decision-making. Inside the decision support concept lies the general discipline of DSS, defined as interactive computer-based systems with the main purpose of helping decision-makers use data and models to identify and solve problems and make decisions. The DSS comprises various types of information systems for supporting decision-making, such as executive information systems, executive support systems, geographic information systems, experts systems, and RS (Bohanec and Institute, 2001; Liang, 2008).

The role of the DSS has evolved from simply aiding decision-makers with analysis to providing automated intelligent decision support (Liang, 2008). The DSS can be differentiated between *passive*, *active* and *cooperative*. The *passive DSS* can support the decision-making process but cannot make decisions, suggestions or solutions. The *active DSS* can generate decision suggestions or solutions. Lastly, the *cooperative DSS* enables the decision-maker to refine the decision recommendation provided by the system before sending it back to the system. The system will improve and refine the suggestions to the decision-maker, sending it back (Felsberger et al., 2016;

Chaplin et al., 2020). A DSS is necessary for predicting the advantages and disadvantages of different solutions and performing recommendations for the best solution concerning the local and global objectives.

The new dynamic landscape of the 4IR manufacturing requires the performance of informed decisions accurately and in real-time. The complex nature associated with 4IR manufacturing environments of process efficiency, resource allocation, quality assurance, and risk management demands an advanced decision support approach beyond traditional methods. The Digital Twin offers a good solution for decision support in 4IR manufacturing since it is the virtual representation of the real-world manufacturing system constantly evolving based on real-time physical system changes. The real-time synchronisation and data collection empower the decision-makers with knowledge and insights about the system, enabling the monitoring, analysis, prediction and simulation of various aspects of the physical system.

A bibliometric study was performed to understand how the theme of decision support is evolving within the Digital Twin, following the previously established methodology (see Figure 2.2). The search was performed in Scopus from 2010 to 2023 with the following query:

```
TITLE-ABS-KEY{("digital twin") AND (("decision support" OR "decision-
support") OR ("decision-making" OR "decision making"))} AND PUBYEAR
> 2009 AND PUBYEAR < 2024
```

It identified 1.773 publications, resulting in a dataset with 1.630 to be analysed after only considering English-written publications and in the final publication phase. Figure 2.7 illustrates the evolution of the number of publications.

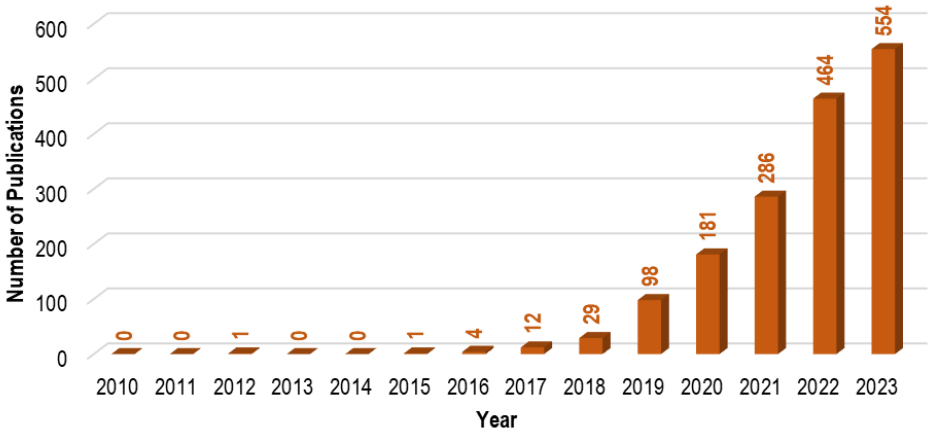

Figure 2.7: Number of publications of Digital Twin and decision support.

The number of publications that mention the two topics has grown in the last few years, representing a growing interest in applying Digital Twin to implement decision support strategies or offer optimised decision-making. Recurring to the VOSviewer, it is possible to assess how the decision support is linked to the Digital Twin, the research trends, and the main application domains of the two concepts. Figure 2.8 presents the author's keywords co-occurrence network.

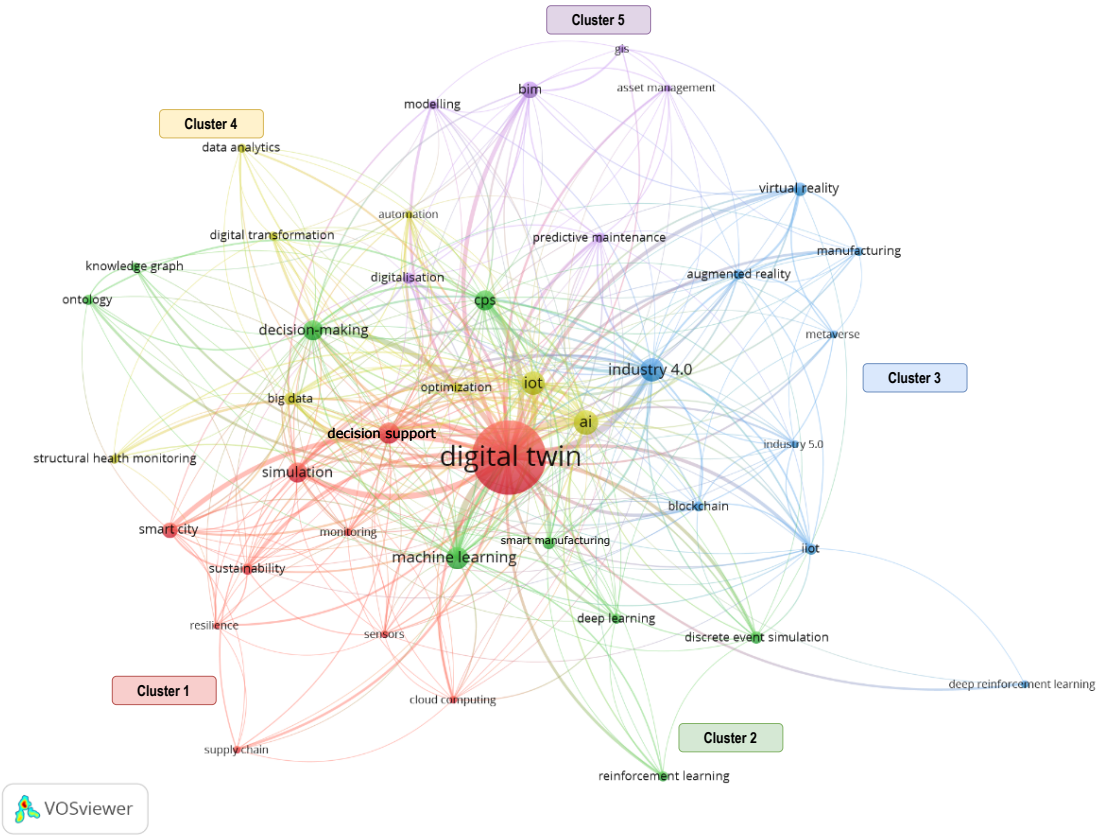

Figure 2.8: Author keywords co-occurrence network map of the publications on Digital Twin and decision support (Time-frame 2010-2023; *n=3880* keywords; threshold of 10 occurrences per keyword, display 42 keywords), with five clusters.

The author's keywords co-occurrence network was divided into five clusters according to the parameters established in the VOSviewer. It was possible to identify, in Cluster 1 (Red), that the *'digital twin'* and *'decision support'* are frequently used together, presenting a strong link between them. Although the concepts do not belong to the same cluster*'digital twin'* and *'decision-making'* also present a strong link. These strong links can indicate that decision support is a key theme for the Digital Twin technology. The integration of DSS with a Digital Twin enables several capabilities, such as monitoring based on real-time data, data-driven and simulation-driven insights, knowledge-based decision support, and predictive capabilities. The confluence of these capabilities results in several advantages, such as increased efficiency, reduced downtime, and improved productivity.

2.2 Recommendation Systems for Decision Support

In the field of decision support, RS contributes to the aiding of decision-makers in selecting relevant options based on their preferences, serving as an intelligent DSS (Malik et al., 2020; Isinkaye et al., 2015).

2.2.1 Formalisation Recommendation Process

A RS can be defined as information filtering and decision support technology that utilises user preferences, behaviour, and item data to provide personalised recommendations. The main goal of this type of system is to help users discover items, products, services, scenarios, or content that are likely to be of interest to them in complex information environments (Isinkaye et al., 2015). The RS evolved as an independent research field in the mid-70s, but it was not only until the mid-90s that the first approaches started to appear (Sharma and Singh, 2016; Pires et al., 2023). The recent increase in the proliferation of RS technology has revealed its power in enhancing system performance (Roy and Dutta, 2022; Gupta and Dave, 2020; Liang, 2008).

The formal definition of the functioning of the recommendation process is presented next, considering that there are two classes of entities that should be referred to as users (u) and items (i). The RS includes the set of all users $(U, u \in U)$, the set of all possible items $(I, i \in I)$ that can be recommended, such as books, movies, gadgets, or simulation scenarios, and a utility function (F) that measures the usefulness of a specific item $i \in I$ to user $u \in U$, i.e., $F : U \times I \rightarrow S$, where S is an order set of recommendations. For an RS, the utility of an item can be defined as the rating, which indicates how a user liked a particular item. The RS will determine for each user $u \in U$ the item $i \in I$ that maximises the user's utility according to Equation 2.1 (Sharma and Mann, 2013; Patel and Patel, 2020).

$$\forall_u \in U, i_u = \arg\max_{i \in I} F(u, i) \tag{2.1}$$

Generally, the rating follows a scale, for example, on a scale from 1 to 5, an item rated with 5 by a user means it is highly liked/preferred, while 1 rating means dislike/unpreferred. An RS can perform the recommendation following two strategies, one when it recommends the items with the highest estimated rating or provides a list with the top-N items (Sharma and Mann, 2013; Patel and Patel, 2020). Figure 2.9 illustrates the general model of the recommendation process.

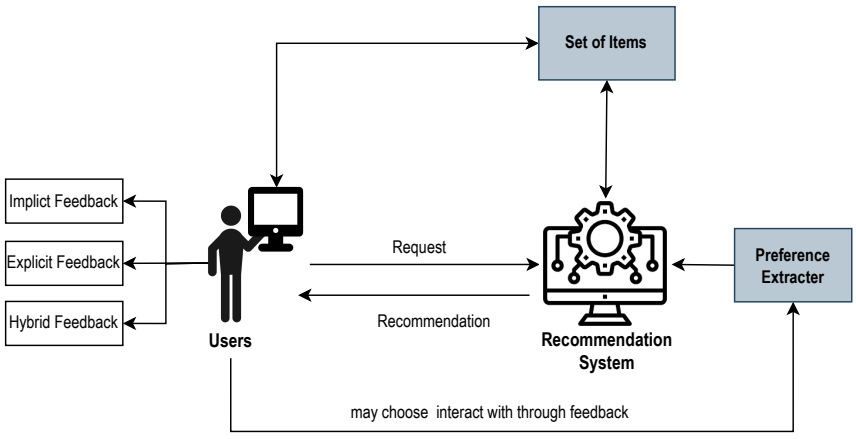

Figure 2.9: Model of the general recommendation process.

The model of the recommendation process in Figure 2.9 is a general model that can be applied

to a broad range of recommendation activities. The user can request recommendations, or the system can suggest recommendations without any request. The user of the system can also interact with the system by providing feedback on their preferences, or the system can require them to do this. Based on the user preferences, the RS can assemble a new set of item recommendations according to the established preferences. The recommendation aids the user in selecting the most appropriate items from the set of items (Sharma and Mann, 2013). In general, an RS collects information from the users, which can be explicit by collecting users' ratings or implicit by monitoring the users' behaviour (e.g., songs heard, websites visited, books read) to get information on the users' preferences for a set of items (e.g., movies, songs, book, websites). The foundation of the RS relies on three types of inputs such as *explicit feedback*, *implicit feedback*, and *hybrid feedback*. The explicit feedback comprises the users' explicit input regarding their preference or interest in the recommendation provided. This input type provides reliability and transparency to the recommendation, promoting accuracy. The implicit feedback is inferred indirectly by observing user behaviour. This type of input requires less effort on the user side but is less accurate. Lastly, hybrid feedback encompasses the combination of explicit and implicit feedback, using the implicit feedback to validate the explicit feedback or providing users with the possibility of giving feedback (Chakraborty et al., 2021; Isinkaye et al., 2015). The RS may use different sources of information to perform the predictions and recommend the items, attempting to balance characteristics such as accuracy, novelty, dispersity, and stability (Bobadilla et al., 2013). The application of RS enables several advantages, such as the performance of recommendations based on the actual user behaviour, based on which the user can make a decision, the personalisation of the recommendations, and the real-time capability. Regarding disadvantages, there is no stop condition when comprehensive information is available, and it is challenging to perform recommendations for changing data and user preferences (Aamir and Bhusry, 2015).

2.2.2 Properties and Requirements for Designing Recommendation Systems

The RS falls in the category of a *cooperative DSS*, which enables the decision-maker to refine the decision recommendations provided by the system (Felsberger et al., 2016). The properties of a RS are the main characteristics a system should have to provide recommendations effectively. Based on Ricci et al. (2015), there is a set of properties that are commonly used by the decision-makers to decide which recommendation approach to select. There are several key properties associated with the RS, such as:

- *User preference or Relevance*, is an essential part of any RS since the system should be aligned with the user's preferences and be able to perform recommendations (Aamir and Bhusry, 2015; Shani and Gunawardana, 2011).

- *Prediction Accuracy*, this property is directly correlated with the decision-maker's choice of using the RS, since these prefer a system capable of generating more accurate recommendations (Aamir and Bhusry, 2015; Menk Dos Santos, 2018).

- *Coverage*, it measures the capability of a system to perform recommendations in terms of the proportion of available items to all potential decision-makers. Usually, a system with low coverage offers a limited decision field to the decision-makers (Weng, 2008; Shani and Gunawardana, 2011; Menk Dos Santos, 2018).

- *Confidence*, this property is related to the system's trust in its recommendations or predictions, for which the system will assign confidence scores to the items. This score can affect the decision-makers' acceptance of the recommendation. The confidence in the predicted property is directly correlated with the amount of data present in the system (Menk Dos Santos, 2018; Shani and Gunawardana, 2011; Aamir and Bhusry, 2015).

- *Trust*, this property differs from the trust in the confidence property. In this case, trust refers to the decision-maker's trust in the system recommendation, which can increase or decrease depending on the recommendations provided by the system. For example, if the system recommends items the decision-maker likes, although the decision-maker gains no new information, it increases his trust in the system (Shani and Gunawardana, 2011; Aamir and Bhusry, 2015; Menk Dos Santos, 2018).

- *Novelty and Serendipity*, these two properties are very close in definition. The system can have *Novelty* in the sense that it can provide novel recommendations, which are recommendations of items similar to the ones already recommended but that the decision-maker did not know about and could not have found himself. In the case of *Serendipity*, is the capability of the system to provide surprising recommendations to the decision-maker (Weng, 2008; Shani and Gunawardana, 2011; Aamir and Bhusry, 2015; Menk Dos Santos, 2018).

- *Diversity*, this property can be defined as the capability of the system to generate distinct recommendations to the ones already provided to the decision-maker. This property can directly affect user satisfaction and other properties, such as accuracy (Shani and Gunawardana, 2011; Menk Dos Santos, 2018).

- *Utility*, it can be defined as the value that either the system or the decision-maker can gain from a generated recommendation, which is highly dependent on the main goal (e.g., more revenue, decrease downtime) of the system for the primary owner (Shani and Gunawardana, 2011; Menk Dos Santos, 2018).

- *Risk and Privacy*, this property is associated with a potential risk that the recommendation provided by the system can have, which can influence the decision-maker's final decision. The privacy property is mainly related to the availability of the preferences and information of the decision-maker used by the RS, which can not be available for a third party (Shani and Gunawardana, 2011; Menk Dos Santos, 2018).

- *Robustness*, in the field of RS, this property can be viewed in three perspectives, one related to the capability of the system to handle fake information, to remain stable, and still be able

to provide appropriate recommendations to the decision-maker, another related with the stability of the system under extreme conditions (e.g., a large number of requests), and another related with the infrastructure of the system (e.g., software and hardware specifications) (Shani and Gunawardana, 2011; Menk Dos Santos, 2018).

- *Scalability*, the RS should be scalable, capable of handling large amounts of decision-makers data, items, and interactions. With the increase of this information, the system should be able to maintain the performance (Shani and Gunawardana, 2011; Menk Dos Santos, 2018).

Additionally, for the development of a RS, it is necessary to consider the following requirements (Bobadilla et al., 2013): the data available in the dataset (e.g., ratings, user information, features and content of items, social relationships), the filtering algorithm (e.g., demographic, content-based, collaborative, social-based, context-aware, trust-based, and hybrid), the model chosen (e.g., memory-based or model-based), the techniques that can be employed (e.g., probabilistic, neural networks, genetic algorithms, fuzzy models, among others), the objective of the recommendation (e.g., predictions of ratings or Top-N recommendations), being also important to consider the sparsity level of the database and the desired scalability, the performance of the system, and the desired quality of the results (e.g., novelty, coverage, and precision). The design of the RS depends highly on the domain to be applied and the main goals of the RS.

2.2.3 Recommendation Approaches for Decision Support

A RS to perform any kind of recommendation is based on approaches that can be used to determine which items might most align with the user preferences and needs. Figure 2.10 illustrates the classification of the recommendation approaches.

Recommendation approaches can be classified into two major groups, the *traditional* and the *social* RS. Traditional RS assumes that users are independent and identically distributed, ignoring any social interactions or connections among them, basing the recommendation and prediction solely on the rating data. This encompasses methods like *Collaborative Filtering (CF)*, *Content-Based Filtering (CBF)*, *Hybrid-Based Filtering (HBF)*, *Demographic Filtering (DF)*, and *Knowledge-Based Approach (KBA)*, which often rely on explicit ratings or preferences to generate personalised recommendations. In the case of the social RS, which first appeared in 1997, it leverages measurable social relationships or networks, combining rating data, trust data and social information to perform recommendations. Considering the traditional RS, there are at least three main approaches: CF, CBF, and HBF. Figure 2.11 illustrates the recommendation procedure for each approach.

The CF approach, introduced in the 1990s, remains a popular traditional approach for recommendation (Goldberg et al., 1992). This domain-independent strategy is based on the assumption that users with similar interests in one area are likely to have similar preferences in other areas as well, allowing for personalised recommendations to be generated by identifying similarities between users or items based on their rating patterns (Sharma and Singh, 2016; Isinkaye et al., 2015).

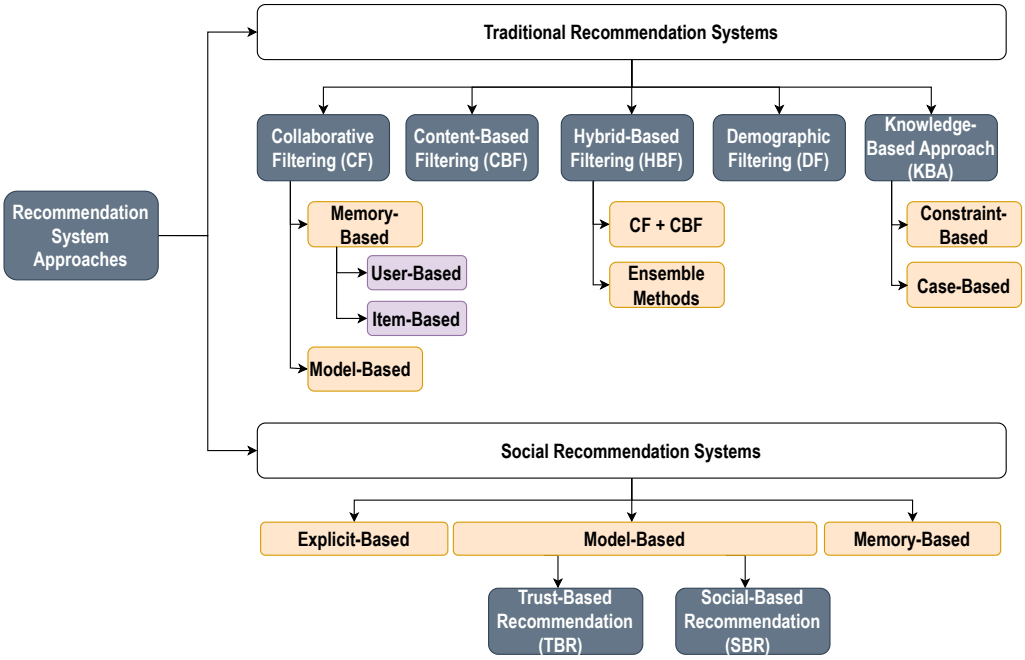

Figure 2.10: Classification of recommendation approaches.

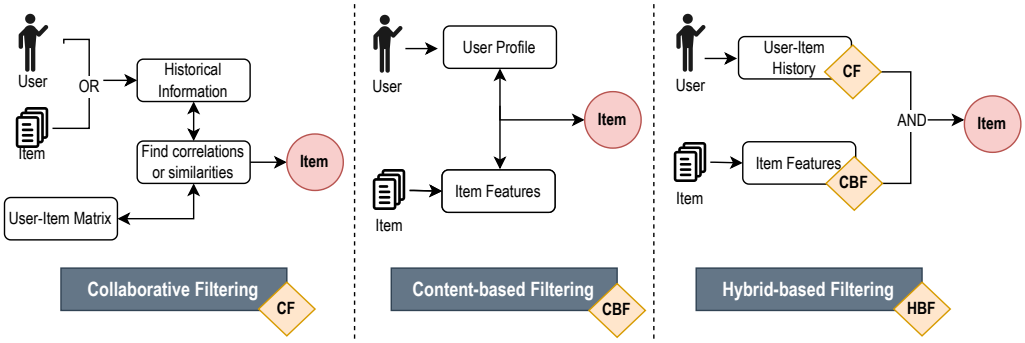

Figure 2.11: Functioning of traditional recommendation approaches.

There are two main categories of CF: *Memory-Based* CF and *Model-Based* CF. Memory-Based CF uses the entire rating matrix to generate recommendations based on user or item similarities. In contrast, Model-Based CF relies on a mathematical model to predict user ratings. Memory-Based CF employs similarity measures such as Pearson Correlation Coefficient (PCC), Cosine Similarity (COS), and user correlation to generate recommendations that are tailored to each user's interests and preferences (Patel and Patel, 2020; Malik et al., 2020; Pires et al., 2023).

The memory-based CF model can be divided further into *User-Based* CF and *Item-Based* CF. The User-Based CF is the model that identifies the users that have similar preferences to the target/active user[2], and the items that were recommended and preferred by the similar users are again recommended to the active user. The recommendation is based on the assumption that users with similar preferences will also prefer similar items. In the case of the Item-Based CF model,

[2]Active/Target user refers to the user currently using the recommendation systems.

they are identified as similar to the ones the active user has already liked. The recommendation is performed under the premise that if the user liked an item in the past, it is likely that they will like similar items. The model-based CF is a learning technique based on mathematical models to learn patterns that use user-item matrix information. This approach builds predictive models that can be generalised from the data available and perform recommendations (Patel and Patel, 2020; Malik et al., 2020; Pires et al., 2023). Apart from this classification, the CF approach can also be divided into two disciplines: neighbourhood approach and latent factor models. The neighbourhood approach focuses on using the relationships between items or, in the alternative, between users. In the item-oriented approach, a user's preference towards an item is determined based on the rating of similar items by the same user. The latent factor models transform items and users to the same latent factor space, making them directly comparable (Koren, 2008).

The CBF approach first appeared in 1992 as a domain-dependent recommendation technique and is one of the most basic recommendation models, having been mostly used in early RS (Malik et al., 2020). The method used by CBF to perform recommendations is based on the features or attributes of the items, recommending to the users items that are similar to the ones that the user already evaluated in the past, considering the description of the item and the profile of the user preferences (Sharma and Singh, 2016). The CBF follows two strategies to recommend items to users: the classifier-based and neighbour methods (Weng, 2008). The classifier-based method uses a classifier that decides if the item should be recommended or not, depending on its content. In the second strategy, the items the user has rated are stored, and the constructed network of items is used to uncover the user's interest in a new item (Portugal et al., 2018). Most of the algorithms that are used in this approach are text mining, semantic analysis, Term Frequency-Inverse Document Frequency (TF-IDF), Neural Networks (NN), Naive Bayes, and Support Vector Machine (SVM) (Malik et al., 2020; Isinkaye et al., 2015).

The HBF approach was proposed as an approach to overcome the limitations of the CF and CBF in terms of scalability and sparsity and improve the recommendation performance of the RS. This approach relies on the premise that to perform recommendations, the system's base combines two or more approaches to attain better performance (Sharma and Singh, 2016). The HBF can be implemented in various forms, e.g., implementing collaborative and content-based methods independently and aggregating their predictions, integrating characteristics from a CBF model into a CF model, and building a new consolidated model that incorporates aspects of both CBF and CF (Thorat et al., 2015). In addition to combining traditional recommendation approaches, recently, data mining and ML techniques have been used to build HBF systems, namely NN, Fuzzy Logic, Singular Value Decomposition (SVD), Bayesian techniques, and Reinforcement Learning (RL) (Çano and Morisio, 2017; Lin et al., 2021; Urdaneta-Ponte et al., 2021). The combination, e.g., with RL, presents several advantages, namely, the recommendation strategies can be updated during interactions, the long-term cumulative reward from the users' feedback is maximised, the exploration and exploitation of recommendations are balanced, and the continuous learning capability allows the update of the recommendations according to the changes of the user interests (Lin et al., 2021). The combination of approaches can be performed by different hybridisation

techniques that can be divided into seven types such as weighted hybridisation, switching hybridisation, cascaded hybridisation, mixed hybridisation, feature combination, feature augmentation, and meta-level (Malik et al., 2020; Isinkaye et al., 2015).

Apart from the three main traditional approaches, others have evolved from these, such as *KBA* and *DF*. The KBA RS, also known as expert-based RS, is a domain-specific approach that uses explicit or domain knowledge from the users to produce personalised recommendations. These systems are known for incorporating human knowledge, rules, or ontologies to provide recommendations. Usually, to perform recommendations, knowledge-based recommenders employ three types of knowledge: knowledge about the users, knowledge about the items, and the matching between items and users. The fact that this approach uses domain knowledge to perform recommendations means that this approach does not suffer from ram-up/cold-start and rating sparsity problems (Tarus et al., 2018; Burke, 2013; Adomavicius and Tuzhilin, 2005). The DF uses the user's attributes (e.g., age, gender, area code, education, employment) to make recommendations based on the demographic classes. This system considers the common or similar personal attributes between users to infer that these are likely to have preferences for similar items (Tarus et al., 2018; Weng, 2008).

These social relationships can be trust relations, friendship, memberships, or following relations (Tang et al., 2013; Ma et al., 2008). Based on the social networks, there are three types of social RS, *Explicit-based*, *Model-based*, and *Memory-based*. Explicit-based methods are based on explicit user connections, for example, on social media. The memory-based social RS uses memory-based CF models, oriented to the user as their basic models. In the case of social RS, these follow two steps: first, they obtain the correlated users for the decision-maker and the second step, the ratings are aggregated from the correlated users to obtain missing ratings. The model-based social RS chooses the model-based CF methods as their basic models. Most of the existing social RS in this category are based on matrix factorisation, being the basic idea behind these methods is that user preferences or ratings are similar to or influenced by the users from whom they are socially connected.

Regarding the most successful social RS approach, the Trust-Based Recommendation (TBR) approach can be defined as a collaborative system using the trust concept as a quantifier for user relationships (Ma et al., 2009; Massa and Avesani, 2007; O'Donovan and Smyth, 2005). Considering the decision-making process, trust has become one key factor, especially in highly dynamic and decentralised environments (Selmi et al., 2016). A general trust definition is the belief and commitment of a person towards a recommended action that in the future will lead to a good outcome (Golbeck and Hendler, 2006). The trust concept has evolved throughout time, and it can be divided into two categories, namely *context-specific interpersonal trust*, which is the user trust in another user regarding a specific situation, and *system-impersonal trust*, which describes the user trust over the system itself (Abdul-Raham and Hailes, 1998). This approach performs recommendations by incorporating trust-related information, considering the trustworthiness and reliability of users, items and other entities involved. The trust information can be extracted from social trust networks created by users to generate individual recommendations (Victor et al., 2011; Pires

et al., 2023). Using trust in recommendation approaches can promote the development of new user relationships, increase connectivity, and alleviate challenges such as data sparsity and cold-start (Isinkaye et al., 2015).

Table 2.2 summarises the advantages and disadvantages of the described RS approaches, providing a comparative analysis between all the presented approaches.

Table 2.2: Advantages and disadvantages of each recommendation approach.

	Approach	Advantages	Disadvantages
Traditional	**Collaborative Filtering**	Very easy to implement and understand New data added easily High quality recommendation in social networks Independent from the item content No over-specialisation problem Domain-independent approach	Highly dependent on user ratings Suffers from new-user and new-item cold-start problem Poor performance for sparse data Limited scalability for large datasets Limited recommendation diversity Prone to shilling attacks
	Content-Based Filtering	User independence Transparency on recommendation explanation Good at recommending new items No dependency on historical user-items interactions Recommendation quality increases over time, and user usage	Harder to have feedback from the users Overspecialisation problem Difficult to generate attributes for items Suffers from new-user cold-start knowledge of the field is often necessary
	Hybrid-Based Filtering	Mitigates limitations of CF and CBF Better prediction performance Combines strengths of different approaches Provides diverse and balanced recommendations	Costly implementation Increased implementation complexity Difficult to provide a recommendation explanation Hard to compare recommendation approaches
	Demographic Filtering	Personalisation based on user demographics Provides targeted recommendations for user No historical data and simple to implement	Security and privacy of the user data General and low-quality recommendations Low adaptability to user changes
	Knowledge-Based Approach	Does not have a ramp-up problem User independent Sensitive to preference changes	Complex knowledge engineering Recommendation performance is static Limited scalability and adaptability to new domains
Social	**Trust-Based Recommendation**	Alleviation of data sparsity and cold-start Increase recommendation coverage and predictive accuracy based on the number of users	Limited for the new item cold-start problem Accuracy can decrease depending on the number of connections to the source user

The CF approach presents several advantages, such as being a domain-independent technique that enables the filtering of any item only based on the historical information about a given user preference (Kim et al., 2010). This approach works very well for recommendation environments with large amounts of data. Another key advantage is that recommendations are only based on the user rating. The memory-based CF makes the RS easy to manage due to the ease of adding new

data incrementally. In the case of the model-based CF, the main advantage is the improvement of the prediction performance (Thorat et al., 2015). Despite the popularity of this kind of technique, it presents limitations regarding data sparsity and, cold-start problems and scalability, requiring considerable computational power to make recommendations for big datasets (Thorat et al., 2015; Çano and Morisio, 2017).

The CBF approach presents some advantages relative to the CF approach, such as the ability to make recommendations even with no available ratings, to adjust the recommendations shortly after the change of the user's preferences, and to provide explanations on how the recommendations were generated (Isinkaye et al., 2015; Thorat et al., 2015). This approach does not suffer from new items cold-start since the recommendations are performed based on the items' descriptions and not on their user ratings. On the other hand, this technique requires a detailed description of item features, and it has difficulties performing recommendations when the users vary their preferences quickly. This technique often suffers from the new user cold-start problem since it is challenging to perform the first recommendations accurately. The CBF approach restricts the recommendations since the approach promotes content over specialisation, focusing the recommendations on the preferred content (Thorat et al., 2015).

The combination of two recommendation approaches, in the HBF approach, enables the improvement of the recommendation process's accuracy and efficiency by overcoming the combined techniques' problems such as cold-start, over specialisation and data sparsity (Thorat et al., 2015). Despite the advantages of combining the different approaches, comparing the recommended techniques is complex, the complexity of implementation increases, and the recommendation explanation is problematic.

In the case of the DF approach, this enables the generation of personalised and targeted recommendations based on the decision-maker demographics, and it does not require historical data being very simple to implement. The fact that requires personal information (e.g., age, gender, income, education) about the user to perform recommendations increases the security and privacy risks (Nawara and Kashef, 2020). Although the generated recommendations are personalised, they can also be general and of low quality since the approach has low adaptability to user changes (Weng, 2008).

One of the main advantages of the KBA over other approaches is that it does not suffer from the ramp-up problem (i.e., cold-start and data sparsity problems) since it does not depend on user ratings. This approach is user-independent, as it does not require gathering information about any particular user since the recommendations are based on the requirements established by the decision-maker. However, it requires extracting knowledge by implementing complex engineering methods, which can be time-consuming (Burke, 2013). The performance of the recommendations is static, not providing a dynamic exploration of other possibilities (Sharma and Singh, 2016). The characteristic of the approach being based on the decision-maker's preferences makes the system sensitive to preference change. However, it makes the system prone to have limited scalability and adaptability to other domains.

Regarding the TBR approaches, the combination of similarity and trust between users im-

proves the recommendation accuracy (Isinkaye et al., 2015) and coverage, which means that the system will consider the entire items list in the recommendation process (Jamali and Ester, 2009). This can alleviate the data sparsity and cold-start problems presented in the CF techniques. For example, in (O'Donovan and Smyth, 2005), trust information is incorporated into the recommendation process, demonstrating a positive impact on the recommendation quality. However, TBR is limited by the definition of a social trust network between users and for the new item cold-start problem. The trust between users is also a limitation, decreasing the accuracy depending on the number of connections of the source user used for the trust calculation. There are also several open research challenges involving the trust theme, such as the alleviation of the trust-based cold-start problem, visualisation of the trust-enhanced RS, theoretical foundations for trust-based research, and introduction of distrust in the recommendation process (Victor et al., 2011). The comparison of the state-of-the-art RS approaches for decision support presents several challenges that require further attention in the future. Challenges such as data sparsity, cold-start problems, scalability, reliance on user information, the definition of trust, and the assessment measure used are several research challenges that still affect recommendation approaches.

2.2.4 Key Challenges of Recommendation Systems

Although the RS are widely used to provide personalised recommendations in various fields (e.g., e-commerce, health, and entertainment), these still present significant challenges. The general challenges associated with the traditional and social RS approaches are as follows,

- *Cold-start problem*: This problem is one of the most common research problems in the RS field, and it relates to the lack of insufficient information, metadata and ratings available and the RS not performing optimally, not being able to perform reliable recommendations. Some authors consider that this problem can be divided into three types, *New Community/System*, *New Item*, and *New User* (Bobadilla et al., 2013; Tey et al., 2021), and some authors only consider the division of the problem in two types *New Item* and *New User* (Fayyaz et al., 2020; Papagelis et al., 2005; Sharma and Singh, 2016). The new community category refers to the moment when a new system is launched, and the items and users present do not have historical data from which it is possible to perform reliable recommendations (Bobadilla et al., 2013; Tey et al., 2021). The new item type refers to introducing new items into the RS from which there is relevant content information. However, there is no rating information, making them unlikely to be recommended. Lastly, the new user problem considers the scenario where new users are introduced to the RS and which do not have information about interactions and rating history, not being possible to generate personalised recommendations (Fayyaz et al., 2020; Papagelis et al., 2005; Sharma and Singh, 2016).

- *Data Sparsity*: This problem is the second most common challenge in the RS area, being responsible for the cold-start problem. Considering that most of the datasets used for recommendation are based on a large number of users and items, it is challenging to ensure that the users rate enough items to guarantee the identification of their preferences (Fayyaz

et al., 2020; Thorat et al., 2015). This results in a sparse dataset, which means that a dataset presents insufficient data for identifying similar users or items, negatively impacting the quality of the recommendations. This problem is more prevalent in RS that rely on peer feedback to provide recommendations (Çano and Morisio, 2017).

- *Scalability*: The amount of data being used as input for RS is growing quickly as more users and items are added to the RS, and large-scale applications are developed. To keep the users engaged, the RS needs to respond interactively in less than a second. The main challenge is to design efficient learning algorithms that can handle small and large-scale datasets (Xin, 2015; Çano and Morisio, 2017). These problems have increased significantly with the availability of large amounts of information, leading to computation difficulties by the filtering algorithms (Fayyaz et al., 2020).

- *Diversity*: This problem can be defined as the ability of the system to perform recommendations based on overlapping items instead of differences, exposing the user to a narrow selection of items and overlooking other good possible items. This is a two-sided problem because the accuracy will decrease if the model focuses strictly on enhancing diversity. This can be evaluated by two measures such as *surprisal*, the ability of the RS to generate unpredictable results and *personalisation*, which is the uniqueness of the different users' recommendation lists (Fayyaz et al., 2020). This issue is important as it helps to avoid popularity bias (Çano and Morisio, 2017).

- *Privacy*: This issue in RS relates to user data's collection, storage, and use in generating personalised recommendations. The provided data may contain sensitive information that the users may want to keep private. The privacy mechanisms can be separated into interactive, which refers to allowing users to query about the data and receive data, and non-interactive, in which a polished version of the data can be published and used for the following operations (Xin, 2015).

- *Shilling Attacks*: This problem, in general, can be defined as malicious entities attempting to manipulate the recommendation algorithms by entering fake or biased data. This can be achieved by "profile injection" attacks, which influence the behaviour of the RS by injecting fake profiles into the system to induce fake ratings on items (Guo et al., 2019). Two types of attacks can be inflicted on an RS: the push attack and the nuke attack. The push attack is responsible for increasing the popularity of an item. In the case of the nuke attack, this is responsible for decreasing the popularity of an item (Sharma and Singh, 2016; Chirita et al., 2005).

- *Accuracy*: This challenge is related to the capability of the RS to accurately predict and recommend relevant items to the users based on their feedback and preferences. The main goal of RS is to provide recommendations that are aligned with the users' preferences, requiring the ability to accurately recommend items, which is not always possible due to various factors, including data sparsity, cold-start, data quality, and scalability. With the

improvement of the accuracy of the RS, it is possible to improve the precision, recall, and relevance of recommendations, leading to higher user satisfaction and engagement (Çano and Morisio, 2017; Fayyaz et al., 2020).

- *Structured recommendations*: This challenge relates to the capability of an RS to, instead of predicting individual items, predict preference for sets of items. This includes two challenges: the number of possible sets grows exponentially with the group size. Unlike individual items, selecting the right score function for sets (Xin, 2015) is unclear. This challenge arises due to the need to incorporate additional constraints and consider complex item relationships.

- *Trust*: This challenge relates to the problem of establishing and maintaining trust between the users and the RS, involving the users' perceptions, beliefs, confidence, reliability, fairness, and credibility of the recommendations provided by the system. Some factors can influence trust in the system, such as the transparency and explainability of the algorithms, the trust of the other users of the system considering their reputation and credibility, and social factors (Sorde and Deshmukh, 2015; O'Donovan and Smyth, 2005).

These challenges underscore the complexity of developing a RS that offers valuable, trustworthy, and personalised recommendations while addressing data, privacy, diversity, and decision-maker trust issues. Developing recommendation approaches to address these challenges is critical for the success of the RS. Two of the main challenges in any environment and field of application of the RS are the *cold-start* and *data sparsity* problems (Guo, 2012; Fletcher, 2017; Nanthini and Pradeep Mohan Kumar, 2023). In the manufacturing domain, the emergence of cold-start (e.g., the system encounters new items, new users, or new interactions without sufficient historical data to make accurate recommendations) and data sparsity (e.g., the available information about user preferences, product characteristics, or historical interactions is insufficient) poses significant challenges for decision support.

2.3 Cold-Start and Data Sparsity Problems

Improving the effectiveness and versatility of RSs requires overcoming the previously referred challenges, particularly cold-start and data sparsity problems. The cold-start problem arises when new users or items with no interaction history are introduced, making it challenging for traditional recommendation algorithms to provide relevant suggestions. Addressing these challenges is critical to ensure that RS can adapt to users' evolving preferences and content. Additionally, data sparsity is a prevalent problem, especially in domains with limited user-item interactions. Finding effective ways to handle sparse data is essential to maintain the system's ability to provide accurate and diverse recommendations despite limited feedback. Both challenges underline the need for innovative approaches. In order to address these challenges, it is essential to understand the current scientific landscape and the approaches used thus far.

2.3.1 Bibliometric Study of the Cold-Start and Data Sparsity Problems

The performance of a bibliometric study, in general, enables the exploration and analysis of large volumes of scientific data. Enabling the identification of possible research trends, journal performance, collaboration patterns, and identification of more relevant articles of a scientific field. Since the cold-start and data sparsity challenges are fundamental limitations of RS, a bibliometric study can help in the identification of the most relevant and influential publications in the field, as well as the most promising approaches and techniques for addressing these challenges. It can also provide insights into the evolution throughout time and of the research trends and their impact on the development of RS. By analysing the existing literature, it will be possible to understand the current state-of-the-art better, identify gaps in the knowledge, and propose new research directions. The bibliometric study follows the methodology established in Figure 2.2, and the following search query:

```
TITLE-ABS-KEY((("cold-start" OR "cold start") AND ("data sparsity" OR
"spars*") AND ("recommendation system" OR "recommender system")) AND
PUBYEAR > 1999 AND PUBYEAR < 2024)
```

Based on the performed query in Scopus, 1.430 publications were identified, supposedly focusing their research on the cold-start and data sparsity problems for RS. Considering only the English-written publications and those in the final publication stage, the final dataset has 1.352 publications. In Figure 2.12 is illustrated the evolution of publication types and numbers over the timespan specified in the search query based on this dataset of 1.352 documents.

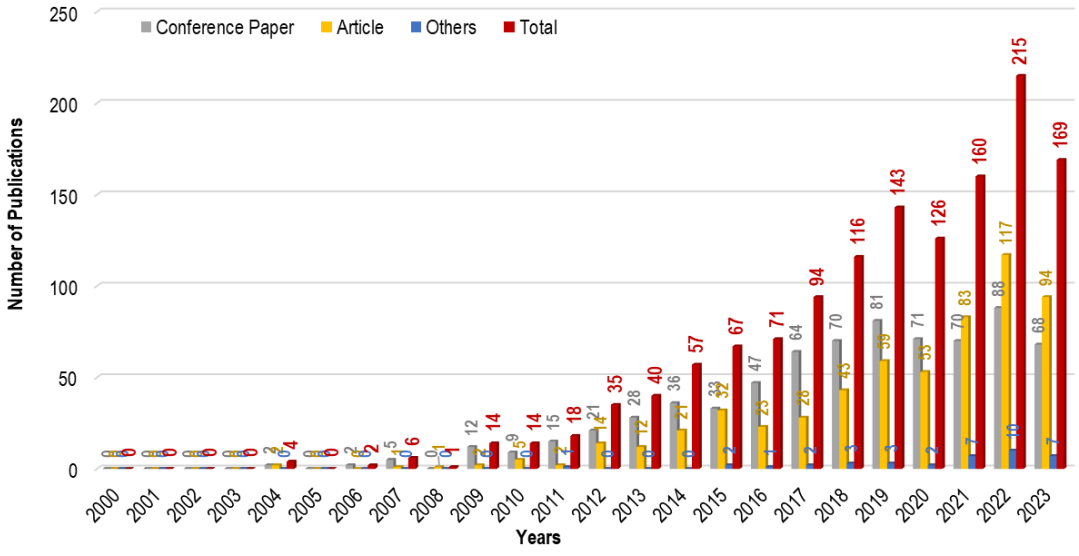

Figure 2.12: Number of publications of the cold-start and data sparsity problems within recommendation systems by publication type.

The number of publications on the topic became more prominent after 2009, showing a higher number regarding conference papers, which was only surpassed in 2021 by journal publications. This can have two possible justifications, the first being a direct consequence of the COVID-19

pandemic since this led to a shift in the publication strategies of most research groups, and the second is directly linked to the increase of the maturity of the research to be carried out on these topics. Since 2018, the number of publications on these challenges has stabilised. However, in the last year, the number of publications significantly increased, demonstrating that the interest in mitigating these problems is still relevant for the research community.

In order to assess the patterns, trends and relationships within the research domain of the cold-start and data sparsity challenges, an author keywords co-occurrence network was generated in the VOSviewer software and presented in Figure 2.13.

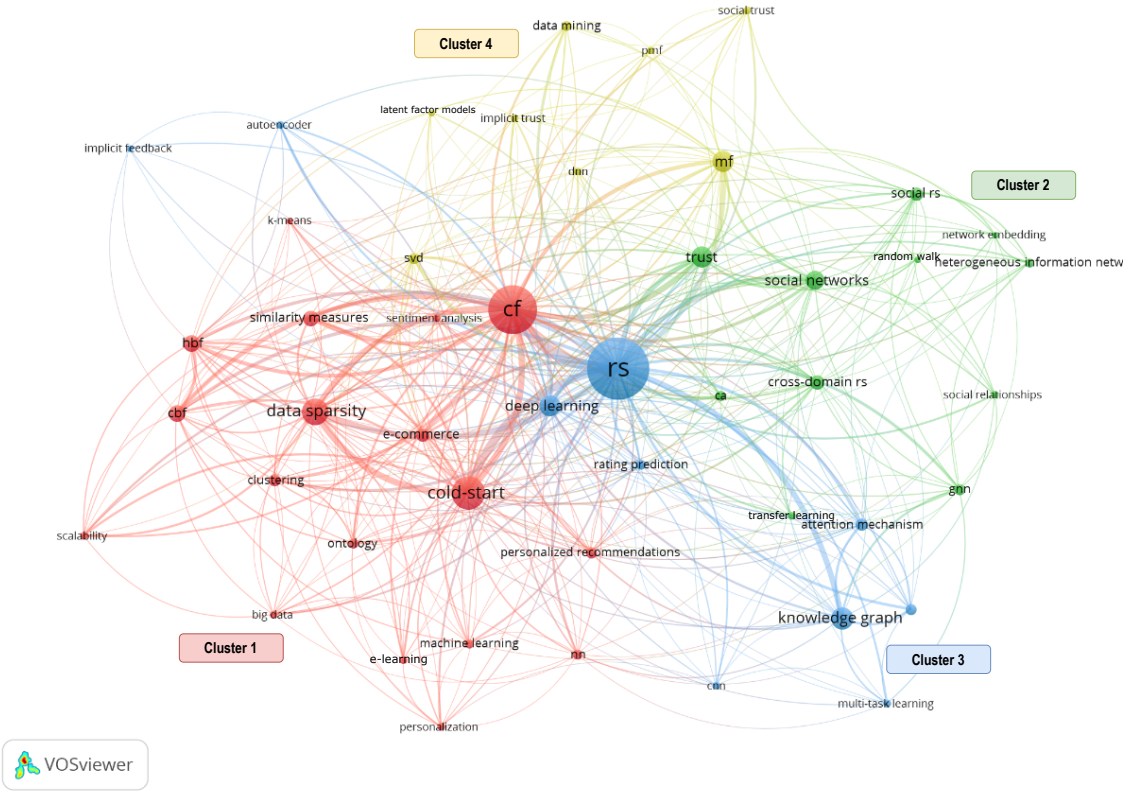

Figure 2.13: Authors keywords co-occurrence network of the literature on cold-start and data sparsity problems in RS (Time-frame 2000-2023; *n*= 2044 keywords; threshold of 10 occurrences per keyword, display 47 keywords), with four clusters.

In this co-occurrence network, it was possible to identify four clusters, and the main topics identified were *RS*, *CF*, *'cold-start'*, and *'data sparsity'*. This makes sense because these topics are the main challenges of the RS field, and one of the main approaches used is CF. In *Cluster 1 (Red)* are presented the main research topics *Cold-Start* and *Data Sparsity*, since these are commonly approached together since they are dependent on each other. Alongside these challenges, there is also another recurring problem in RS, which is *'scalability'*, are also presented two of the proposed RS approaches to mitigate these challenges, the *'HBF'* and the *'CBF'*. This cluster also presents the most used mitigation techniques, the *'similarity measures'* and *'machine learning'*,

and presents one of the major application domains of RS the *'e-commerce'*. In *Cluster 2 (Green)* are presented mitigation techniques more related to the *'social RS'*, including *'trust'*, and *'social networks'*. *Cluster 3 (Blue)* represents the group of keywords showing a trend of implementing AI-based algorithms within the RS as *'deep learning'*, *'Convolutional Neural Networks (CNN)'*, *'autoencoder'*, *'Graph Convolutional Networks (GCN)'*, and *'multi-task learning'*. This cluster also has two growing recommendation approaches: *cross-domain* and *knowledge graph*. Despite being one of the most widely used approaches, CF is ineffective in environments with frequent entry of new users and high levels of data sparsity. In this network are also presented algorithms, methods and frameworks that can be used in this field as *'k-means'*, *'SVD'*, *'Context-Aware (CA)'*, *'data mining'*, and *'ontology'*. There are noticeable patterns in the network as the application of AI-based algorithms, the use of similarity measures and trust relations/metrics to mitigate the cold-start and data sparsity challenges.

2.3.2 Approaches to Handle Cold-Start and Data Sparsity

Based on a high-level assessment of the dataset of 1.352 publications retrieved from the Scopus database, it was possible to identify several approaches to mitigate the cold-start and data sparsity challenges.

All the literature reviews, surveys, and overviews (70 publications), since the focus was to find experimental approaches proposed to the cold-start and data sparsity, so these publications were removed from the initial dataset. The initial timespan was shortened to have the more recent state-of-the-art, keeping only the years from 2013 to 2023 (94 publications). In order to assess the most mature research and approaches in the state-of-the-art, the journal papers were selected, resulting in 548 publications to analyse. In order to narrow down the search even further, publications with the terms "cold-start" or/and "data sparsity" in the title and keywords were selected, proceeding later to the validation of the abstract and the document in full, this last step was performed to ensure that the papers selected focus on the two topics at hand. It is important to note that the publications selected during the bibliometric analysis do not distinguish the articles that only mention the topic from those that address it. Table 2.3 presents the identified approaches within the dataset, and others considered baseline methods that mitigate one or two of these challenges. (**Symbol Caption:** if the challenge is addressed arises a $\checkmark$, if the enabling method is used to mitigate the challenge arises a $\triangle$, lastly to identify the application domain is used a $\bowtie$).

It is assumed that in this dataset, most publications are proposals and validation of approaches to mitigate these challenges. Considering the selection criteria and the entire reading of the documents, 32 articles were selected to determine the current state-of-the-art in cold-start and data sparsity challenges. Most documents focus on the cold-start problem, and only 9 focus on the two problems simultaneously. Although they mention in the abstract that they will address both problems, they only present results for one of them in the experimental part. The general classification of the approaches that are mostly used for the mitigation of these challenges is HBF and TBR, followed by CF, and recently emerging is the CA.

Table 2.3: Characterisation of recommendation approaches focusing on mitigating cold-start and data sparsity challenges.

			Challenges		Enabling Meth.			Domain			
Reference	**Approach**	**Name**	**Cold-Start**	**Data Sparsity**	**Trust**	**Intelligence**	**Similarity**	**E-commerce**	**Entertainment**	**Manufacturing**	**Service**
Massa and Avesani (2007)	TBR	MoleTrust	✓	✓	△			⋈			
Ma et al. (2008)	CF	SoRec		✓	△				⋈		
Ahn (2008)	CF	–	✓				△		⋈		
Koren (2008)	CF	SVD++	✓	✓		△			⋈		
Jamali and Ester (2009)	TBR	TrustWalker	✓		△		△	⋈			
Ma et al. (2009)	TBR	RSTE		✓	△			⋈			
Jamali and Ester (2010)	TBR	SocialMF	✓		△			⋈	⋈		
Ma et al. (2011)	TBR	SoReg		✓	△		△	⋈			
Zhang et al. (2013)	CF	–	✓	✓			△			⋈	⋈
Marung et al. (2014)	HBF	–		✓		△		⋈			
Hwang and Jun (2014)	HBF	–	✓			△			⋈		
Guo et al. (2014a)	TBR	Merge	✓	✓	△		△	⋈	⋈		
Zhang et al. (2014)	HBF	BiFu	✓			△	△		⋈		
Liu et al. (2014)	CF	NHSM	✓				△	⋈	⋈		
Guo et al. (2015)	TBR	TrustSVD	✓		△	△		⋈	⋈		
Ji and Shen (2015)	HBF	TKR	✓			△			⋈		
Zhang et al. (2015)	HBF	DualDS	✓			△			⋈		
Moradi et al. (2015)	CF	DGCTARS	✓	✓	△	△	△	⋈			
Barjasteh et al. (2016)	HBF	DecRec	✓				△		⋈		
Chen et al. (2017)	HBF	GeoMF	✓	✓		△	△				⋈
Yang et al. (2017)	TBR	TrustMF	✓		△				⋈		
Yang et al. (2017)	TBR	TrustPMF	✓		△				⋈		
Sun et al. (2018)	HBF	MFUIpT	✓			△			⋈		
Mohamed et al. (2019)	HBF	–	✓	✓		△	△		⋈		
Zhou et al. (2019)	CF	Inverse_CF_Rec	✓				△				⋈
Rupasingha and Paik (2019)	TBR	–		✓	△	△	△				⋈
Zhang et al. (2020)	CF	CRCF	✓	✓			△		⋈		
Natarajan et al. (2020)	CF	RS-LOD	✓				△		⋈		
Gharahighehi et al. (2022)	HBF	PULCO	✓			△			⋈		
Sejwal and Abulaish (2022)	HBF	RecTEC	✓				△	⋈			
Panteli and Boutsinas (2023)	HBF	–	✓			△			⋈		
Rodpysh et al. (2023)	CA	CSSVD	✓	✓		△	△		⋈		

The cold-start problem has been addressed in the literature employing different mitigation strategies as a new heuristic similarity measure, Proximity-Impact-Popularity (PIP) (Ahn, 2008), an improved PIP measure as Proximity-Significance-Singularity (PSS) (Liu et al., 2014); a random walk method combining trust-based and item-based recommendations, conjugating the ratings and the similarity between items using PCC (Jamali and Ester, 2009); using social networks among users based on trust propagation in the matrix factorisation approach (Jamali and Ester, 2010); using supervised learning algorithms as random forest regression, random forest classification

and elastic net (Hwang and Jun, 2014); employing bi-clustering and fusion techniques along with user/item similarity (PCC similarity measure) (Zhang et al., 2014); incorporating both explicit and implicit influence of trusted users on the SVD++ algorithm (Guo et al., 2015); integrate content-based information about users and items into a neighbourhood approach (Ji and Shen, 2015); exploring the correlations between users and items through a dual regularisation (Zhang et al., 2015); using decoupling mechanisms exploiting similarity information among user/items (COS) (Barjasteh et al., 2016); fusing ratings and trust data into a matrix factorisation model, and a probabilistic interpretation to determine truster/trustee, and accurately infer interest patterns of users (Yang et al., 2017); a deep network model extracting and fusing information from different sources (Sun et al., 2018); an intelligent method comprising opposite users and possible friends using PCC similarity measure to determine these (Zhou et al., 2019); using semantic features of items or users from the linked open data using a similarity measure (Natarajan et al., 2020); use a two step positive unlabelled learning method, using semi-supervised learning and a multi-target regressor (Gharahighehi et al., 2022); using user rating data, topic embedding, and contextual information and integrating them into a user-based CF approach using user similarity measures (COS) (Sejwal and Abulaish, 2022); and applying an approach of both clustering and association rule mining, extracting discriminant frequent patterns (Panteli and Boutsinas, 2023).

The mitigation of the data sparsity problem has also been addressed in the literature by employing social RS based on a social network graph with probabilistic matrix factorisation specifying how much a user trusts another user (Ma et al., 2008); consider trust relationships in social networks for recommending based on the preferences and tastes of the trusted friends (Ma et al., 2009); incorporate user's social network information as a regularisation term to constrain the matrix factorisation using also the knowledge from the similarities between the users, being used the Vector Space Similarity (VSS) and the PCC (Ma et al., 2011); using AI-based algorithms to perform recommendation based a memetic algorithm with visual clustering method based on genetic algorithm (Marung et al., 2014); and using a ontology-based clustering which uses domain specificity and service similarity, and bases its recommendation on the trust value between users calculated using the PCC (Rupasingha and Paik, 2019).

The conjugation of the two problems has some representation in the state-of-the-art, proposing mitigation measures as trust networks (Massa and Avesani, 2007); introduce techniques based on AI such as, SVD++ (Koren, 2008); combining social networks (preference and tagging relationships) with CF by applying similarity measures, as the PCC for preference and rating similarity (Zhang et al., 2013); incorporate social trust information with CF by merging ratings of trusted and similar neighbours of an active user, combining three parts the trust value, the rating similarity (using PCC), and the social similarity (Jaccard Index), and adding rating confidence through user similarity (Confidence-aware PCC) (Guo et al., 2014a); combining CF with similarity values (PCC similarity measure for users and items) and trust statements (implicit trust), and a novel graph clustering algorithm (Moradi et al., 2015); incorporate geographical information by designing a neighbourhood clustering method, with two similarity neighbourhood regularisation terms using PCC as a similarity measure (Chen et al., 2017); merging explicit and implicit data through

similarity measures (COS), clustering techniques and association rules (Mohamed et al., 2019); a novel neighbourhood reduction before computing the similarity (PCC) and prediction by removing redundant elements (Zhang et al., 2020); and combine similarity measures of user-item (Item-Features PCC (IFPCC) and Demographic PCC (DPCC)) with SVD and contextual information through a similarity criterion (Context-based Performance (CWP)) (Rodpysh et al., 2023).

In terms of the experimental validation, all approaches were evaluated offline (i.e., the assessment of the performance of the approach recurring to existing datasets without real-time user interaction) using different datasets available online (e.g., Epinons, FilmTrust, Douban, CiaoDVD, MovieLens, Yahoo, Flixter, and Yelp), most of the approaches uses at least one dataset. However, some approaches are validated in multiple datasets from the same domain or other domains. The domain that is most commonly used in the validation process is *Entertainment*, which includes movies, music, or book recommendations, followed by the *E-commerce*, which includes the recommendation of products, brands or product reviews, an emerging domain is *Service* recommendations, more specific web-services. A less explored domain in the performance of recommendations is the *Manufacturing* domain, representing a good future research domain. Most of the approaches uses as evaluations metrics the Mean Absolute Error (MAE), Root Mean Square Error (RMSE), *Precision*, *Recall*, *F-Measure*, and *Coverage*.

The identified methods applied to mitigate the cold-start problem are mostly related to the use or improvement of similarity measures used alone or complemented by the integration of AI-based algorithms. Most approaches identified in handling data sparsity use techniques based on social networks with trust relationships, including similarity measures and intelligent algorithms as complements. The initial approaches to handling cold-start and data sparsity challenges simultaneously started with more straightforward methods, focusing on a single enabling method such as trust, similarity or intelligence. The proposed approaches are currently more complex, combining two or three methods to attain better results. In the analysed approaches, specific authors, including Moradi et al. (2015) and Rupasingha and Paik (2019), advocate for a comprehensive strategy that combines three enabling methods (Moradi et al., 2015; Rupasingha and Paik, 2019). These approaches combine trust, similarity, and intelligence to tackle the cold-start problem and data sparsity in RS. Trust-based mechanisms can leverage information from user social trust networks to enhance recommendation accuracy, while similarity measures can identify patterns and relationships within sparse data. Furthermore, incorporating intelligence through advanced AI-algorithms allows valuable insights to be extracted from user interactions and feedback. The synergy of these techniques enables RS to mitigate the challenges associated with the cold-start problem and data sparsity, resulting in more robust and efficient systems.

The presented approaches are evaluated against baseline approaches (e.g., userCF, itemCF, MoleTrust, SoRec, SVD++, TrustWalker, RSTE, SocialMF, SoReg, TrustSVD, TrustMF), being able to conclude by the performance results that the combination of several methods enables more efficient RS. Therefore, it was possible to identify a gap in this field of research, which presents several unexplored possibilities of a combination of different approaches inside of the general enabling methods (i.e., trust, similarity and intelligence) that can surpass the existing approaches.

2.4 Enabling Methods for Cold-Start and Data Sparsity

RS requires addressing the challenges of cold-start and data sparsity to provide personalised and accurate recommendations. As discussed in the previous section, researchers have explored various methods to overcome these challenges, including trust, similarity, and intelligence. As the field continues to evolve, integrating trust, similarity, and intelligence remains a promising area for further research. This section will delve into each of these methods in detail.

In the world of RS, tackling the challenges of the cold-start and data sparsity problems has become crucial for providing accurate and personalised user recommendations. Researchers have explored various methods to overcome these challenges, such as trust, similarity, and intelligence. Trust-based mechanisms help enhance recommendations' accuracy by leveraging the relationships and preferences established among users. Employing similarity measures, such as PCC, COS or PIP, helps recognise patterns and relationships within sparse data. Additionally, integrating intelligence, often through advanced AI-based algorithms, empowers RS to extract valuable insights from user interactions and incorporate explicit and implicit feedback. Combining these enabler methods can mitigate the cold-start problem and address the inherent sparsity in data, resulting in more robust and efficient RS. As the field advances, the fusion of trust, similarity, and intelligence continues to represent a promising avenue for further innovation in RS research. In this section, each one of the enablers' methods is going to be explored.

2.4.1 Trust in Recommendation Systems

In the dynamic landscape of RS, trust plays a pivotal role in shaping user experiences and enhancing the accuracy of personalised recommendations. Trust is a fundamental element that bridges the gap between users and the vast array of available items, inducing a sense of reliability and confidence in the recommendation process. By incorporating trust mechanisms into the classical RS, users can identify their individual preferences, evaluate the reliability and credibility of the information provided by other users or the RS itself, and it has the potential to improve the overall performance of the RS (Gupta and Nagpal, 2015). This introduction sets the stage for an exploration of the trust concept in RS, delving into its definitions, the importance it holds for users and systems alike, and the diverse models and approaches that leverage trust to navigate challenges such as the cold-start problem and data sparsity.

Trust has become a key enabling method in decision-making, especially in highly dynamic and decentralised environments with uncertain data (Selmi et al., 2016). Trust within RS helps to deal with the challenges as cold-start decision-makers and data sparsity (Jamali and Ester, 2009; Guo, 2012; Jha et al., 2023). There are several problems when using trust: it is a general and complex concept, it has different meanings for each person, it is context and time-dependent, and it lacks coherence among researchers (Jha et al., 2023).

2.4.1.1 Trust Definitions

According to the Oxford Reference Dictionary, *trust* is defined as *"the firm belief in the reliability or truth or strength of an entity"*. In the RS context, the definition of this concept started around the year 2000 with Abdul-Rahman and Hailes (2000) proposing the use of *direct trust*, which represented the direct trust relationship (e.g., trustworthy, untrustworthy) between two agents (i.e., users). In 2004, Massa and Avesani (2004) proposed a new concept involving the web of trust (i.e., the representation of trusted users about ratings and opinions on items), represented through a trust network of users and trust statements. O'Donovan and Smyth (2005) described trust as a partner's reliability in providing accurate recommendations in the past. The work from Golbeck and Hendler (2006) takes advantage of *explicit trust* ratings based on the premise that *"trust in a person is a commitment to an action based on a belief that the future actions of that person will lead to a good outcome"*. There more recent trust definitions more related to belief, faith and correlation between preference Yuan et al. (2010), for example, considers trust as a *"measure of willingness to believe in a user based on its competence and behaviour within a specific context at a given time"*. Victor et al. (2011) proposed that the recommendations performed by TBR systems are based on *trust networks* based on the following trust definition, being *"the local belief of one user in the usefulness of recommendations provided by another user"*. One definition adopted in the field of RS is the one proposed by Guo (2013), where *"trust is defined as one's belief towards the ability of others in providing valuable ratings"*. Lastly, the authors of Gupta and Nagpal (2015) defined trust as *"one's faith towards others in providing accurate recommendations"*.

According to Josang et al. (2007), there are two common definitions for the trust concept entitled *reliability trust* and *decision trust*. In the case of the reliability trust, this is defined as *"the subjective probability by which an individual (A) expects that another individual (B) perform a given action on which its welfare depends"*. The decision trust defines *trust* as a wider view of the concept, where the *"trust is the extent to which one party is willing to depend on something or somebody in a given situation with the feeling of security, even though negative consequences are possible"*. It was also possible to classify trust as *explicit trust* and *implicit trust*. Explicit trust denotes the trust values explicitly indicated by users, while implicit trust is the trust value inferred from some evidence, such as feature similarity of users or email exchange among two users. Explicit trust can also be divided into two types *direct trust* and *indirect trust*. Direct trust is the trust value explicitly indicated by users. In the case of indirect trust, this is inferred from direct trust using transitivity of trust (Jamali and Ester, 2009; Gupta and Nagpal, 2015).

The trust concept can be approached from two perspectives: a *context-specific interpersonal trust* and *system and impersonal trust*. Context-specific interpersonal trust relates to the relationships between decision-makers, where a decision-maker has to trust another relating to one specific situation but not necessarily to another. In the case of the system and impersonal trust, this describes the decision-makers trust in a RS (O'Donovan and Smyth, 2005). The trust relationships can be divided into two types, *objective* and *subjective*. The objective trust can be calculated based on the similarity of opinion of the decision-makers, including rating or preference similarity. The

subjective trust is determined based on familiarity among the decision makers (Josang et al., 2007; Guo et al., 2019).

2.4.1.2 Trust Properties

The characterisation of the trust concept involves the definition of its properties. The following properties are derived from a study that identified the trust properties in the context of social networks, which can also be used in the field of computation (Golbeck, 2005):

- *Asymmetry:* trust is asymmetric, meaning that trust may not be identical in both directions in a two-person relationship. For example, user A may trust user B, but the inverse may not be valid (Golbeck, 2005; Golbeck and Hendler, 2006; Guo et al., 2014b; Selmi et al., 2016).

- *Dynamic:* trust value can increase or decrease with new experiences. In the case of a good experience of user A with user B, the trust value will increase (Golbeck, 2005; Guo et al., 2014b).

- *Context Specific:* trust is a concept closely related to one person's opinion about a specific area. For example, user A trusts user B in chemistry but does not trust user B in AI (Golbeck, 2005; Guo et al., 2014b; Selmi et al., 2016).

- *Propagation:* the trust value can be derived from the trust of a set of users. If user A trusts user B, and user B trusts user C, user A will have some trust in user C (Golbeck, 2005; Victor et al., 2011).

- *Aggregation:* the trust value can be calculated by combining the trust scores of different paths between users (Golbeck, 2005; Victor et al., 2011).

- *Transitivity:* trust can be transitive, meaning that it can pass to outside a specific domain, or non-transitive, it does not allow the trust to go outside the set domain (Golbeck, 2005; Golbeck and Hendler, 2006; Yuan et al., 2010; Guo et al., 2014b; Selmi et al., 2016).

Defining a new trust model involves considering the trust properties and the criteria such as *Trust Relationship*, *Trust Note*, *Trust Value*, *Trust Properties*, and *Trust Measure*. For defining a trust model, there is at least one type of trust relationship, which can be classified as *Local*, *Global* or *Collective*. The *Local* relationships are determined based on the decision maker's ratings or preferences to infer the relationship with the other users of the system. In *Global* relationships, reputation is the base of the decision-maker relationships. Lastly, in the *Collective* relationships, the decision-maker considers the opinion of third-party users about other users to form relationships. The trust note criteria verify the trust between two users, determining if it is *Explicit* or *Implicit*. In the case of *Explicit* trust, this is directly established by the users, and the *Implicit* trust can be inferred based on the users' history. In terms of the trust value, this can assume two classifications, *Binary* (e.g., 0 or 1) or a *Gradual* number (e.g., any real value in [0, 1]) belonging to a

continuous interval. Considering the criteria of trust properties, this is related to three main properties analysed in the models, such as *Propagation*, *Aggregation* and *Contextualisation*. The trust *Propagation* can be obtained by predicting the trust value between two users in a trust network path. The trust *Aggregation* value is obtained by combining several trust scores from different paths. Finally, the trust *Contextualisation* is the trust value between the users, which is strongly related to the context (e.g., health, industry). Lastly, the trust measure parameter identifies the base measure used to predict the trust scores (e.g., similarity, distance) (Haydar, 2014).

2.4.1.3 Trust Metrics

One of the main challenges when employing trust is determining its value, for this are used *trust metrics*, in which the main goal is to predict, given a particular user, trust in unknown users based on the complete trust network. These metrics can be divided into two categories: *local* and *global*. The local trust metrics consider the users' personal and subjective views and predict different trust values for every user. In the case of global trust metrics, it predicts a global value that approximates how the community perceives a particular user. Regarding computational power, local trust metrics are more expensive since they have to perform calculations for every user in the network.

In contrast, the global metrics are computed once for all the community (Massa and Avesani, 2004, 2009). Several authors developed *trust metrics* that propose calculating trust, mainly based on the knowledge that users whose ratings are close to or similar to each other tend to be trustworthy. The authors of Papagelis et al. (2005) based its trust metric on the similarity measure computed by PCC presented in Equation 2.12, where $PCC(u,v)$ is the similarity between users u and v, and trust is assigned as similarity, i.e., $Trust_{u,v} = PCC(u,v)$. Another similarity approach to trust is the definition of a threshold of similarity proposed by Yuan et al. (2010), which accounts for the similarity value and the number of co-rated items; if these pass the threshold, the user is trustworthy.

Other authors like (Elisa et al., 2009) and (Guo et al., 2013) also propose a trust metric based on PCC. Lathia et al. (2008) proposed a trust metric based on users who provide ratings apart from the ones who do not provide opinions. Trust is defined as the average of provided values over all the rated items according to the following Equation 2.2.

$$Trust_{u,v} = \frac{1}{|I_{u,v}|} \sum_{i \in I_{u,v}} (1 - \frac{|r_{u,i} - r_{v,i}|}{r_{max}}) \tag{2.2}$$

where $|I_{u,v}|$[3] is the set of items commonly rated by users u and v, $r_{u,i}$ and $r_{v,i}$ are the ratings given by user u and user v to item i, and r_{max} is the maximum rating scale predefined by a RS. Other authors, such as Hwang and Chen (2007), propose a trust metric based on Resnick's prediction based on averaging the prediction error on co-rated items, Shambour and Lu (2012) adopted the same strategy and compute trust based on the Mean Square Distance (MSD). The O'Donovan and Smyth (2005) propose two kinds of trust based on the notion of correctness, profile-level and

[3] $|I_{u,v}|$ refers to the cardinality of a set, the number of elements in a mathematical set.

item-level trust, and Pitsilis and Marshall (2004) adopted a subjective logic to derive trust based on uncertainty and disbelief.

The authors Chen et al. (2021) propose a trust metric (Equation 2.3) based on the social trust network to alleviate the data sparsity problem and to improve the recommendation accuracy, calculating trust between users that considers direct trust $Dtrust_{u,v}$ and indirect trust $Itrust_{u,v}$.

$$Trust_{u,v} = \begin{cases} Dtrust_{u,v}, Dtrust_{u,v} \neq 0, \\ Itrust_{u,v}, Dtrust_{u,v} = 0, Itrust_{u,v} \neq 0, \\ 0, Dtrust_{u,v} = 0, Itrust_{u,v} = 0 \end{cases} \tag{2.3}$$

where the direct trust is calculated based on trust weight, and the indirect trust is calculated based on trust transfer mechanisms between users. Other authors that define the trust calculation based on direct and indirect trust are Xiao (2009) and Zhang et al. (2018).

Selecting the right trust metric for a recommendation engine involves careful consideration of several factors. These include the specific context in which the engine will operate, the system's unique characteristics, its users' behaviour, and the goals that the engine is designed to achieve (Pal et al., 2021). Each element is critical in determining the most appropriate trust metric.

In RS, the cold-start and data sparsity, as explored in the previous sections, are challenges that can affect the system's accuracy. However, it is possible to address these two problems by implementing trust mechanisms. These mechanisms can leverage information from trusted sources, incorporate trust information about the user or item, and serve as auxiliary information for systems without explicit user-item interactions. The embedded trust propagation property enables the extraction of knowledge based on the propagation of trust through the social network. Lastly, by applying these mechanisms, it is possible to incorporate qualitative data about users and their relationships (Guo et al., 2014b; Gupta and Dave, 2020; Sheibani et al., 2023). Several authors propose applications for alleviating and mitigating cold-start and data sparsity (Massa and Avesani, 2007; Ma et al., 2008; Jamali and Ester, 2009; Ma et al., 2009; Jamali and Ester, 2010; Ma et al., 2011; Guo et al., 2014b, 2015; Moradi et al., 2015; Yang et al., 2017; Rupasingha and Paik, 2019; Shi et al., 2020; Sheibani et al., 2023). All of these authors attest to the improvements in their approaches, mitigating the cold-start and the data sparsity and improving recommendation accuracy through implementing trust mechanisms.

2.4.2 Intelligence in Recommendation Systems

The integration of AI in RS has transformed how users explore and engage with content, products, and services. RSs are designed to aid users in navigating extensive information spaces by forecasting and proposing items that match their interests. AI-driven methods have brought about a new age of intelligent and flexible systems, building on traditional approaches like CF and CBF.

The integration of AI with RS focuses on the personalisation of customer experience by analysing the user preferences and behaviour (Soori et al., 2023). Several AI-based techniques are employed in RS, enabling abilities such as learning, reasoning, planning, knowledge creation,

natural learning processing, perception and data manipulation. The main techniques being used are Deep Neural Networks, Transfer Learning, Active Learning, RL, Fuzzy Techniques, Evolutionary Algorithms, Natural Language Processing, and Computer Vision (Gabrani et al., 2017; Zhang et al., 2021).

Considering the *Neural Networks*, these are rarely applied in RS since the recommendation task relates to ranking items rather than classification. However, the increasing data availability prompted the employment of deep learning-based RS. Different types of deep neural networks can be applied in RS such as multi-layer perceptron, autoencoder, CNN, Recurrent Neural Networks (RNN), Generative Adversarial Networks (GAN), and Graph Neural Networks (GNN). In the case of transfer learning techniques, they can extend recommendation requests from a single domain to multiple domains. This enables the correlation of information across all domains. Active learning techniques in RS are used to select the most representative items and deliver them to the users to rate them. Many active learning strategies, such as rating impact analysis and bootstrapping, are used with RS. The nature of using RS is an interactive process between the user and the system with a series of states and actions, which is very similar to RL. The RL-RS aim to maximise the engagement and satisfaction of users in the long term. Deep RS is widely used to transform the recommendation process into a sequential task (Yinggang and Xiangrong, 2022). The fuzzy techniques effectively deal with information uncertainty problems since item features and user behaviours are usually subjective. The evolutionary algorithms combine the outputs of multiple recommendation algorithms used as multi-objective optimisation problems. The natural language processing methods enable the extraction of information to complement the rating matrix. Lastly, combining RS with computer vision has allowed the recommendation of image-based systems (Zhang et al., 2021).

According to Gabrani et al. (2017); Zhang et al. (2021), the AI, particularly computational intelligence and ML methods and algorithms, have been applied to RS with the main goal of mitigating challenges such as data sparsity and cold-start and improve the recommendation accuracy. Several methods can be used to mitigate the cold-start and data sparsity problems, such as deep learning techniques, RL techniques, clustering techniques, and association rules, among others (Batmaz et al., 2019; Lin et al., 2021; Sobhanam and Mariappan, 2013; Jooa et al., 2016). The deep learning techniques extract features from side information, integrate them into user-item preferences, and reduce dimensions of high-level and sparse features into low-level and denser features (Batmaz et al., 2019). The RL-based recommendation methods have become a new research trend in RS, outperforming the supervised learning methods(Lin et al., 2021). Another emerging trend is the combination of deep learning with RL, which enables greater scalability, applying the recommendation approach with large state and action spaces (Afsar et al., 2022). The clustering technique is used for grouping items and, based on similarity measures, making predictions for new items, solving the new item problem (Sobhanam and Mariappan, 2013). The association rules specify how one event relates to another (Jooa et al., 2016). Several works already apply these methods and techniques for cold-start and data sparsity, as Wei et al. (2017) proposed a staked denoising autoencoder (SADE), employing deep learning and a CF approach,

to predict the unknown ratings and perform recommendations of cold-start items. Ke et al. (2021) to mitigate the cold-start and data sparsity problems proposed dynamic items RS based on RL, this learns through the reduction of entropy loss error on real-time applications. Zuo et al. (2016) proposed an algorithm based on deep neural networks for handling data sparsity problems based on user-defined tags. Huang et al. (2021) propose a deep-RL and a RNN approach to alleviate the cold-start problem improving the accuracy in the long term. Vizine Pereira and Hruschka (2015) propose a simultaneous co-clustering and learning (SCOAL) algorithm for addressing the cold-start problem. Lastly, Shaw et al. (2010) proposes to use association rules to mitigate the cold-start problem by using these as a source of information to expand the user profile.

The application of AI-based methods in handling cold-start and data sparsity has proven to be indispensable, from leveraging item features and latent factors, learning complex patterns and filling in missing values. Furthermore, implementing these enables the adaption of dynamic user preferences and ensures robust personalisation, making it indispensable for addressing real-world scenarios.

2.4.3 Similarity Measures for Recommendation Systems

In the field of RS, similarity plays a crucial role as it helps to quantify mathematically the degree of similarity between two different items or users. This measure is fundamental in predicting the preferences and patterns of users and recommending relevant items. By comparing the attributes of different items or users, similarity helps identify the ones that are more closely related and likely to be preferred by the user (Isinkaye et al., 2015). Similarity measures are essential in handling the cold-start problem and addressing data sparsity by enabling the system to make informed recommendations for new users or items based on identifying relationships with existing items or users in the system, even when there is limited information. These measures are often applied in CF approaches to handle the cold-start problem and data sparsity (Ahn, 2008).

The similarity measures can be divided according to the classification *Local* or *Global* similarity measures, which assess the similarity or the relationships between items or users. Local similarity measures focus on the similarity between a specific pair of items or users (e.g., *COS*, *PCC*, Euclidean Distance (ED), and Jaccard Similarity (JD)). Global similarity measures assess the overall similarity of an entire dataset, considering relationships between all the users or items (e.g., clustering, SVD, and matrix factorisation) (Anand and Bharadwaj, 2011). Usually, the global similarity measures are used to support the local similarity measures. Another classification proposed in the literature, but not so often used, is the classification of similarity measures as *traditional* and *heuristic* (Bag et al., 2019). The most common similarity measures applied to CF used are *COS*, *Adjusted Cosine Similarity (ACOS)*, *ED*, *JD*, *MSD*, and *PCC* (Jain and Mahara, 2019; Singh et al., 2020; Rodpysh et al., 2023), which the formulas are defined as follows:

Cosine Similarity technique uses vectors to represent user and item rating information. The cosine between the two vectors representing two users (or two items) indicates a certain similarity value between each other. If the similarity value is close to 1, it indicates a strong correlation between the two variables. If the value is close to 0, it indicates no correlation between the two

entities (Sarwar et al., 2001; Fkih, 2022). Equation 2.4 represents the cosine formula for user similarity.

$$Cosine(u,v) = \frac{\sum_{i \in I_{u,v}} r_{ui} r_{vi}}{\sqrt{\sum_{u \in I_u} r_{ui}^2} \sqrt{\sum_{v \in I_v} r_{vi}^2}} \tag{2.4}$$

where I_u and I_v represent the sets of items rated by users u and v, respectively, and I_{uv} represents the set of items commonly rated by bot u and v. The r_{ui} and r_{vi} are the ratings values on item i given by users u and v, respectively. Equation 2.5 represents the cosine formula for item similarity.

$$Cosine(i,j) = \frac{\sum_{u \in U_{ij}} r_{ui} r_{uj}}{\sqrt{\sum_{u \in U_i} r_{ui}^2} \sqrt{\sum_{u \in U_j} r_{uj}^2}} \tag{2.5}$$

where U_i and U_j represents the sets of users who rated the items i and j, respectively, and U_{ij} represents the set of users who rated both items i and j. The variables r_{ui} and r_{uj} are the ratings values assigned by the same user u on the items i and j, respectively.

Adjusted Cosine Similarity is a type of COS that considers the fact that different users have different rating schemes. Therefore, some users might rate items highly in general, and others might give items lower ratings as a preference, which can be mitigated by subtracting average ratings for each user from each user's rating for the pair of items in question (Fkih, 2022). Equation 2.6 is the formula for the ACOS for two users.

$$ACosine(u,v) = \frac{\sum_{i \in I_{uv}} (r_{ui} - \overline{r_i})(r_{vi} - \overline{r_i})}{\sqrt{\sum_{i \in I_{uv}} (r_{ui} - \overline{r_i})^2} \sqrt{\sum_{i \in I_{uv}} (r_{vi} - \overline{r_i})^2}} \tag{2.6}$$

where $I_{u,v}$ represents the set of items commonly rated by both u and v, and $\overline{r_i}$ represents the average ratings on i. The r_{ui} and r_{vi} represent, respectively, the ratings of user u and v on the item i. Equation 2.7 is the formula for the ACOS for two items.

$$ACosine(i,j) = \frac{\sum_{u \in U_{ij}} (r_{ui} - \overline{r_u})(r_{uj} - \overline{r_u})}{\sqrt{\sum_{u \in U_{ij}} (r_{ui} - \overline{r_u})^2} \sqrt{\sum_{u \in U_{ij}} (r_{uj} - \overline{r_u})^2}} \tag{2.7}$$

where U_{ij} denotes the set of users who rated both items i and j, and $\overline{r_u}$ represents the average ratings by u. The r_{ui} and r_{uj} are the ratings of user u on items i and j, respectively.

Euclidean Distance is the length of a line between the two users (or items) in the Euclidean space. In the case of the user, this is represented in the Cartesian coordinates with respect to the basis of items, and vice versa for the items, and the distance between two users is the absolute value of the numerical difference of their coordinates (Fkih, 2022). Equation 2.8 represents the formula to calculate the ED between two users u and v.

$$ED(u,v) = \sqrt{\sum_{i \in I_{uv}} (r_{u,i} - r_{v,i})^2} \tag{2.8}$$

where I_{uv} represents the set of items commonly rated by both u and v, r_{ui} and r_{vi} represent the

rating of the user u and v, respectively, on item i. Equation 2.9 provides the formula for the ED between two items i and j.

$$ED(i,j) = \sqrt{\sum_{u \in U_{ij}} (r_{u,j} - r_{u,i})^2} \tag{2.9}$$

where U_{ij} denotes the set of users who rated both items i and j, r_{ui} and r_{uj} represents the rating of the user u on items i and j, respectively. The ED has to be normalised to become the Euclidean Similarity (ES), through $ES(u,v) = \frac{1}{1+ED(u,v)}$ and $ES(i,j) = \frac{1}{1+ED(i,j)}$.

Jaccard Similarity is used to measure user similarity when the preference information is binary, i.e., like or do not like an item. Equation 2.10 defines the formula for the calculation of the JD coefficient between two users (Anand and Bharadwaj, 2011; Fkih, 2022).

$$JS(u,v) = \frac{|R_u| \cap |R_v|}{|R_u| \cup |R_v|} \tag{2.10}$$

where R_u and R_v are the set of elements preferred by user u and v, respectively.

Mean Squared Distance between two users u and v is calculated by the ratio of sum square of the difference of ratings on co-rated items and the cardinality of co-rated items (Bag et al., 2019). Equation 2.11 is the formula for calculating MSD.

$$MSD(u,v) = 1 - \frac{\sum_{i \in I_{uv}} (r_{ui} - r_{vi})^2}{|I_{uv}|} \tag{2.11}$$

where r_{ui} and r_{vi} are the rating of the item i given by user u and v, respectively. The I_{uv} indicates the co-rated items of users u and v.

Pearson Correlation Coefficient was proposed by Karl Pearson to measure linear relationships (Fkih, 2022). The value returned by the PCC formula is between -1 and 1, where 1 indicates a strong positive correlation, -1 indicates a strong negative correlation, and 0 indicates no correlation at all (Resnick et al., 1994). Equation 2.12 represents the calculation of similarity between two users u and v.

$$PCC(u,v) = \frac{\sum_{i \in I_{uv}} (r_{ui} - \overline{r_u})(r_{vi} - \overline{r_v}))}{\sqrt{\sum_{i \in I_{uv}} (r_{ui} - \overline{r_u})^2} \sqrt{\sum_{i \in I_{uv}} (r_{vi} - \overline{r_v})^2}} \tag{2.12}$$

where I_{uv} refers to the set of items commonly rated by both users u and v, the $\overline{r_u}$ and $\overline{r_v}$ refers to the average ratings of the users u and v on item i in I_{uv}, respectively. The r_{ui} and r_{vi} are ratings of users u and v on the same item i. Equation 2.13 represents the calculation of similarity between two items i and j.

$$PCC(i,j) = \frac{\sum_{u \in U_{ij}} (r_{ui} - \overline{r_i})(r_{uj} - \overline{r_j})}{\sqrt{\sum_{u \in U_{ij}} (r_{ui} - \overline{r_i})^2} \sqrt{\sum_{u \in U_{ij}} (r_{uj} - \overline{r_j})^2}} \tag{2.13}$$

where U_{ij} refers to the set of users who rated both items i and j, followed by $\overline{r_i}$ and $\overline{r_j}$ refers to the average ratings on i and j in U_{ij}, respectively. The r_{ui} and r_{uj} are ratings of user u on items i and j, respectively. Apart from the traditional PCC similarity measure, there are several variations such

as *Constrained PCC*, *Sigmoid PCC*, and *Weighted PCC* (Jain and Mahara, 2019).

The combination of similarity measures can enhance the overall robustness and effectiveness of the RS since the strengths from one measure can alleviate the weaknesses from the other measure. According to Anand and Bharadwaj (2011); Liu et al. (2014); Hu (2018), the application of the more traditional similarity measures (e.g., PCC, COS) may not always be enough to handle the cold-start and data sparsity problem. However, combining them with trust measures can complement and enhance the performance of recommendations.

2.5 Summary

This chapter presents and discusses the literature surrounding the Digital Twin technology and its application to perform decision support by implementing a RS. It was possible to identify that the most recurring problems in RS are the *cold-start* and the *data sparsity* problems, which can be defined as dealing with new items, users, or situations where there are insufficient historical data to make accurate and personalised recommendations, and the available data is insufficient or incomplete, making it challenging to accurately model user preferences or item interactions, respectively. Independent of the applied recommendation approach, these problems can be more or less prominent but are always present.

The literature study has shown that these problems have been addressed over time, proposing approaches including trust, similarity, and intelligence. Although applying a single measure of these three can improve the attained results, their combination has yet to be widely explored in the state-of-the-art. Only some authors explore this as a new mitigation measure for these challenges, leaving a possible research gap open.

In summary, considering the presented background information, including the overview of the main topics, it was possible to identify the current gaps in the literature, and regarding the performed survey on existing approaches that tackle these two problems, cold-start and data sparsity, it was possible to identify the main enabling methods. Based on this information, a different approach to the problems was developed, hoping it could be a representative improvement over the existing ones.

Chapter 3

SimQL Trust-based Recommendation Model

Based on the assumption that the Digital Twin is a key technology for enabling decision support, which, when performed through RS, allows decision-makers to select relevant options based on their preferences and the knowledge generated by the Digital Twin. RSs have proven to be very efficient in decision support. Despite this, these challenges, such as cold-start and data sparsity, are still present and ready to be solved or mitigated. Trust, similarity, and intelligence are the main methods used in the literature to mitigate these challenges. This thesis proposes a Digital Twin architecture for decision support based on an innovative RS approach, comprising the integration and combination of trust, similarity and intelligence. This approach promises to minimise the effects of the cold-start and data sparsity problems in the performance of recommendations to new users or of new items with low data availability.

This chapter describes the proposed architecture, i.e., the Digital Twin architecture based on a new recommendation approach entitled *SimQL*, to enable decision support based on RS that will mitigate the cold-start and the data sparsity problems through the integration of trust and similarity measures, and a AI-based algorithm with the Digital Twin functionalities. This also presents the formalisation of each layer and the defined recommendation strategies.

The remainder of this chapter is structured as follows:

- Section 3.1: presents a comprehensive overview of the proposed Digital Twin architecture for the decision support based on RS in the manufacturing domain. This section will shed light on the roles of each layer and how they collaborate to provide recommendations for decision support.

- Section 3.2: describes the role, main capabilities, inputs and outputs of the *Simulation Layer*, along with the formalisation of the what-if simulation model and the algorithm of the what-if engine.

- Section 3.3: describes the role, main capabilities, inputs and outputs of the *Decision Support Layer*, along with the formalisation of the recommendation algorithm, in this case, the Q-Learning RL algorithm, the reward calculation and the recommendation module.
- Section 3.4: presents the role, main capabilities, inputs and outputs of the *Human Trust Layer*, describing the proposed cold-stat and data sparsity mitigation measures, trust measures (e.g., user trust in recommendation, and user trust in the system) and similarity measures (e.g., user similarity (PCC), scenario similarity (COS), and user reputation).
- Section 3.5: presents the different recommendation strategies defined for the several recommendation scenarios that are possible to occur, e.g., with historical data, no historical data, a new scenario, or a new decision-maker.
- Section 3.6: concisely summarises the key points addressed in this chapter.

3.1 System Architecture

The proposed architecture for the Digital Twin integrating the RS comprises six layers that interact with each other in order to achieve the system goals, as illustrated in Figure 3.1.

The architecture comprehends two dimensions, the *physical world* and the *virtual world*. It comprises six layers, being the *Physical Layer*, the *Communication Layer*, the *Data Analysis Layer*, the *Simulation Layer*, the *Decision Support Layer*, and the *Human Trust Layer* (Pires et al., 2021a). Each layer has different responsibilities, capabilities, and embedded features to enable the performance of the decision support. The *Physical Layer* represents the physical systems or assets for which the Digital Twin is being employed to provide decision support. Apart from the system or assets, this layer accounts for the control system (e.g., actuators, PLCs, or Manufacturing Execution System (MES)), which is responsible for the implementation of the action identified in the provided recommendations. In this layer, real-time data collection is a crucial aspect aiming to "feed" the virtual model. The *Communication Layer* layer enables the connection, communication and data exchange between the system and assets in *Physical Layer*, in the physical world, and the other layers in the virtual world and vice-versa. The data exchange is based on a data model responsible for organising in a standardised manner the different types of data being exchanged and utilising standard industrial communication protocols (e.g., OPC-UA, ModBus, or EtherNet/IP). Depending on the asset or system that is being considered in the *Physical Layer*, it is possible to have multiple communication protocols working together in collecting and storing data from the different assets or systems. The *Data Analysis Layer* is responsible for the performance of monitoring, prediction, diagnosis, and optimisation, among others. These actions are performed based on AI-algorithms and enable the system assessment for anomalies or system degradation. Based on the system assessment results, this layer can generate triggers for RS to generate recommendations exploring the optimisation of the system in the background. After the RS receives the trigger, from the *Data Analysis Layer* or from the user itself, the *Simulation Layer* comes into play, being

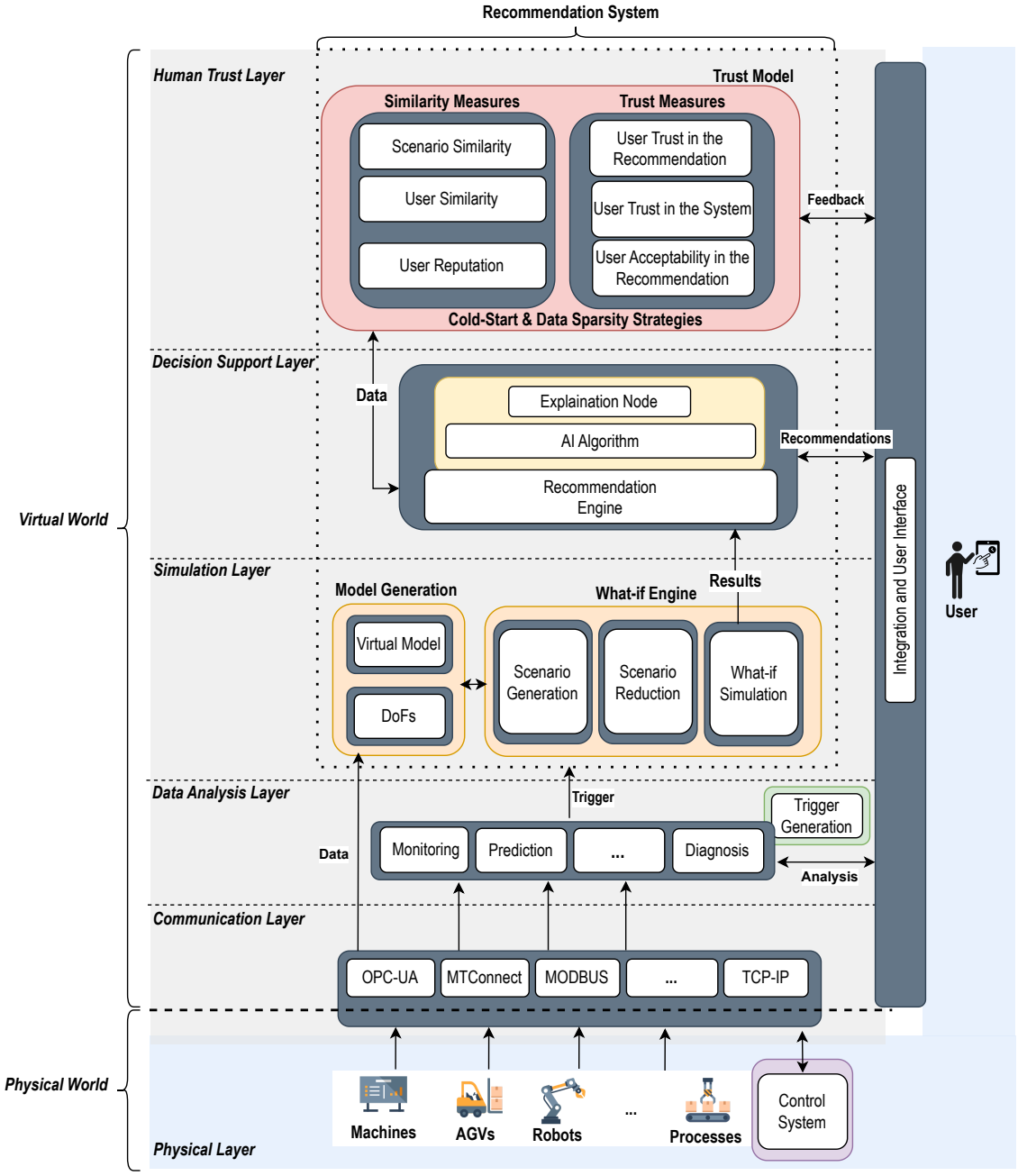

Figure 3.1: Digital Twin architecture for decision support based on recommendation system to mitigate cold-start and data sparsity effects.

responsible for executing what-if simulations of the virtual model of the physical system, aiming to explore new options and configurations, evaluate different approaches or verify a different implementation that could be applied to the physical system. The explorations of these options depend on why the system was triggered. This is performed through the performance of what-if simulations enabled by a what-if engine, responsible for the generation of the what-if scenarios in an automatic manner, considering Discrete Event Simulation (DES) model and the appropriate Degrees of Freedom (DoF). Note that the DoF are the adjustable variables for the problem. Since

the number of what-if scenarios generated may be significant, the application scenario reduction techniques are possible, creating a sub-set of what-if scenarios. From this point, the sub-set of what-if scenarios is simulated using appropriate software packages. The simulation results are sent to the *Decision Support Layer* as a base for the decision support actions.

The *Decision Support Layer* generates the recommendations, aiming to support users in the physical world in the decision-making cycle (strategic and/or operational). The performance of the recommendations is based on a recommendation engine using an AI-algorithm, which takes advantage of the what-if simulation results and from the data from the trust model (i.e., user rating, user similarity, scenario similarity, user trust in the system, user trust in the recommendation, and user acceptability of the recommendation) that comes from the *Human Trust Layer*. This engine can also explain why the given recommendations were provided to the user, increasing the transparency and acceptability of the system. After the user receives its recommendations, this can give feedback about them, which will be received by the *Human Trust Layer*, allowing the update of the established trust model. The *Human Trust Layer* is a trust model comprising mitigation strategies for the cold-start and data sparsity challenges. These strategies comprehend similarity measures (i.e., scenario similarity, user similarity, and user reputation) and trust measures (i.e., user trust in the recommendation, user trust in the system, and user acceptability in the recommendation). Therefore, every time the RS is faced with cold-start or data sparsity recommendation conditions, the recommendation engine requests this layer mitigation measures to improve the generation of recommendations.

The interaction was formalised through Unified Modelling Language (UML) sequence diagrams showing the interaction between the decision-maker and the RS and the interaction between the layers of the system. This type of diagram is commonly used to show the interactive behaviour of a system. Figure 3.2 illustrates a high-level UML sequence diagram of the interactions.

The diagram presents the division of the RS into three layers: the *Simulation Layer*, the *Decision Support Layer*, and the *Human Trust Layer*. Considering the initiation trigger, the decision-maker requests the RS to start the recommendation cycle in the *Simulation Layer*. Therefore, the decision-maker requests a recommendation to the system, setting the DoF and the virtual model. After these parameters are defined, these are sent to the RS *Simulation Layer*. In this layer, the what-if scenarios are generated, scenario reduction techniques can be applied, and the what-if scenarios are simulated. After this, the simulated what-if scenarios are sent to the *Decision Support Layer*, which triggers user data requests to the *Human Trust Layer*. The learning model is applied after the data arrives, and the possible recommendations are calculated. Depending on the recommendation conditions, the algorithm can request the application of cold-start and/or data sparsity measures to the *Human Trust Layer*. Finally, the *Decision Support Layer* send the recommendations to the decision-maker. Lastly, the decision-maker provides the appropriate feedback to the RS, which updates the trust model in the *Human Trust Layer*.

Each layer performs a unique function (e.g., data analysis and what-if simulation) to assist decision-making based on RS. The components of each layer can be formalised using various strategies, including mathematical formalisation. This work focused on three of the six layers of

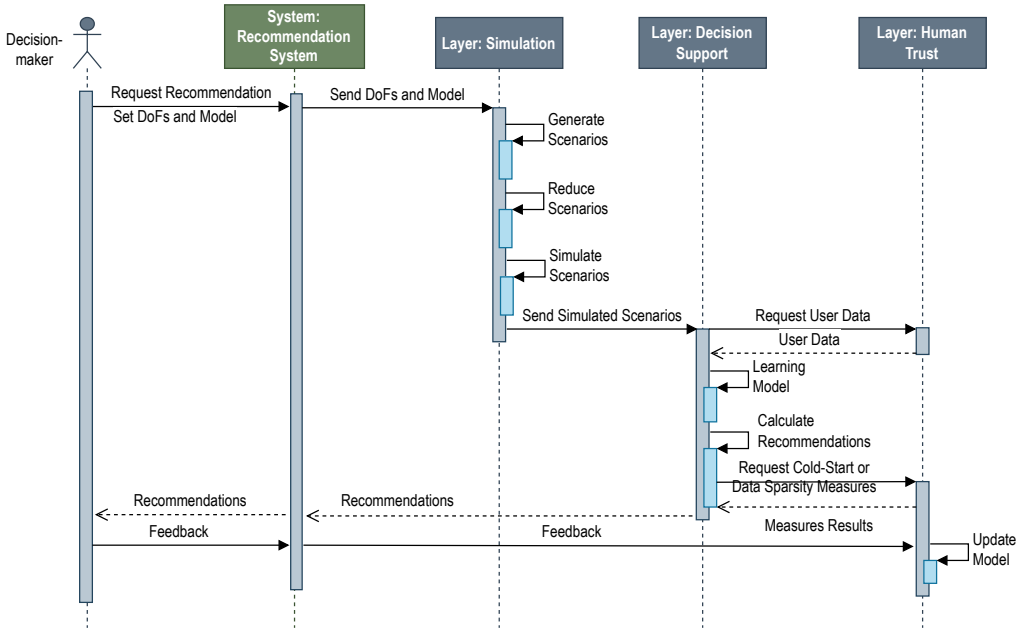

Figure 3.2: High-level UML sequence diagram of the interaction between the decision-maker, the recommendation system and its layers.

this architecture, focusing on the *Simulation Layer*, *Decision Support Layer*, and *Human Trust Layer*. The upcoming sections describe each layer's key capabilities, including its primary role, how it collaborates with other layers, and how they work together to support decision-making. It also presents the formalisation of the different resources of each layer.

3.2 Simulation Layer

The *Simulation Layer* is an important part of the Digital Twin architecture since this enables the Digital Twin-based what-if simulation, allowing for the decision-makers to have a broader knowledge about the physical system's behaviour and possibilities of intervention (Golfarelli and Rizzi, 2009; Pires et al., 2021b). This layer is divided into the what-if simulation model in general and the what-if engine algorithm. The main functional features of the *Simulation Layer* are summarised next:

- *Role*: attending to what-if simulation requests, generating the what-if scenarios, reducing the number of scenarios and performing the actual simulation.
- *Input*: virtual model of the physical system and DoF; *Data Analysis Layer* trigger, or decision-maker trigger, or periodic trigger.
- *Output*: results from the simulation of the what-if scenarios.
- *Main Capabilities*: what-if scenario generation, scenario reduction and what-if simulation.

When applied for decision support, the Digital Twin is frequently combined with simulation methods such as what-if simulation, which is a type of computational model that enables the hypothetical test of different "what-if" scenarios by changing input variables or DoF and observing the resulting outcomes. By performing this type of simulation, it is possible to make informed decisions, assess risks, and identify potential opportunities or challenges.

3.2.1 What-if Simulation Model

The what-if simulation is responsible for running different simulation scenarios of the virtual model of the physical world assets or systems, which can serve as validation, evaluation and verification tools. The integration of what-if simulation within RS promotes timely decision support by enabling the analysis of the simulation results of hundreds of different scenarios, recommending only the most appropriate according to the final objective of the system. Figure 3.3 illustrates the proposed what-if simulation model.

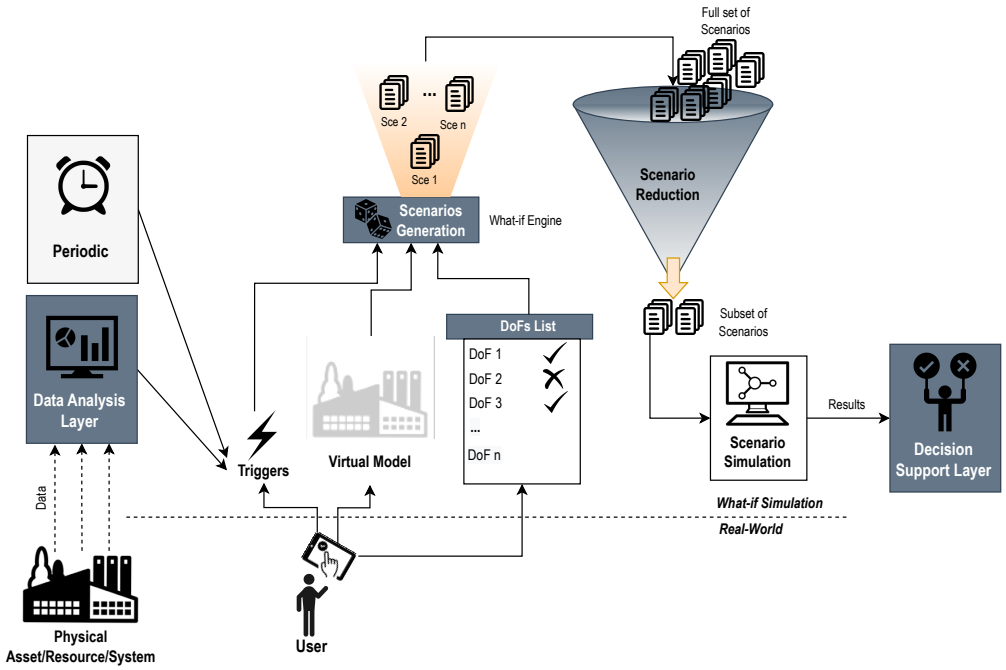

Figure 3.3: What-if simulation model.

The what-if simulation model has three primary triggers: one is the direct trigger from the decision-maker, another is the anomaly detection, failure prediction or performance degradation of the physical system detected by the *Data Analysis Layer* capabilities, and the last one is related to the periodic trigger, that allows the system to explore optimisation scenarios in background operations. The first step for the functioning of this model is the generation of the trigger, followed by the definition of the virtual model (e.g., DES models, 3D models, mathematical models) and the appropriate DoF, which the user defines. These DoF are adjustable variables depending on the problem. The DoF of the physical system can be classified into two categories: independent or dependent. The independent DoF can be defined as independent variables of a physical system

which do not depend on other variables (e.g., shift duration). The dependent DoF is the dependent variable of the physical system (e.g., the system's throughput), whose calculation is dependent on independent variables. These variables are defined by the decision-maker and sent to the what-if engine, where all what-if scenarios are created based on exploring all possibilities combining all the different DoF. The second step of this model is the generation of the what-if scenarios through a what-if engine presented in detail in subsection 3.2.2. After generating the scenarios, the number of scenarios will be reduced if there is already historical data on the problem. This will be performed by applying AI-algorithms. This reduction is based on historical knowledge acquired during similar what-if simulations, including past scenarios' performance scores and users' trust levels in the recommended scenarios. This will result in performing a faster analysis performing simulation only of the most promising scenarios. The sub-set of the most promising scenarios is then simulated using the developed virtual model to simulate the appropriate software (Pires et al., 2021b,a).

3.2.2 What-if Engine Algorithm

As previously stated, the main function of the what-if engine is to generate a collection of what-if scenarios. The proposed what-if engine algorithm, illustrated in algorithm 1, was designed for generating what-if scenarios for a possible physical scenario considering a virtual model, specifically assessing the impact of changes in certain DoF on the model's behaviour.

The algorithm requires several inputs, including a set of virtual models, $\{Model_1, Model_2, ..., Model_n\}$, which will be used to generate different scenarios, and a set of DoF_o^n for each virtual model, $\{DoF_1^n, DoF_2^n, ..., DoF_o^n\}$. For each DoF, a range of values is established, setting a *Minimum*, a *Maximum*, and an increment, x, defined to determine how much the DoF is changed in each iteration. The algorithm starts by iterating over each virtual model, and for each model, it enters into nested loops for each DoF, iterating over the specified ranges. The algorithm verifies whether the defined DoF depend on each other within the nested loops. This dependency, $D_{j,j}^n$, can involve some logic or specific conditions associated with the problem domain. Whether the DoF are dependent or not, the algorithm creates a scenario for the current model and DoF combination, including specific values for the given iteration. After iterating over all the models and all the combinations of DoF, the algorithm outputs a set of scenarios for performing what-if simulations in the chosen simulation software.

The operation of the what-if engine is dependent on the number of virtual models, the number of DoFs, and the dependencies between the DoFs, the calculation of the maximum number of what-if scenarios generated by the what-if engine can be performed by applying Equation 3.2. Considering that, the number of possible DoF for a type of DoF for a specific model, $NDoF_j^n$, is calculated through Equation 3.1. Each DoF can be independent, or it can have dependencies, $D_{j,j}^n$, which affect the final number of scenarios, being necessary to remove this number from the

Algorithm 1: What-if Engine Algorithm

Input:
Define the simulation model: $\{Model_1, Model_2, ..., Model_n\}$
Set of DoF for the scenario: $\{DoF_1, DoF_2, ..., DoF_o\}$
Define range and increment for each DoF: $DoF_1 = (Min, Max, x), DoF_2 = (Min, Max, y)$
Set of what-if scenarios: S_m
Initialise:
S_m initialise empty;
for *Model i=1 to n* **do**
  for DoF_1 *j=Min to Max step x* **do**
    for DoF_2 *k=Min to Max step y* **do**
      if DoF_1 & DoF_2 *are dependent* **then**
        Verify dependency;
        $S = Model_i\{DoF_{(1,j)}, DoF_{(2,k)}\}$, with restrictions;
      end
      else
        $S = Model_i\{DoF_{(1,j)}, DoF_{(2,k)}\}$, without restrictions;
      end
    end
  end
end
Output:
Set of what-if scenarios: $\{S_1 : Model_i\{DoF_{(1,Min)}, DoF_{(2,Min)}\}; S_2 : Model_i\{DoF_{(1,Min+x)}, DoF_{(2,Min)}\}, ..., S_m : Model_i\{DoF_{(1,Max)}, DoF_{(2,Max)}\}\}$

calculation.

$$NDoF_j^n = \frac{Max - Min}{Increment} \tag{3.1}$$

This equation considers the interval of the maximum, Max, and minimum, Min, values that each DoF has to respect, ($DoF_j^n = \{x \in \mathbb{R} : Min \leq x < Max\}$), considering also the defined increment ($Increment$).

$$NS_m = \sum_i^n \left(\left[\prod_j^o NDoF_j^n \right] - D_{j,j}^n \right) = ((NDoF_j^i \times NDoF_j^i) - D_{j,j}^i) + ... + ((NDoF_o^n \times NDoF_o^n) - D_{o,o}^n) \tag{3.2}$$

where NS_m represents the maximum number of scenarios that will be generated by the what-if engine, the i represents the number of virtual models in the interval, $i \in [1, n]$, and j represents the number of DoF selected to be analysed in the specified model according to the interval, $j \in [1, o]$.

In order to show how the what-if engine works, let's consider an example in which only one model is taken into account. The model includes the following DoF: DoF_1, which is the assessment of the number of AGVs; DoF_2, which is the assessment of the best recharging threshold; and DoF_3, which is the assessment of the best resume threshold the AGVs in an assembly line. To determine

the optimal values for this DoF, it is important to consider the number of AGVs, the battery recharge limits (i.e. the percentage of battery for which the AGV will trigger the charge), and the battery resume limits (i.e. the percentage of battery for which the AGV is ready to restart its job). For this particular scenario are going to be considered the following intervals for the DoF: $DoF_1 = \{x \in \mathbb{R} : 1 \leq x < 4\}$, with an increment of 1; $DoF_2 = \{y \in \mathbb{R} : 30 \leq y < 80\}$, with an increment of 10%; and $DoF_3 = \{z \in \mathbb{R} : 40 \leq z < 90\}$, with an increment of 10%.

It is important to note that DoF_2 and DoF_3 have dependencies between each other ($D^n_{2,3}$), meaning that the resume threshold can never be smaller than the recharge threshold for the same scenario. Considering this, the engine generates 60 what-if scenarios instead of the 75 it would generate if the dependencies were ignored.

3.3 Decision Support Layer

The *Decision Support Layer* is a crucial element of the RS, being responsible for the generation of recommendations to the decision-maker. The execution of this layer is based on the results from the *Simulation Layer*, more precisely of the what-if simulation, and the data from the trust model of the *Human Trust Layer*. This layer is divided into two parts: the recommendation environment, which represents the base data for the recommendations and the feedback data, and the recommendation engine, which includes the recommendation algorithm. The main functional features of this layer are presented next:

- *Role*: generate and present recommendations to the decision-maker, integrating what-if simulation results and user trust data.
- *Input*: what-if simulation results, user trust data, and historical data (if available).
- *Output*: recommendations of what-if scenarios and recommendations explanations.
- *Main Capabilities*: generate recommendations based on an AI-algorithm, integrate what-if simulation results and user trust data, and learn with the decision-maker interaction.

The integration of RS in DSS as the proposed system based on the Digital Twin architecture enhances its performance and enables the personalisation of the recommendations, improving the overall user experience and acceptance of the system. Figure 3.4 illustrates the proposed DSS based on an RS using an AI-based recommendation algorithm.

The RS aims to predict the user's interest in the available what-if scenarios and provide the appropriate recommendations. The AI-based recommendation algorithm used in the RS follows the standard terminology of a RL system, which is built based on the *environment*, *learning agent*, and the *reward*. In this case, the recommendation environment is represented by the results from the *Simulation Layer*, the data attained from the *Human Trust Layer*, and the interaction with users regarding its feedback to transform it into a reward. The environment is based on state and action spaces, which allows establishing trust states (i.e., this is a feature representation of the user

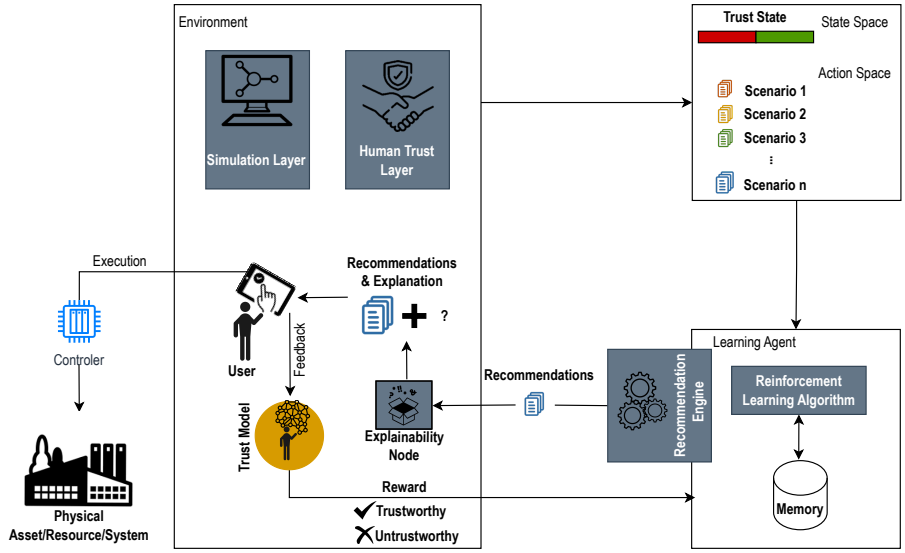

Figure 3.4: Decision support system based on a recommendation system and in an AI-based recommendation algorithm.

trust in a given recommendation (UT_R)) and action space (i.e., represents the scenarios features of the simulation data that can be recommended to the user). The learning agent represents the recommendation algorithm, a RL algorithm (i.e., Q-Learning). The algorithm generates scenario recommendations and expected ratings from the user (E_R) sent to the explainability node. In this node, appropriate explanations on how the recommendation of the scenario was generated are produced and provided to the user jointly with the recommendations. The user expresses trust in the given recommendation (UT_R) as the actual rating of the recommendation (A_R) and states the intention to accept the recommendation to be applied in the physical system, the user acceptability (U_{Acc}).

3.3.1 Recommendation Algorithm

The recommendation algorithm that is used in the *SimQL* trust-based recommendation model is based on the *Q-Learning algorithm* proposed by Sutton and Barto (1998), based on a Q-table and Q-function. The Q-table represents the relationship of Q-values $(Q(s_t,a_t) \equiv Q(s,a))$ between the trust state of the decision-maker $(s_t \equiv s)$ and the actions $(a_t \equiv a)$ represented by all the possible scenarios to be recommended. The Q-learning is a model-free RL algorithm used to learn the optimal action-selection policy for a given environment. The Q-learning algorithm can be seen as Markov Decision Process (MDP), where the states (s) are the states of the environment that belong to a state space $(s \in S)$, defined as all the possible trust states, the actions (a) are the actions taken by the agent belong to an action space $(a \in A)$, defined as the possible scenarios to be recommended, the transition probabilities (P) are given by the environment through a transition function specifying the probability of transitioning to a new state (s') given the current state and action, which can be represented by $P(s'|s,a)$, and the reward $(r_t \equiv r)$ is given by the reward function (R) of the environment, this function assigns a real value r to each state-action pair

$(s,a) : R(s,a)$. The algorithm aims to find the optimal action selection policy that maximises the expected future reward. The Q-learning algorithm updates the Q-function according to the Bellman equation in Equation 3.3.

$$Q(s,a) = R(s,a) + \gamma \times max_{a'} Q(s',a') \tag{3.3}$$

where s' is the next state, a' is the action taken in $s', R(s,a)$ is the reward for taking action a in state s, and γ is the discount factor, where $0 \leq \gamma \leq 1$, determining the importance of future rewards compared to current rewards. Q-learning aims to learn a function $Q(s,a)$, which gives the expected rewards for taking action a in states s and following the optimal policy afterwards. Figure 3.5 illustrates of the MDP of the RL algorithm.

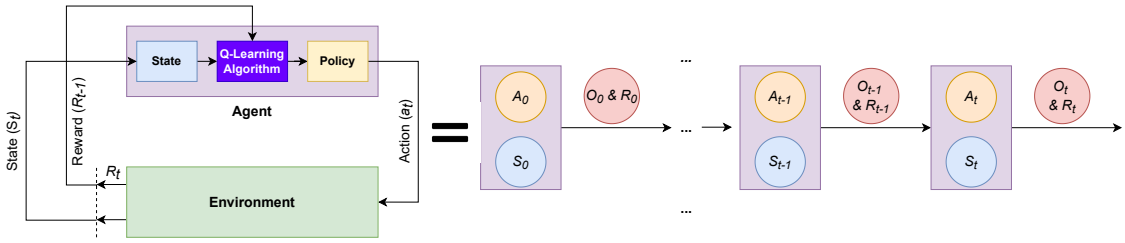

Figure 3.5: Block diagram of the reinforcement learning algorithm Markov Decision Process (Based on Geravanchizadeh and Roushan (2021)).

The RL algorithm consists of an environment that represents the outside world. This agent has, in this case, the Q-learning algorithm receiving states (S_t) and performing actions (A_t) according to an established policy, the actions receive rewards (R_{t-1}) by the users (O_t) (or decision-makers) present in the environment. The agent and the environment interact over a sequence of discrete-time steps. The Q-learning algorithm implementation is based on the algorithm 2 (Pires et al., 2023).

Algorithm 2: Q-Learning Algorithm

$Q(s,a)$ initialise randomly;
Repeat(for each episode)
  state s initialised;
  Repeat(for each step of episode)
    action a chosen from state s using policy derived from Q-Table;
    action a recommended;
    reward r and state s' observed;
    Update
    $Q(s,a) = Q(s,a) + \alpha \times [r + \gamma \times max_a Q(s',a') - Q(s,a)]$;
    $s = s'$;

This considers the actions and states mentioned earlier and the *learning rate* (α), which establishes the learning pace of the algorithm respecting the limits of $0 \leq \alpha \leq 1$, and it also considers the *discount factor* (γ), which represents the importance of future rewards compared to current

rewards, respecting the limits of $0 \leq \gamma \leq 1$. The algorithm also considers the reward (r_t), which defines the good and bad events for the learning agent.

3.3.2 Reward Calculation

In the context of the Q-learning algorithm, the calculation of the reward function is a fundamental aspect that guides the learning process of an agent within a given environment. The reward function is critical for reinforcing positive behaviours and discouraging unfavourable actions, ultimately shaping the agent's decision-making strategy. In this case, the r_t, attained by the Equation 3.4, for each scenario recommendation results from the different measures calculated in the trust model.

$$r_t = W_1 \times UT_R + W_2 \times U_{Acc} + W_3 \times UT_S \tag{3.4}$$

This equation aims to reward trustworthy scenarios and penalise untrustworthy scenarios, calculated by a multi-criteria function where the three components are weighted with W_1, W_2, and W_3 according to the system properties. The first component is the UT_R, which is the trust of the user in the given scenario recommendation, which ranges from $[-V_{min}, V_{max}]$. Note that this scale is symmetrical, which means that absolute values for V_{min} and V_{max} are the same. The second component is the U_{Acc}, which represents user acceptability; the last component is the UT_S, which represents the user's trust in the RS. The reward value is sent to the RL algorithm, updating it for future recommendations. Subsection 3.4.2 presents how the UT_S is calculated.

3.3.3 Recommendation Module

A recommendation module was defined for the proposed model with the recommendations being performed based on a recommendation value, R_{value} calculated by the Equation 3.7, which is based on the Equation 3.5 and Equation 3.6. For both equations, the $Q(s_t, a_t)$ represents the Q-value for the s_t rating state, positive/negative trust rating values ($[-V_{min}, V_{max}]$), and a_t represents the possible recommended scenarios.

$$Trust_P = \frac{|Q(s_t, a_t) + ... + Q(s_{V_{max}}, a_t)|}{Numberofs_t}, if s_t > 0 \tag{3.5}$$

This represents the average value for positive trust states assigned to a specific scenario.

$$Trust_N = \frac{|Q(s_t, a_t) + ... + Q(s_{-V_{min}}, a_t)|}{Numberofs_t}, if s_t \leq 0 \tag{3.6}$$

This represents the average value for negative trust states assigned to a specific scenario.

$$R_{value} = Trust_P - Trust_N \tag{3.7}$$

which intends to penalise the negative trust states, also known as untrustworthy behaviour. The output of the recommendation model is a list of what-if scenarios ordered according to the R_{value} parameters.

3.4 Human Trust Layer

The *Human Trust Layer*, the last layer of the architecture, is responsible for implementing the trust model, which has mitigation strategies for the cold-start and data sparsity problems through similarity measures, such as user similarity $(sim(u,v))$, scenario similarity $(sim(a_t,a_j))$, and user reputation $(rep(u))$, and trust measures, such as user trust in the acceptability of the scenario (U_{Acc}), user trust in the system (UT_S), and user trust in the recommendation (UT_R).

- *Role*: provide cold-start and data sparsity mitigation strategies based on a trust model, and provide the reward to the *Decision Support Layer* recommendation algorithm.
- *Input*: user feedback, user data, scenarios data.
- *Output*: user trust data (i.e., similarity and trust measures) and reward for the recommendation algorithm.
- *Main Capabilities*: calculate scenario similarity, user similarity, user reputation, user trust, and reward calculation.

The proposed trust model uses three trust inputs, *User trust in the given scenario recommendation* (UT_R), which can be measured by the feedback in the form of a rating given by the user; *User trust in the RS itself* (UT_S), which is set initially by the user and continuously updated given the accuracy of the recommendation of the system; and *User social in the work network*, in which each user can give a trust score to another user depending on a set of work-related factors to calculate the user's reputation. Figure 3.6 illustrates the structure and main components of the trust model.

The trust model comprises two main features: the *Similarity Measures* and the *Trust Measures*. The similarity measures mitigate cold-start and data sparsity problems based on user and scenario similarity and social trust networks. The trust measures are responsible for assessing the user's trust in the system and the recommendation, resulting in the reward calculation.

3.4.1 Similarity Measures

The integration of similarity measures in RS enables the identification of patterns, relationships and preferences between users or/and items based on their historical interactions or features. Common similarity metrics such as PCC and COS can be employed to determine the similarity between user preferences or item features. This model employs two similarity measures to determine the similarity between scenarios and between users, using COS and PCC similarity metrics, respectively.

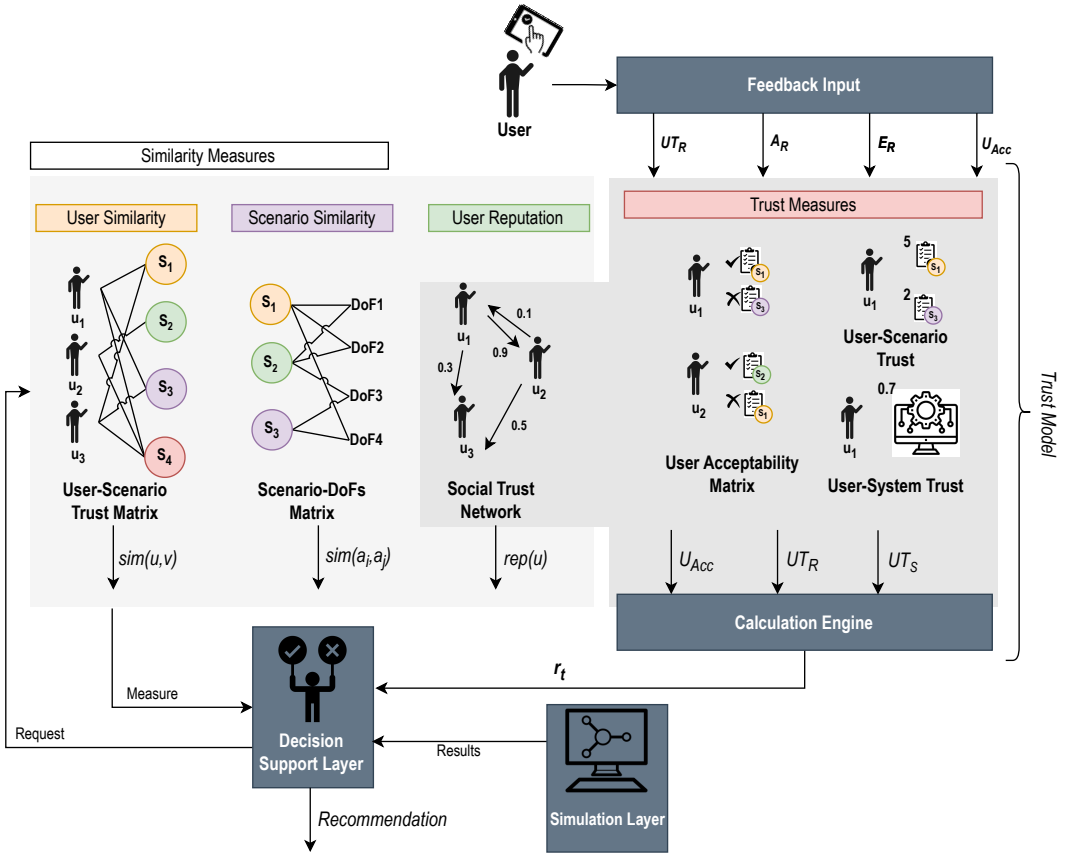

Figure 3.6: Trust model structure and components based on similarity and trust measures.

Scenario Similarity. The scenario similarity measure refers to the degree of how similar two scenarios are in the context of recommendation. This can be used to recommend new scenarios similar to the ones the user preferred in the past. Equation 3.8 presents how the similarity between scenarios is calculated considering the DoF of each scenario. It should be that a_t represents the recommended scenario, and a_j represents the other recommended scenarios. This equation also considers a variable k that represents the unique identifier for each DoF involved in the calculation going from 1 to m.

$$sim(a_t, a_j) = \begin{cases} \frac{\sum_{k=1}^{m} V_{DoF(a_t,k)} \cdot V_{DoF(a_j,k)}}{\sqrt{\sum_{k=1}^{m} V_{DoF(a_t,k)}^2} \cdot \sqrt{\sum_{k=1}^{m} V_{DoF(a_j,k)}^2}}, & N_{DoF(a_t)} = N_{DoF(a_j)} \\ 0, & N_{DoF(a_t)} \neq N_{DoF(a_j)} \end{cases} \tag{3.8}$$

The presented measure uses the COS formula (proposed in Fkih (2022)), which takes into account the different values of the DoF ($V_{DoF(a_t,k)}$) for each scenario. However, the similarity is only calculated if the number of DoF ($N_{DoF(a)}$) is the same for both scenarios and if they are correlated; otherwise, the similarity is zero. Considering that the scenario similarities are calculated, the Q-values used as the starting point will be from the most similar scenario rated by the active user. Figure 3.7 presents the UML activity diagram for implementing the scenario similarity calculations.

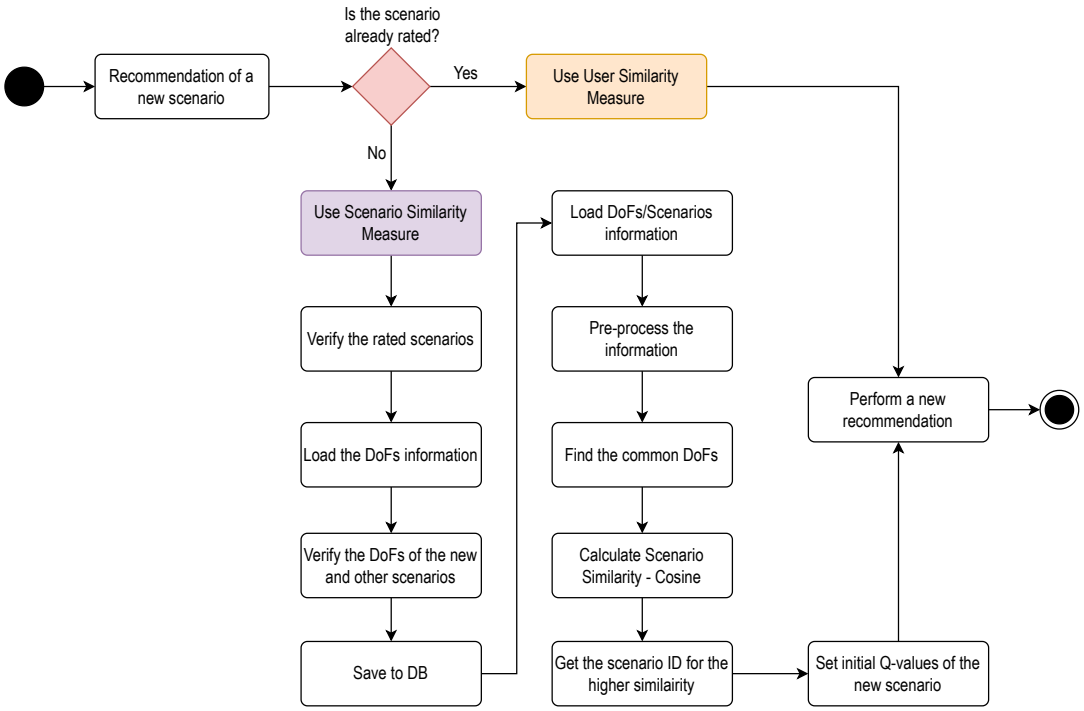

Figure 3.7: Scenario similarity UML activity diagram.

After the system gives the initial recommendations, the user may request the generation of new recommendations for new scenarios, but these newly generated scenarios suffer from the cold-start problem. Therefore, if the new scenario to be recommended has never been rated, its recommendation will depend on its initial Q-values, which can be random or based on the scenario similarity measure. Based on this, the system verifies if the scenario was never rated before and which scenarios the active user rated. Considering the rated scenarios, the list of DoF is loaded, verifying which ones belong to each scenario. After these initial assessments, all the information from the DoF and scenario information is loaded, pre-processed, and sent to the COS function to perform the calculations. This is based on the similarity between the scenarios calculated based on the common DoF. With this, it is possible to identify the scenario with a higher similarity. From this, the Q-values from the identified scenario are set as the initial values of the new scenario.

User Similarity. The similarity measure between the users when rating similar scenarios to support the RS. In this case, the user similarity is calculated according to the PCC metric, assessing the degree of rating similarity between users. Equation 3.9 presents the formula for calculating user similarity (based on the proposed by Fkih (2022)).

$$sim(u,v) = \frac{\sum_t^n (UT_R(u,a_t) - \overline{UT_R(u)}) \cdot (UT_R(v,a_t) - \overline{UT_R(v)})}{\sqrt{\sum_t^n (UT_R(u,a_t) - \overline{UT_R(u)})^2 \cdot \sum_t^n (UT_R(v,a_t) - \overline{UT_R(v)})^2 + C}} \tag{3.9}$$

where $UT_R(u,a_t)$ is the user trust in a given recommendation, a_i, the $\overline{UT_R(u)}$ is the average user trust rating in the given recommendation, and the C term is the shrinking term. Applying the PCC formula minimises the *user bias*, and integrating the shrinking term minimises the *support*

problem. The user bias problem can be defined as some users giving a higher rating than others, favouring some scenarios. The support problem can be defined as the balance between having the similarity calculated based on a few or a large amount of data and normalising the similarity value. Figure 3.8 illustrates the support problem.

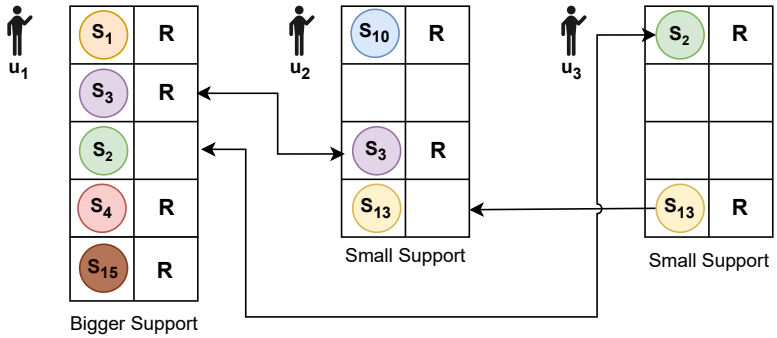

Figure 3.8: Support problem for calculating user similarity.

The support problem is divided into two approaches: the quantity of data available in the user-scenario matrix and the rating of the same scenarios. For example, the support for the calculation of the user similarity for user u_1 is higher than users u_2 and u_3 since these have rated fewer scenarios and only have in common one rated scenario and one scenario that was even rated. This can lead to false results for the user similarity calculation, leading to higher similarity between users than it is. In order to mitigate this problem, the PCC similarity measure enables the addition of a shrinking term, C, that ranges between 1 and 10. This term is added to the PCC similarity to mitigate this problem by reducing the user similarity with small support to the same scale as the higher support.

The user similarity measure is applied when a scenario is recommended, and this was already rated by another user in the system. If this condition is verified, the calculation of user similarity is enabled. Figure 3.9 illustrates the user similarity algorithm's implementation as a UML activity diagram.

The first step of the user similarity calculation involves loading and pre-processing the user-scenario trust matrix information. Subsequently, this information is used to calculate the similarity score, denoted as $sim(u,v)$, between two users, considering the scenarios in common that the users have rated. After all the similarity measures are calculated, in the event of similarity score ties between users, a tie-breaking measure is used, specifically *User Reputation* ($rep(u)$). If there are no ties, it is determined which user has the highest similarity score, and the Q-values from the recommended scenario for the more similar user are set as initial Q-values for the active user.

User Reputation. The user reputation concept refers to how much the other users trust the active user and if the active user always gives a fair trust measure to scenarios (Song et al., 2017). The user reputation, $rep(u)$, is a vital factor in the event of similar score ties. This measure is

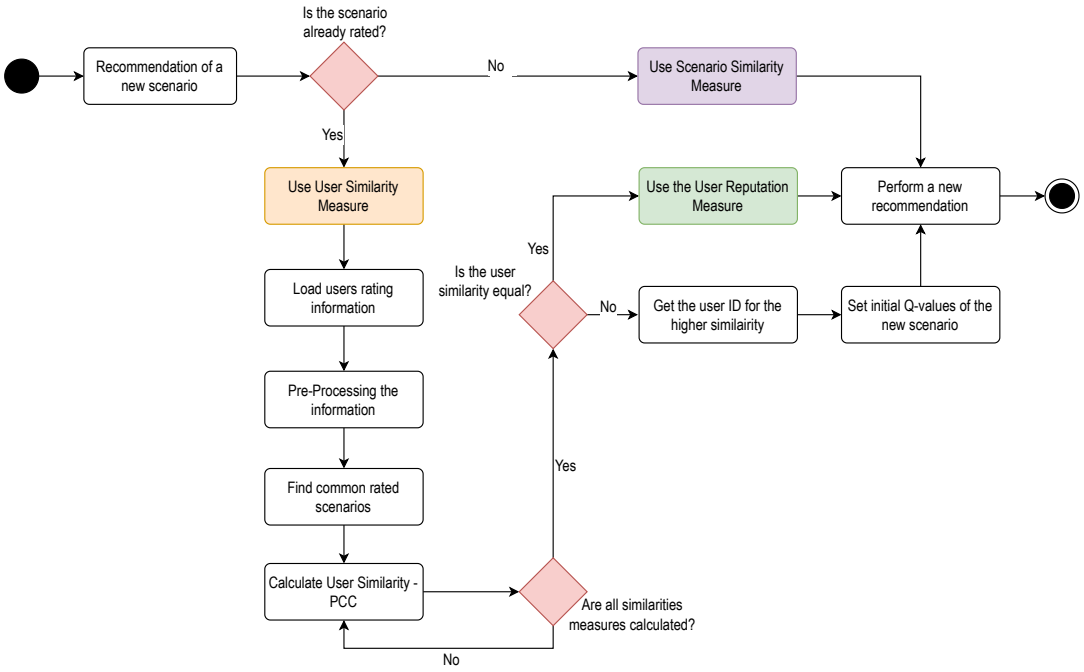

Figure 3.9: User similarity UML activity diagram.

calculated according to Equation 3.10.

$$rep(u) = \frac{\sum_{a_t \in S(u)} \left| UT_R(u,a_t) - \overline{UT_R(a_t)} \right|}{S(u)} + \frac{\sum_{u \in U(u)} \left| Trust_{u,v} - \overline{Trust_u} \right|}{U(u)} \tag{3.10}$$

The calculation is performed by a weighted average of the UT_R as ratings given by the user to a set of scenarios and the other users' trust in the active user. The user trust by other users, $Trust_{u,v}$, is calculated based on the user social trust network presented in subsection 3.4.2. Illustrated in Figure 3.10 is the UML activity diagram that represents the implementation of the user reputation.

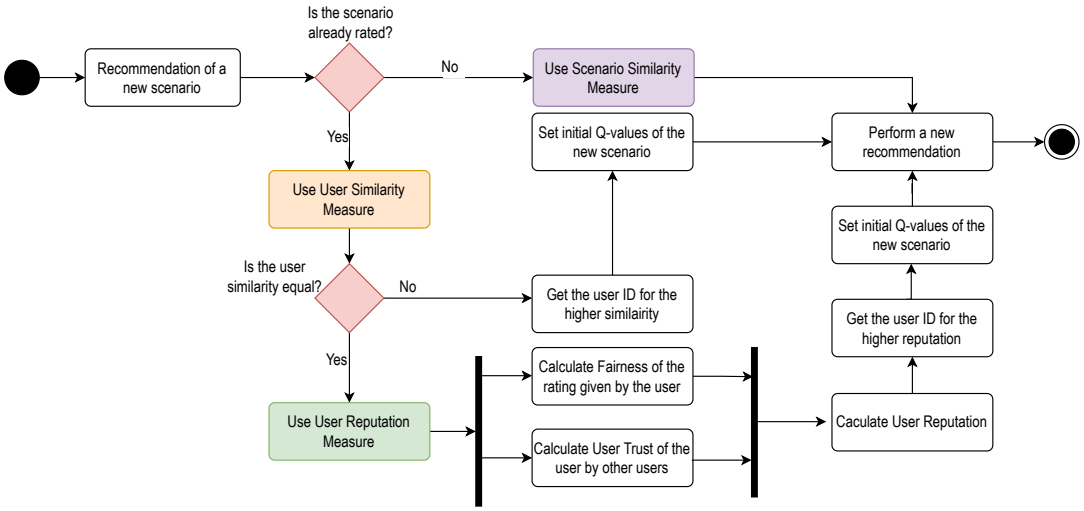

Figure 3.10: User Reputation UML activity diagram.

When user similarity calculations produce identical scores, each user's reputation becomes the decisive factor. In order to resolve the tie, the fairness of each user's ratings and the trust that other users have placed in them is possible to establish the user reputation. With the user's identification with a higher reputation, the Q-values from this user are set as initial Q-values for the active user. Even though user reputation is considered a similarity measure, it is based on trust measures, specifically social trust networks (presented in the Subsection 3.4.2).

3.4.2 Trust Measures

Trust has become a key aspect when performing recommendations, enabling the mitigation of the cold-start and data sparsity problems by leveraging rating information from trusted users, for example. The trust measures used in the proposed model can be divided into two dimensions: the user's direct definition and the user's social behaviour.

Considering the first dimension, this is used in the case of the *Reward Calculation* (Subsection 3.3.2). This is based on the UT_R and the UT_S. The user provides the values for the reward calculation in the first system iteration. However, for the subsequent iterations, the UT_S value is updated according to the performance of the RS. The user trust in the system, UT_S, is calculated according to the following Equation 3.11.

$$UT_S = \begin{cases} C_{UT} - \left(\frac{|E_R - A_R|}{|E_R|}\right), & E_R > A_R \vee E_R < A_R \\ C_{UT} + \theta, & E_R = A_R \end{cases} \tag{3.11}$$

where the C_{UT} is the current values of the user trust in the system, which is established initially by the user and continuously updates at the end of each iteration, $C_{UT} = UT_S$. The value of θ represents a positive value to be defined by the user on how much a trust increment is valid for a correct rating prediction. From the user feedback from the evaluation of the recommended scenarios, it is possible to obtain the values from the UT_R, A_R, E_R, and U_{Acc}, which are used in the UT_S and are also sent to the trust model. The calculation engine, present in the trust model, is responsible for calculating the reward value, r_t, for the recommendation algorithm, based on the UT_S, UT_R, and U_{Acc}.

Additionally, the second dimension of the trust measures comprises the case of the *User Reputation* calculation (Subsection 3.4.1), based on the trust between the users, which is calculated based on a social trust network built on users' trust connections and weights representing the user's trust in the other user. An example of a social trust network is presented in Figure 3.11.

The calculation of the trust between users, $Trust_{u,v}$ is a combination of direct and indirect trust (Chen et al., 2021), which can be calculated according to Equation 3.12.

$$Trust_{u,v} = \begin{cases} Dtrust_{u,v}, Dtrust_{u,v} \neq 0, \\ Itrust_{u,v}, Dtrust_{u,v} = 0, Itrust_{u,v} \neq 0, \\ 0, Dtrust_{u,v} = 0, Itrust_{u,v} = 0 \end{cases} \tag{3.12}$$

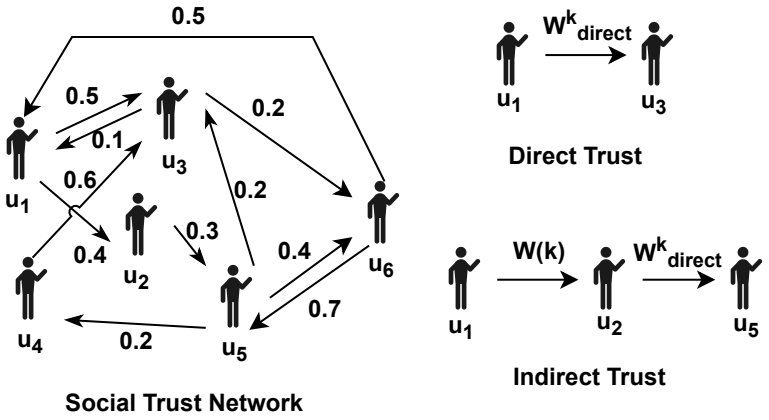

Figure 3.11: Social trust network, and direct and indirect trust.

where $Dtrust_{u,v}$, represents the direct trust between users and $Itrust_{u,v}$, represents the indirect trust between users. The direct trust calculation, $Dtrust$, can use the social trust network with weighted paths between users. The direct trust is calculated by Equation 3.13.

$$Dtrust_{u,v} = W^k_{direct} \tag{3.13}$$

The indirect trust, $Itrust$, is calculated by using W^k_{direct}, which represents the trust value before user u reaches the user v, and also by using the $W(k)$ that represents the weight of the k path that indirectly connects the users. $W(k)$ is calculated by multiplying the direct weight of the path according to Equation 3.14.

$$W(k) = \prod_{i=1}^{l-1} Dtrust_i(x,y) \tag{3.14}$$

The indirect trust, $Itrust$ is calculated according to Equation 3.15.

$$Itrust_{u,v} = \frac{\sum_{k=1}^{n}(W(k) \times W^k_{direct}))}{\sum_{k=1}^{n} W(k))} \tag{3.15}$$

These equations are the base for the calculation of the user reputation value.

3.5 Applying Recommendation Strategies

The recommendation module considers whether the active user has already rated the recommended scenarios or not, implying different recommendation strategies and different equations for calculating the expected rating or rating prediction, E_R. The calculation of the E_R changes according to different recommendation environments that the system is presented with. Figure 3.12 illustrates the different variants of calculating the E_R considering the different recommendation environments.

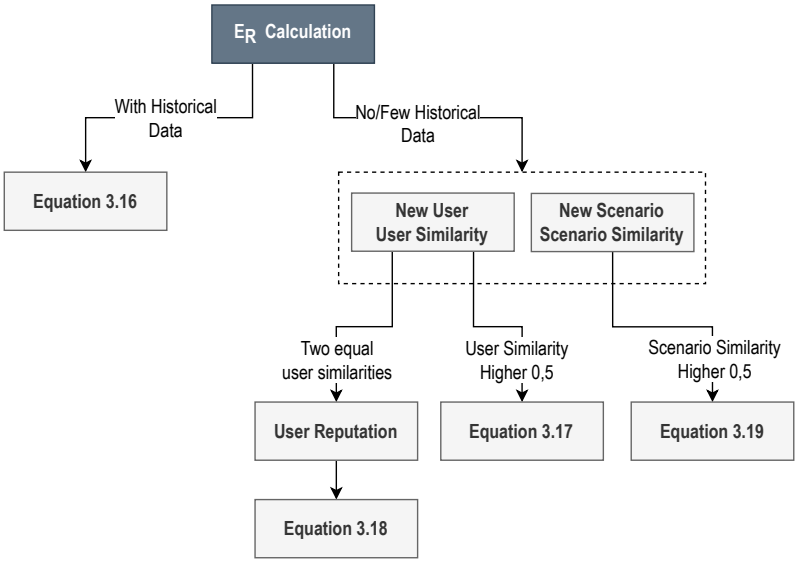

Figure 3.12: Calculating E_R according to the different recommendation environments.

There are at least four main recommendation environments, which are *With Historical Data*, *No Historical Data*, *New User*, and *New Scenario*.

With Historical Data If there is enough information regarding the user and scenario trust rating, the rating prediction is calculated by Equation 3.16.

$$E_R = \frac{\overline{U_{Acc}(u)} + \overline{UT_R(a_t)}}{\overline{UT_R(u)}} \tag{3.16}$$

This equation considers the average user acceptability, with the average trust rating of the scenario divided by the average user trust rating of the active user.

No Historical Data The most common recommendation conditions are with no/few historical information for which significant measures have been established, specifically considering the new users or scenarios. In a general perspective, when there is no historical data, the initial state of the recommendations lies in establishing the initial values of the Q-table as random and performing offline training through the performance of random actions. The training should be done through episodes and by maximising the total reward of each episode. After this, the system recommends the best scenarios and E_R. The user-scenario trust matrix, user acceptability, and UT_S are updated based on the user feedback and acceptability, and r_t is calculated. These calculations will update the Q-table for the active user[1] in iteration 1 to N (it is important to note that each user has its Q-table, which contains the q-values for each scenario) (Pires et al., 2023). Next, the measures for calculating the E_R for a new user and scenario will be presented.

New User In the event of recommending to a new user, the metric of *user similarity*, $sim(u,v)$, is used by the RS to recommend a scenario, given that other users have already rated the scenario. In this case, the PCC equation is used (see Equation 3.9). The Q-values used by the recommenda-

[1] Active user refers to the user currently using the RS.

tion module to build the top-N list are from the most similar user to the active user. Therefore, if the user similarity exceeds 0.5, the E_R is calculated according to Equation 3.17.

$$E_R = \overline{UT_R(u)} + \frac{(UT_R(u,a_t) - \overline{UT_R(u)}) \times sim(u,v)}{sim(u,v)} \tag{3.17}$$

This equation considers the average rating of the user trust, $\overline{UT_R(u)}$, and how much the user trusts that the scenario will work in the physical system $(UT_R(u,a_t) - \overline{UT_R(u)})$.

A specific case of the new user relies upon when the user similarity values are the same and greater than 0.5, for which a tie-breaking measure is applied by considering the *user reputation*, $rep(u)$, defined according to Equation 3.10. The recommendation module uses the Q-values of the user with a higher reputation towards the active user. The E_R, using the user reputation, is calculated according to Equation 3.18.

$$E_R = \overline{UT_R(u)} + \frac{(UT_R(u,a_t) - \overline{UT_R(u)}) \times rep(u)}{rep(u)} \tag{3.18}$$

In this case, instead of using the similarity to calculate the rating prediction, the $rep(u)$ of the user with the higher value is used, considering the average rating trust and how much the user trusts in that scenario.

New Scenario In the case of a new scenario that any user in the system never rated, *scenario similarity*, $sim(a_t,a_j)$, is used to get the best data from a similar scenario. For this purpose, the scenario similarity is calculated by using the COS function based on the DoF values for the tested scenarios through the Equation 3.8. A scenario can be considered new in two situations: when it has never been rated by a specific user (i.e., the active user), is new to that user, or when no system user has rated it.

In this case, the Q-value to be used in the Q-table for the recommendation calculation will be the Q-value from the most similar scenario rated by the active user. If the scenario similarity value is greater than 0.5 for the rating prediction, the E_R is calculated using Equation 3.19.

$$E_R = \overline{UT_R(u)} + \frac{\overline{(UT_R(u,a_t)} + \overline{UT_R(u,a_j))} \times sim(a_t,a_j)}{sim(a_t,a_j)} \tag{3.19}$$

which considers the average trust of the user, $\overline{UT_R(u)}$, plus the average trust of the user in scenario t,$\overline{UT_R(u,a_t)}$, and the average trust of the user in the most similar scenario $\overline{UT_R(u,a_j)}$.

3.6 Summary

In order to minimise the effects of the cold-start and data sparsity problems in the performance of recommendations to new users and new scenarios, this chapter proposes an architecture for a DSS based on the Digital Twin concept integrating six layers. This work focused on three of the six layers of this architecture, focusing on the *Simulation Layer*, *Decision Support Layer*, and *Human Trust Layer*.

Within the Digital Twin architecture, a new RS approach was proposed based on TBR. The conjugation of the features proposed in the *Simulation Layer*, *Decision Support Layer*, and *Human Trust Layer* results in a TBR RS, entitled *SimQL*, which joins what-if simulation model, with an RL-algorithm and a trust-based model. The integration of a RS in a Digital Twin architecture presents several advantages, such as the possibility of a background optimisation of the physical system enabling a proactive RS, access to continuous real-time data from the physical system, and the possibility of integrating new operational parameters to the physical system after validation in an up to date virtual model of the physical system.

Summarising, the main innovative approach aspects associated with the proposed *SimQL* model are the integration within a Digital Twin architecture, the combination of an RL algorithm with similarity and trust measures to minimise the effects of cold-start and data sparsity problems and the different forms of calculating the predicted trust rating to improve the predicting rating calculation accuracy.

Chapter 4

Case Study and Evaluation Measures

The previous chapter described the proposed Digital Twin architecture and the *SimQL* trust-based recommendation approach to perform decision support in the manufacturing domain. In order to ensure the effectiveness of the recommendation approach in terms of performance, it is necessary to perform an experimental evaluation.

This chapter focuses on the chosen case study to test the proposed approach across various manufacturing scenarios. A performance measurement procedure is introduced, divided into evaluation methods and metrics commonly used in the RS field. An evaluation plan has been established, which outlines the steps that will be taken in order to assess and validate the trust-based recommendation approach known as *SimQL*.

The remainder of this chapter is structured as follows:

- Section 4.1: describes the proposed case study of a battery pack assembly line, providing the problem statement and the different recommendation scenarios where the defined *SimQL* trust-based recommendation approach will be tested and validated.
- Section 4.2: defines the performance measurement procedure considering the evaluation methods and metrics already defined and used in the RS state-of-the-art, and the most appropriate evaluation methods and metrics for the case study in question.
- Section 4.3: presents a summary of the information provided in this chapter.

4.1 Experimental Case Study

The application of the case study research method is crucial in generating innovative knowledge and evaluating proposed strategies in real-world scenarios. This method assesses the practicability and feasibility of such approaches, which leads to valuable insights and data-driven solutions. The results obtained in this thesis are verified and validated through one case study in the manufacturing domain, thereby ensuring their applicability.

The objective of this case study is to perform the validation, showcase the feasibility, and highlight the essential features and capabilities of the *SimQL* recommendation approach comprising three main objectives:

- Validation of the what-if simulation model verifying its applicability within the decision support focusing in the RS area.
- Validation of the recommendation approach, *SimQL*, to verify if the system works as specified in average or extreme recommendation conditions (e.g., cold-start, data sparsity).
- Evaluation of the performance of the recommendation approach, allowing to conclude the proposed concepts in the approach.

The proposed model was designed for implementation in manufacturing environments that align with the Industry 4.0 framework. The recommendation model was assessed on a limited scale within a controlled laboratory setting, with a particular focus on evaluating the logistics component of the manufacturing process.

In order to determine the level of maturity of the developed solution, the Technology Readiness Level (TRL) was considered. This measurement assesses the maturity of technology at different stages of research and development (APRE and CDTI, 2022). The TRL scale ranges from 1 to 9, and as the technology advances, its maturity level increases, requiring different resources, actors, and funding possibilities, as illustrated in Figure 4.1.

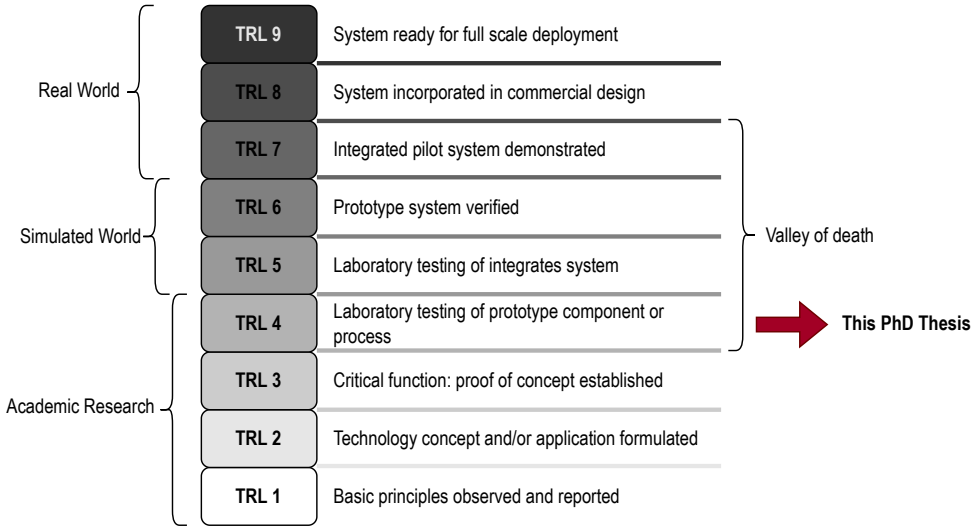

Figure 4.1: Framing of the case study at TRL (Based on (APRE and CDTI, 2022)).

Academic research mostly focuses on TRLs 1 to 4, which encompasses basic and applied research development. Conversely, industrial development spans from TRLs 7 to 9, concentrating on prototype and system development. The "valley of death" refers to the challenging phase between academia and industry adoption, which falls between TRLs 4 to 7. The proposed system finds itself at the TRL 4, being tested in the laboratory using a real case study.

4.1.1 Description of the Case Study

The International Manufacturing Centre at Warwick Manufacturing Group (WMG) has developed a full-scale system that integrates logistics with manufacturing operations on an assembly line. This system showcases advanced Industry 4.0 methods and encompasses new production systems and legacy equipment in various advanced manufacturing scenarios. The system is being utilised for research and training in collaboration with industrial partners. The proposed case study for this PhD thesis is based on this full-scale battery pack assembly line called Integrated Manufacturing & Logistics (IML). The main product assembled on this system is an automotive battery pack, which conjugates industry-standard battery cells and custom-containment modules. As shown in Figure 4.2, the battery pack assembly line is divided into five zones:

- *Zone 1* is a launch manual station responsible for initiating the assembly process through the interaction between the human operator and the MES;
- *Zone 2* is a legacy loop that employs a conveyor system to move the battery modules through four stations, two of them pick and place units which bring together cells forming the packs (the robot stations are manually fed with battery cells);
- *Zone 3* is a welding station that stands alone for pack spot welding;
- *Zone 4* is a quality station that stands alone and performs inspection of the spot welding quality;
- *Zone 5* is a disassembly station.

The assembly line comprises AGVs responsible for running logistics operations of the line. The AGVs system in the assembly line of the described case study is composed of MiR100, which has an average of 10 hours running time (or 20 km), reaching a maximum speed of 1.5 m/s forward and 0.3 m/s backwards. These are powered by a battery (Li-NMC, 24V, 40Ah), taking around 4.5 hours to charge fully.

The manufacturing system is designed to produce battery packs for electric vehicles, each comprising six modules that house 18650 or 26650 form-factor cylindrical cells. Based on MES orders, the transportation of products between different zones is handled by AGVs, which uses conveyor trolleys. The assembly line obeys a task sequence as follows:

- The operator initiates the assembly order for the product to the MES, requesting the AGV to transport the correspondent battery module components.
- The AGV travels to the legacy loop, where the conveyor trolley feeds the legacy loop stations.
- The robot stations fill the modules with battery cells, followed by the pick and place stations that assemble various modules to build a battery pack.

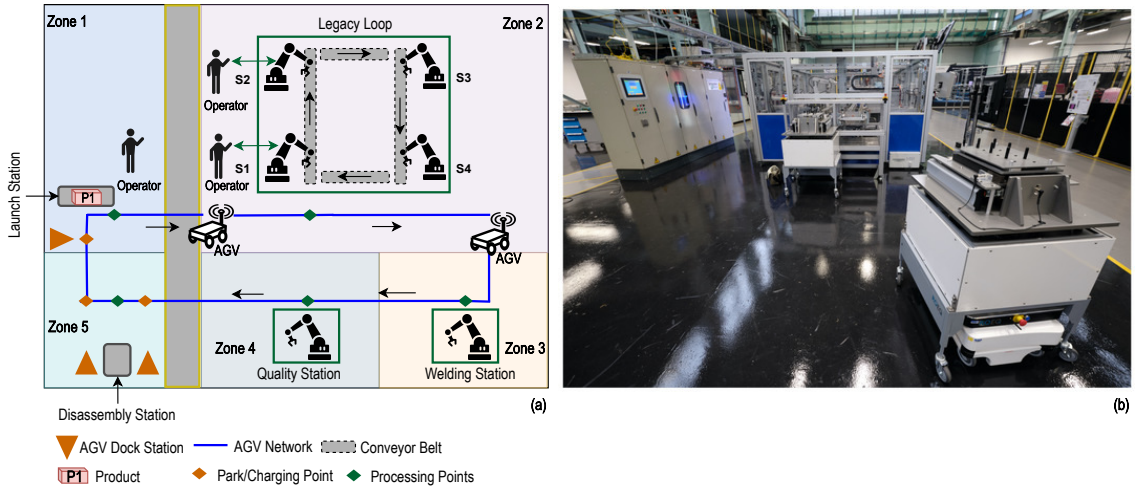

Figure 4.2: IML demonstrator: (a) the IML layout and zones; and (b) the real IML with an AGV carrying battery cells to the legacy loop module.

- The battery pack is assembled and transported on a trolley to the stand-alone welding and inspection stations.
- Lastly, the battery pack is returned to the disassembly station.

Considering the presented case study, a model of the extended battery pack assembly line was considered, providing a more complex and richer benchmark, considering two parts, Part 1 (P1) and Part 2 (P2), and each one served by two sets of AGVs for transporting parts. This model was inspired by the presented case study, increasing its complexity in terms of the logistic operations. The model of the extended assembly line is illustrated on Figure 4.3.

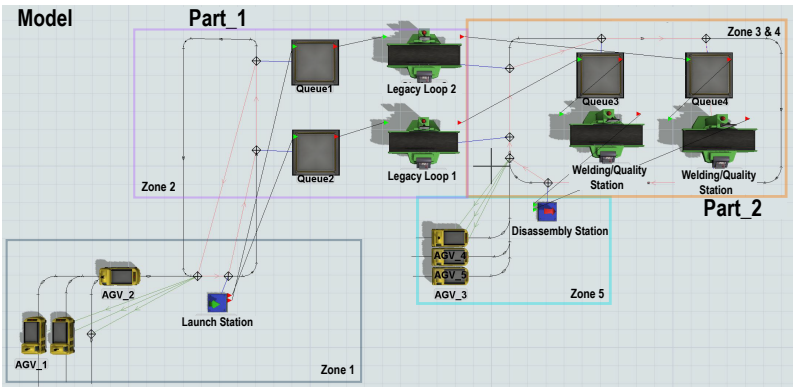

Figure 4.3: Virtual model of the extended assembly line in FlexSim®.

The model includes zones like the IML demonstrator. However, a new legacy loop has been added, and the welding and quality stations have been combined into a single station. Focusing on the logistical operations carried out by the AGVs, the model has been divided into two separate lines, with one serving the legacy loops and the other serving the welding and quality stations. This separation increases the complexity of the case study, allowing for the generation of more

what-if scenarios, increasing the number of AGVs in the system. Despite increasing the system's complexity, the AGVs presented in the model have the same characteristics as the presented case study regarding charging time, running time and maximum speed in the physical world.

4.1.2 Problem Statement

Considering the presented case study, when it comes to making decisions regarding logistical operations, such as determining the necessary number of AGVs to meet product demand or establishing an optimal charging profile of the AGVs for a specific throughput, relying solely on the decision-makers expertise can prove challenging. This is particularly true when the decision-maker is new to the assembly line, having little or no knowledge of how the system has behaved in the past.

To address this problem, a proposed Digital Twin architecture based on the *SimQL* trust-based recommendation approach enables decision support for the new decision-maker. The system leverages information obtained from real-time monitoring, data analysis, what-if simulation, and a RS. The proposed system can analyse hundreds of different configurations before assisting any decision, depending on the DoFs defined for the case study. The DoFs defined for the proposed case study to analyse the different logistical operations are presented in Table 4.1.

Table 4.1: Characterisation of the degrees of freedom.

DoF	Name	Minimum	Maximum	Increment
1	**Recharge Threshold**	10%	80%	10%
2	**Resume Threshold**	30%	90%	10%
3	**N^o of AGVs (P1)**	1	3	1
4	**N^o of AGVs (P2)**	1	3	1
5	**Time Horizon**	8h	24h	8h

Based on this, a set of DoFs were defined comprehending five DoFs:

- ***Recharge Threshold***, DoF 1, can be defined as the percentage level of the battery on which an AGV is required to go the recharging station;
- ***Resume Threshold***, DoF 2, is the percentage level of battery that a charging AGV has to reach to return to the transport route;
- ***Number of AGVs (P1)***, DoF 3, is the number of AGVs in Part 1 of the assembly line;
- ***Number of AGVs (P2)***, DoF 4, representing the number of AGVs in Part 2 of the assembly line;
- ***Time Horizon***, DoF 5, representing the sifts in terms of hours (1 shift - 8 hours, 2 shifts - 16 hours, 3 shifts - 24 hours).

Two general recommendation scenarios were defined considering the proposed case study and the defined DoFs related to the logistical operations.

- ***Recommendation Scenario 1***: the main objective of this scenario is to determine the optimal number of AGVs, which takes into consideration DoF 3 and DoF 4 as the main variables influencing the decision-making process.
- ***Recommendation Scenario 2***: the main objective of this scenario is to determine the optimal number of AGVs and the best charging profile for each established time horizon. This scenario considers all the established DoFs for the case study.

It is important to note that all the experiments performed in the experimental validation (Chapter 5) considered the defined model of the extended assembly line and the established DoFs for the logistical operations. Each experiment considered one of the specified recommendation scenarios.

4.2 Performance Measurement

This section establishes the performance measurement procedure for the established *SimQL* recommendation approach. Performance measurement is the process of using a tool or a procedure to evaluate a specific system parameter. In the RS field, the evaluation procedure can be divided into two, the *Evaluation Methods*, which generally involve the assessment of effectiveness, efficiency, and relevance of a system, including methods such as case studies, experiments and qualitative analysis, and the *Evaluation Metrics*, which can be quantitative or qualitative measures used to assess the performance, effectiveness or quality of a system.

4.2.1 Evaluation Methods

There are different classifications of RS evaluation methods presented in the literature, such as offline and online evaluations (Zheng et al., 2010), data-centric and user-centric (Said, 2013), live user experiments, and offline analysis (Herlocker et al., 2004), and user studies, online and offline evaluations (Ricci et al., 2010; Beel and Langer, 2014). In this work, the focus will be on the evaluation methods proposed by Ricci et al. (2010) and Beel and Langer (2014), illustrated in Figure 4.4.

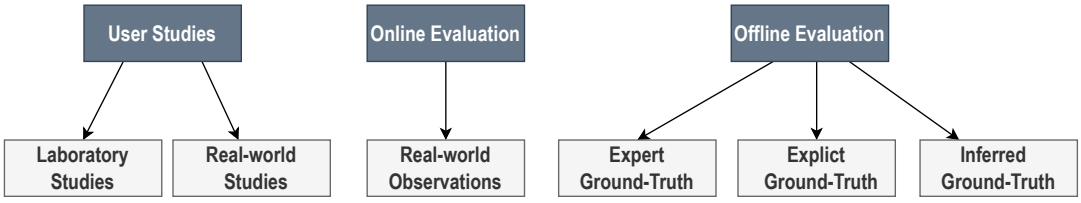

Figure 4.4: Classification of the evaluation methods for recommendation systems (Beel and Langer, 2014).

According to these authors, regarding the evaluation of a RS approach, there are three main evaluation methods: *User Studies*, *Online Evaluation*, and *Offline Evaluation*. The user studies are an effective evaluation method, which essentially measures user satisfaction based on explicit ratings provided by the decision-maker. Decision-makers are asked to rate their overall satisfaction

with the recommendations or other aspects of the RS. Two types of studies can be conducted using this method: laboratory studies, where participants know they are part of a study, and real-world studies, where participants are not informed of the study (Beel and Langer, 2014). The online evaluation method measures the acceptance rates of recommendations in the field of RS. The metrics used in this method assess how the system behaves in real-time with actual users, measuring the impact on user behaviour and engagement. Common metrics used in this method include A/B testing, comparison between different recommendation strategies or algorithms, Click-Through Rate (CTR), which measures the ratio of user clicks on the recommended items, and Conversation Rate (CR), which evaluates the proportion of recommendations that lead to desired user actions (Ricci et al., 2015; Patel and Patel, 2020). Lastly, the offline evaluation method can evaluate the accuracy, efficiency and reliability of a RS based on ground truth. Ground truth refers to datasets that contain explicit, inferred, or expert information. Explicit ground truth contains data on items that users have rated or liked. The system can be evaluated by removing certain ratings and predicting their values. The closer the predicted ratings are to the original ratings, the more accurate the system is. The inferred ground truth is based on items in the personal list of users, and it is considered accurate if the system recommends the items in the user list. The expert ground truth is based on item classification by experts, using these items to train the system to recommend items of similar categories. This method uses non-real-time datasets to evaluate, including measures such as RMSE, MAE, precision, and recall (Beel and Langer, 2014).

Considering the advantages of applying the evaluation method of offline evaluation, such as in terms of scalability, it enables a more efficient way of algorithm analysis, promotes a more cost-effective way to assess the recommendation approach since it does not require the actual users, enables the performance of an exploratory analysis, reduces the privacy concerns, when accessing the real user data can be an issue, it allows to have more control over the experiments being conducted, helping in understanding the impact and performance of an algorithm in specific circumstances (e.g., cold-start and data sparsity), and provides a way to work with historical data when data availability is a problem. Based on these advantages and the characteristics of the selected case study to validate the *SimQL* trust-based recommendation approach, the evaluation method for this work is the offline evaluation, using the explicit ground-truth method.

4.2.2 Evaluation Metrics

The evaluation metrics in the context of RS can be classified into *Quantitative Metrics* and *Qualitative Metrics*. Quantitative metrics are based on the direct and quantifiable assessment of the RS performance, and the qualitative measures are more subjective and reflect properties related to the user experience on how the decision-makers perceive and interact with the produced recommendations. Using both measures can help improve the user experience and overall effectiveness of the RS. In the presented case study, the evaluation of the recommendation model will be conducted using an offline evaluation methodology, being the explicit ground truth, focusing on quantitative metrics.

The recommendation quantitative metrics can be divided into two major groups: the ***Statistical Accuracy Metrics*** and the ***Decision Support Accuracy Metrics***. Statistical accuracy metrics are measures used to evaluate a prediction algorithm's accuracy by comparing the predicted ratings' deviation with the actual ratings. The decision support accuracy metrics evaluate how effective the recommendations are to the users in selecting quality items (Papagelis et al., 2005).

Statistical Accuracy Metrics The statistical accuracy metrics evaluate the proposed RS by assessing the results by calculating the average over the calculated deviations between ratings. Within these types of measures are include metrics such as RMSE, and MAE (Patel and Patel, 2020; Gaillard, 2014; Papagelis et al., 2005).

The measure of RMSE (see Equation 4.1) is used to evaluate and compare the performance of a RS model compared to other models, being a measure of the stability of predictions (Frémal and Lecron, 2017; Isinkaye et al., 2015).

$$RMSE = \sqrt{\frac{1}{N_p}\sum_{u,i}^{N}|p_{ui} - r_{ui}|^2} \tag{4.1}$$

where N_p is the total number of rating predictions, p_{ui} is the predicted rating that a decision-maker, u, will select an item, i, and r_{ui} is the real rating. In this case, the lower it is RMSE, the better the recommendation accuracy/performance of the algorithm.

The MAE (see Equation 4.2) is one of the most popular and commonly used measures for RS, being a measure of the efficiency of predictions (Frémal and Lecron, 2017; Isinkaye et al., 2015). This measures the deviation of recommendation from the decision-maker's specific value.

$$MAE = \frac{1}{N}\sum_{u,i}^{N}|p_{u,i} - r_{u,i}| \tag{4.2}$$

where N_p is the total number of ratings on the item set, p_{ui} is the predicted rating that a decision-maker, u, will select an item, i, and r_{ui} is the real rating. The lower the MAE, the more accurately it works the RS engine in predicting the decision-maker ratings.

In the case of the conjugation of the two measures, if a dataset has a small MAE but a high RMSE, it means that generally, the predictions are near correct values, but there are some strongly incorrect results (Frémal and Lecron, 2017).

Decision Support Accuracy Metrics The decision support accuracy metrics are used to evaluate the top-N recommendations for a decision-maker. There are RS which produce recommendations as a ranked list of items, ordered by decreasing relevance. These measures are related to the decision-maker's ability to select high-quality recommendations for the offered items. These metrics include *precision*, *recall*, and *F1 score* (Gaillard, 2014).

For the computation of these metrics, it is necessary to take into account the following four different values:

- **True Positive (TP)**: the system recommends an item that the decision-maker is interested in;

- **False Positive (FP)**: the system recommends an item that the decision-maker is not interested in;
- **True Negative (TN)**: the system does not recommend an item that the decision-maker is not interested in;
- **False Negative (FN)**: the system does not recommend an item the decision-maker is interested in.

These values, TP, FP, TN and FN, are used to build a confusion matrix that represents the four possible outcomes of any recommendation, and if the recommended item is relevant to a decision-maker, it will be considered successful; otherwise, it is not successful. The metrics mentioned above are computed based on calculating the confusion matrix (see Table 4.2).

Table 4.2: Confusion matrix for a recommendation system.

	Successful Recommendation	**Unsuccessful Recommendation**
Recommended	TP	FP
Not Recommended	FN	TN

The precision determines the proportion of relevant items in the recommended list presented to the decision-maker (Fayyaz et al., 2020). The calculation is performed according to Equation 4.3.

$$Precision = \frac{TP}{TP+FP} \tag{4.3}$$

The recall metric, calculated according to Equation 4.4, represents the proportion of relevant recommended items to the total number of items that should be recommended, measuring the coverage of the recommended items (Fayyaz et al., 2020).

$$Recall = \frac{TP}{TP+FN} \tag{4.4}$$

The F1 score (see Equation 4.5) is one of the most common F-measures derived from the precision and recall measures, conveying the balance between these two measures. If the measure is 1, it means that the precision and recall are perfect, while if it is 0, it implies that it is not possible to have precision and recall (Fayyaz et al., 2020).

$$F1score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{4.5}$$

In the state-of-the-art assessment of recommendation approaches, the most used metrics are RMSE and MAE, which are also the choice for the assessment of the *SimQL* approach.

4.3 Summary

The case study of IML, the battery pack assembly line, involves the analysis of different scenarios that include various logistical challenges like the optimal number of AGVs and their battery charg-

ing and resumes profiles. However, using the real system for experimentation is not feasible for several reasons, firstly, using the real system requires a significant number of human interactions, making it inefficient in testing diverse experimental scenarios. Secondly, ethical concerns arise when considering the human involvement in the experimental setup, such as consent, participant safety, and privacy. Lastly, sustaining the real system demands the availability of multiple individuals to cover shifts during its operational period, posing logistical challenges and increasing the necessity of human resources.

In this way, regarding the evaluation of the *SimQL* approach is going to be applied an offline evaluation method, based on the characteristics of the case study and the advantages of the method, for example, the ability to help in the understanding of the impact and performance of an algorithm in specific circumstances as is the case of cold-start and data sparsity problems. Considering the evaluation metrics used in the state-of-the-art and the chosen evaluation method, the RMSE and MAE were the chosen evaluation metrics.

Chapter 5

Experimental Validation and Results

In the previous chapter, the case study and the problem statement were presented along with the evaluation procedure, including the evaluation method and metrics for assessing the *SimQL* trust-based recommendation approach. The validation of the proposed recommendation approach plays a vital role in guaranteeing its applicability and ensuring the quality and accuracy of the recommendations that are produced.

This chapter intends to present the experimental validation of the proposed *SimQL* trust-based recommendation approach based on an academic case study, described in Section 4.1, related to a battery pack assembly line, called IML, at WMG, University of Warwick, that lies within the manufacturing domain. The performance measures it allows to assess the proposed approach are also described in Section 4.2. The performance of the experimental validation of the *SimQL* approach enables the answer to the research questions set out at the beginning of this document (Section 1.2) and proves the thesis statement.

The remainder of this chapter is structured as follows:

- Section 5.1: presents the preliminary experiments assessing the validation of the space scenario generation for the what-if simulation, the validation of the RL algorithm capability to perform recommendations and optimal parameters, and the validation of the similarity and trust measures.

- Section 5.2: presents the comparison in terms of performance of the *SimQL* approach with a simpler version, *QL* algorithm, and with the traditional and social state-of-the-art approaches, using the RMSE and the MAE evaluation metrics.

- Section 5.3: presents the results for the sensitivity analysis for the *SimQL* approach performed using a fuzzy logic approach.

- Section 5.4: summarises results presented this chapter.

5.1 Preliminary Experiments

In the preliminary phase of this PhD research work, the results from the validation of the individual parts, as the what-if engine and simulation, the RL algorithm for performing recommendations, and the similarity and trust measures of the *SimQL* recommendation approach are presented. It is important to note that these experiments do not intend to validate the *SimQL* approach. However, instead, the achieved results will demonstrate the functionality of each one of the blocks that make up this approach.

5.1.1 Validation of the What-if Engine

The validation of the what-if engine was performed following experiments going from simpler to more complex examples considering the model of the extended assembly line. Each experiment has its own goal related to the action of performing decision-making (e.g., decide what is the best number of AGVs), having to define which DoFs are possible to be involved in making that decision, which will be used in the scenario generation. An extended description of the DoFs is provided in Subsection 4.1.1.

The experiments were performed using an Intel Core M-5Y71 1.20GHz CPU with 8 GB RAM on a Windows 10 Pro System, using Python 3.7 to implement the what-if engine and the FlexSim®simulation software to perform the simulation of each generated what-if scenario.

Experiment WT-IF1: considering the ***Recommendation Scenario 1***, the main goal of this experiment was to determine what was the best number of AGVs for the assembly line represented in the model (defined in Subsection 4.1.1) considering a fixed recharge and resume threshold of 30% and 80%, respectively (DoF 1 and DoF 2). The established independent DoF where (DoF 3) the number of AGVs in Part 1 (from 1 to 3 with the increment of 1), (DoF 4) the number of AGVs in Part 2 (from 1 to 3 with the increment of 1), and the (DoF 5) time horizon of 24h. Figure 5.1 illustrates the relationship between the DoFs in ***Experiment WT-IF1*** for the generation of the what-if scenarios.

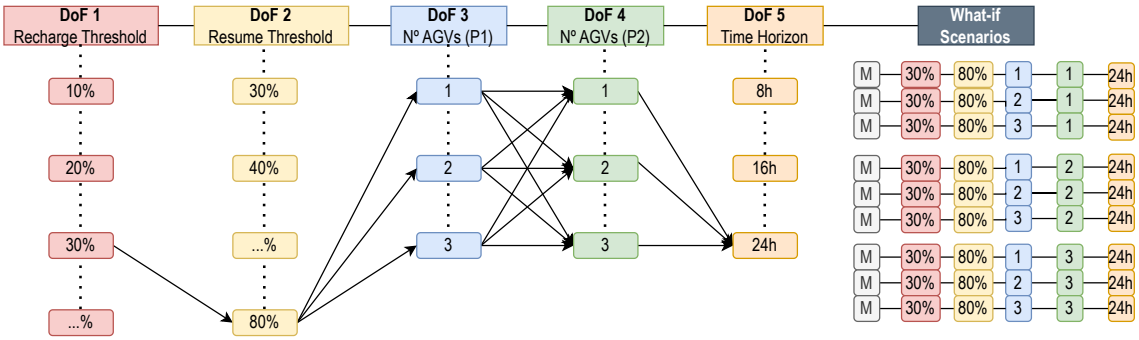

Figure 5.1: What-if engine, ***Experiment WT-IF1***, generating 9 what-if scenarios.

Based on the established what-if engine and the established DoFs for the experiment, it was possible to generate 9 what-if scenarios (Pires et al., 2021b). The total execution time for the what-

if engine to generate the 9 what-if scenarios was 0.002 seconds, and the total simulation time was 0.81 hours.

Experiment WT-IF2: considering the ***Recommendation Scenario 2***, in this experiment, it was considered a general assessment of what would be the best assembly scenario in terms of the number AGVs, charging profile for the different time horizons, having as base all the five DoFs, the recharge threshold (DoF 1), ranging from 30% to 70% with increments of 10%, the resume threshold (DoF 2) ranging from 40% to 90% with increments of 10%, the number of AGVs per semi-line, varying between 1 and 3 (DoF 3 and DoF 4), and the time horizon varying between 8, 16 and 24h (DoF 5). Figure 5.2 illustrates the relationship between the DoFs in ***Experiment WT-IF2*** for the generation of the what-if scenarios.

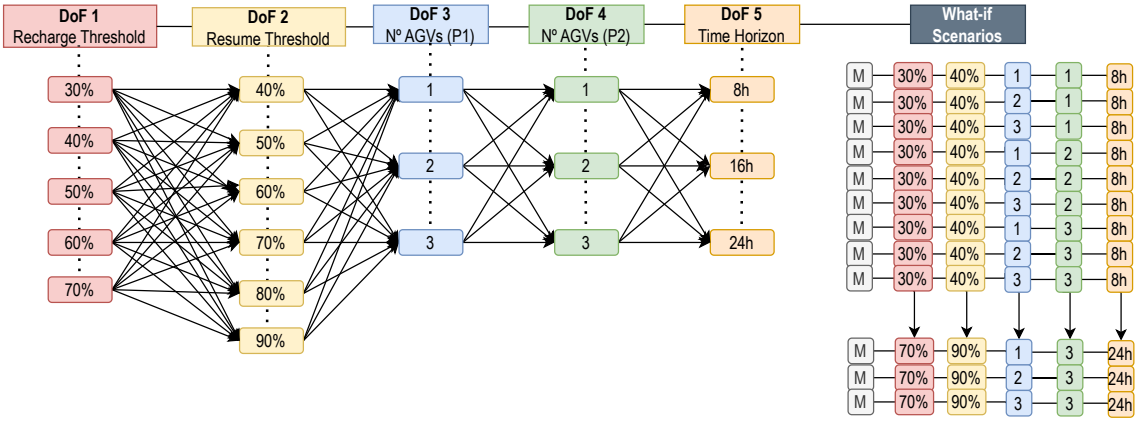

Figure 5.2: What-if engine, ***Experiment WT-IF2***, generating 540 what-if scenarios.

Based on the established what-if engine and DoFs established for the experiment, it was possible to generate 540 what-if scenarios (Pires et al., 2023). The total execution time for the what-if engine to generate the 540 what-if scenarios was 0.053 seconds, and the total simulation time was 13.73 hours.

The what-if engine was validated considering the defined model and the recommendation scenarios in Subsection 4.1.2. The two experiments show the capability of the what-if engine to generate different testing scenarios even with increased complexity, involving five dynamic DoFs. The engine can also handle the restrictions/dependencies between DoFs (e.g., the recharge threshold has to be smaller than the resume threshold), possibly adding more restrictions depending on the case study. In terms of the time for generating the what-if scenarios, the system can generate relatively quickly (e.g., it took 0.053 seconds to generate 540 scenarios), even with reduced computational power. Regarding the simulation time, the system takes 13.73 hours to simulate 540 what-if scenarios given the DoFs used for the recommendation scenarios, but this can be improved by providing a more powerful computational platform to simulate all the what-if scenarios.

5.1.2 Validation of the RL algorithm

The validation of the RL algorithm, in this case, the Q-Learning algorithm to perform recommendations, considers the established virtual model and the what-if scenarios generated in ***Experiment WT-IF1*** of the validation of the what-if engine. As was mentioned before, the main goal of the experiment was to determine the best number of AGVs for the assembly line (***Recommendation Scenario 1***). After having the what-if scenarios simulated, the best scenario will be recommended based on the Q-Learning algorithm, integrated with similarity and trust measures.

The recommendation algorithm was implemented using the Python programming language, and the initial Q-Table was attained by filling it with random values. Two trust states were defined, "1" if "trustworthy" and "0" if "untrustworthy". For the trustworthy states towards the scenarios, the reward is 1 if this has a throughput above average and presents two or less AGVs in the Part 1 and two or less AGVs in the Part 2. If one of these parameters is not met, the reward is 0.5. In the case of untrustworthy states, the reward will be -1. After performing the training phase, the first recommendation comprises the three best scenarios ranked according to the following value calculation $R_{value} = Q(1,a_t) - Q(0,a_t)$, which penalises the untrustworthy scenarios.

The algorithm is responsible for learning from the user feedback and giving an appropriate recommendation in a timely manner. Considering all the variables involved in the Q-function, the learning rate, $\alpha, (0 < \alpha < 1)$, determines the rate at which the algorithm will learn new information.

Based on this, an experiment was conducted to determine the capacity of the Q-Learning algorithm to perform recommendations of the what-if scenarios (Note: the what-if scenarios can be identified in Figure 5.3 by their ID on the right on each graph) in a timely manner varying the learning rate from 0.1 to 0.9 with increments of 0.1, verifying the effect of the learning rate in the RS. Figure 5.3 illustrates the results of the performed experiment.

The results show that the learning process for the learning rate of 0.1 is very slow, taking at least four iterations to learn from the users' feedback and change the order of the scenarios. Although the order changes, the system does not propose a new scenario for replacing the scenario with no user interest in the ten iterations performed. With a learning rate of 0.7, it is possible to observe a faster pace in the learning process, taking only one iteration to learn and propose a new scenario. However, after the third iteration, it is possible to note an unstable behaviour in selecting alternative scenarios due to the existence of a few scenarios and the fact that the Q-Learning algorithm is value-based. The exploration of alternative recommendations is restricted to the algorithm's available scenarios and learning capabilities. Therefore, there is a need to adjust the ideal learning rate value to balance the learning process between a fast pace and one that does not lead to an unstable and chaotic situation. Considering the presented results, it is possible to conclude that the Q-Learning algorithm, as a recommendation algorithm, can perform timely recommendations, depending on the choice of learning rate value.

Considering the obtained results, from now on, the learning rate to be used in all the experiments involving the Q-Learning algorithm is $\alpha = 0.7$.

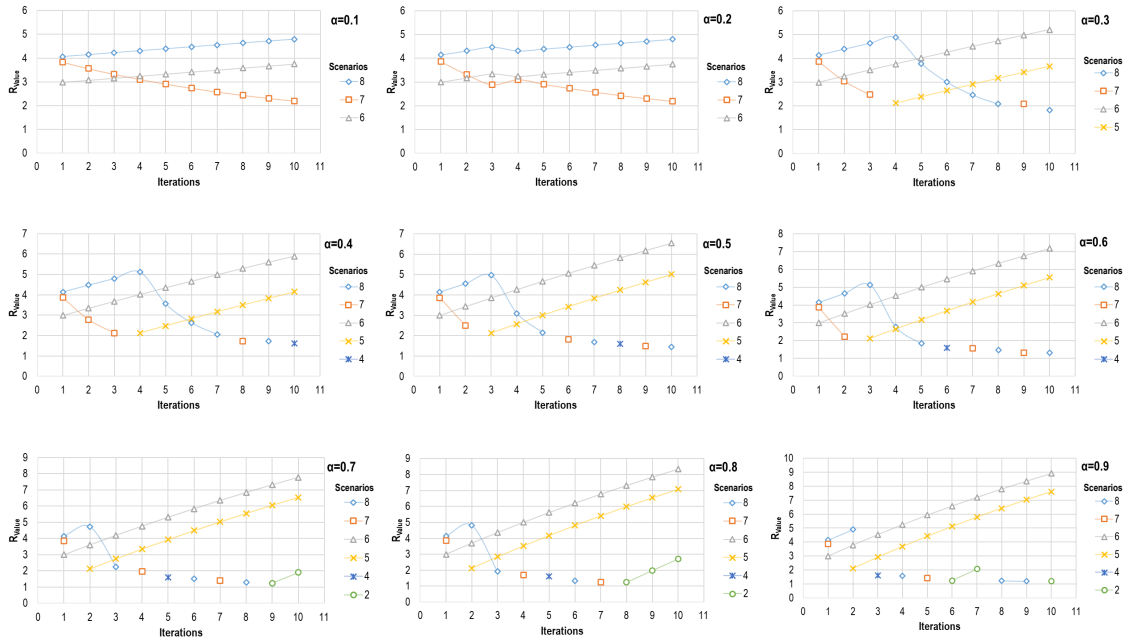

Figure 5.3: Comparison of results for $0.1 \leq \alpha \leq 0.9$.

5.1.3 Validation of Similarity and Trust Measures

For the validation of the application of similarity and trust measures in the *SimQL* approach, it was considered experiments focusing on the recommendation of the scenarios with the best number of AGVs for the assembly line (***Recommendation Scenario 1***), considering the system's performance measured by the throughput and by the user trust history about the previously recommended scenarios. In this case, the defined virtual model was considered to illustrate the basic functioning of the *SimQL* recommendation approach and the results from applying the similarity and trust measures. For these experiments, it was taken into consideration the results from the what-if simulation obtained in the ***Experiment WT-IF1***, being the simulation results of the what-if scenarios summarised in Table 5.1.

Table 5.1: Simulation results for the what-if scenarios from ***Experiment WT-IF1***.

Scenario ID	DoF 1	DoF 2	DoF 3	DoF 4	DoF 5	Throughput	Charging Time (h)
1	30%	80%	1	1	24h	1403	9.03
2	30%	80%	1	2	24h	1677	13.56
3	30%	80%	1	3	24h	1524	15.06
4	30%	80%	2	1	24h	1219	10.54
5	30%	80%	2	2	24h	1725	16.56
6	30%	80%	2	3	24h	2103	22.59
7	30%	80%	3	1	24h	1139	13.55
8	30%	80%	3	2	24h	1498	18.06
9	30%	80%	3	3	24h	2075	22.58

The what-if scenarios present the DoFs, some simulation results, the throughput and the total

charging time of the AGV system. In order to test the proposed approach, it was also considered different trust rating profiles for the users according to the information in Table 5.2.

Table 5.2: Characterisation of the user's trust rating profile.

User	Description
User #1 (u_1)	Trust rating profile for scenarios with a high number of AGVs (five to six AGVs in total for the assembly line)
User #2 (u_2)	Trust rating profile for scenarios with the same AGVs number in the Part 1 (P1) and Part 2 (P2)
User #3 (u_3)	Trust rating profile with a preference for scenarios with three AGVs in the Part 2 (P2)

The experimental validation was performed in two types of experiments: the first with *Decision-Makers Experiments*, presenting the results of the recommendations according to the decision-maker trust rating profile, and after a change in that profile, and the second with the *Cold-Start Experiments*, in which new scenarios that other decision-makers never rated were included, and also a variation where other decision-makers already rated the new scenario.

5.1.3.1 Decision-Maker Experiments

The first set of experiments presents the basic functioning of the *SimQL* approach based on three different decision-makers trust rating profiles (see Table 5.2). Figure 5.4 illustrates the recommendation results for the trust rating profile of *User 1*.

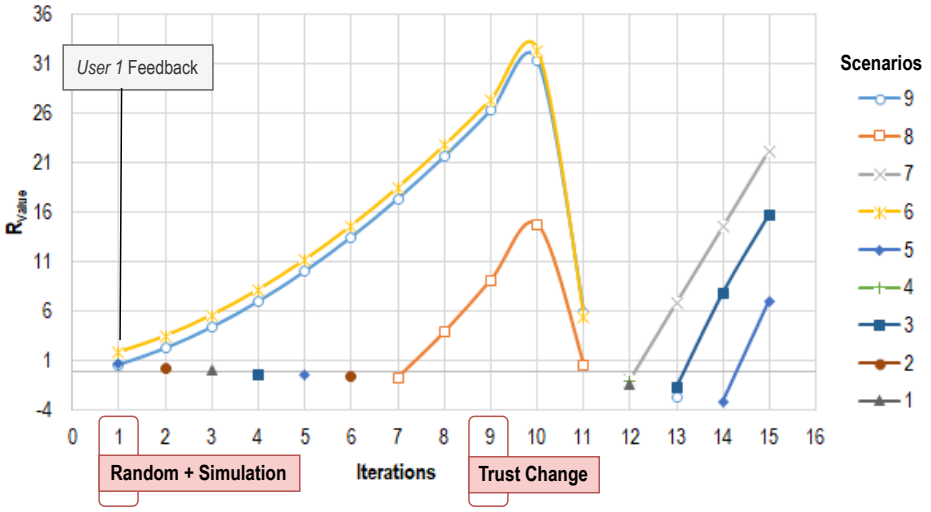

Figure 5.4: Recommendation results considering *User 1* trust rating profile change.

The results show that the RS model is capable of learning throughout time, being able to suggest scenarios aligned with the preferences of *User 1*, which are scenarios with a higher number of AGVs. Initially, the recommendations are based on a random initialisation and the simulation results of the scenarios, with the recommended scenarios presenting throughput values higher than average (in this case, scenarios #9, #6, and #5). After the first iteration, the user starts providing its trust rating for the recommendations. Considering the *User 1* trust rating profile, the system takes

six iterations to learn the profile preferences of the user and recommend scenarios #9, #8, and #6, which are the scenarios with the higher number of AGVs.

In order to demonstrate the system's adapting capabilities for changes in the user trust rating profile, there is a shift in the *User 1* trust rating profile at the ninth iteration. This can be justified, for example, by a shift in the user perception of what is best for the functioning of the line. In this way, the *User 1* shifts its trust rating profile, starting to prefer scenarios with four AGVs in total (e.g., two AGVs in the Part 1 and two AGVs in the Part 2). The system begins recommending the new scenarios that follow the change of the user trust profile on the twelfth iteration. However, only on the fourteenth iteration, the system recommends scenarios with four AGVs, i.e., scenarios #7, #3, and #5. This means that the RS learns takes four iterations to perform recommendations according to the new trust rating profile of the user.

5.1.3.2 Cold-Start Experiments

The second set of experiments considers the cold-start problem and considers that there are two types of experiments being performed, defined as follows:

- ***Experiment CSE1:*** relates to the recommendation performance of a new what-if scenario never rated by any user in the system.
- ***Experiment CSE2:*** considers the recommendation of a new what-if scenario for a specific user, which was already rated by other users in the system.

Considering the recommendation approach performing recommendations, for the ***Experiment CSE1***, the results are presented in Figure 5.5 illustrating the recommendation without any mitigation measure.

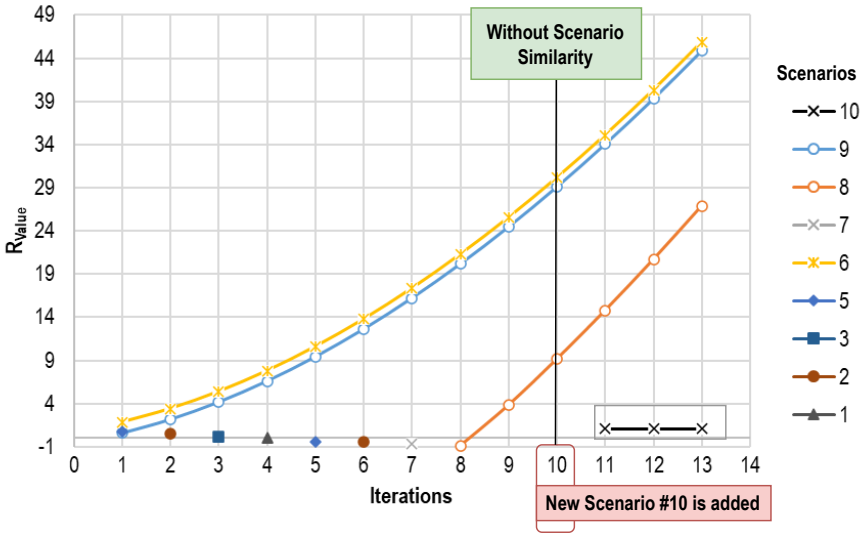

Figure 5.5: Recommendation results considering *Without Scenario Similarity* measure.

This graph presents the recommendations being performed considering the initial set of scenarios, and when reaching the tenth iteration with the system, a new what-if scenario (#10) is

added to the system. This scenario (#10) has the following DoFs, DoF 1 20% recharge threshold, DoF 2 50% resume threshold, DoF 3 with 3 AGV in P1, DoF 4 with 3 AGV in P2, and DoF 5 with 8 hours time horizon. This new scenario has no rating history, suffering from cold-start problems. Even though this what-if scenario has the qualities that the user rating is looking for, without similarity measures, it will only recommended when the RL algorithm is performing exploration of new what-if scenarios, which can take time. Figure 5.6 illustrates the recommendation results considering the application of the *Scenario Similarity* measure for a new what-if scenario that was never rated (***Experiment CSE1***). The new what-if scenario is added at the tenth iteration of the user with the system.

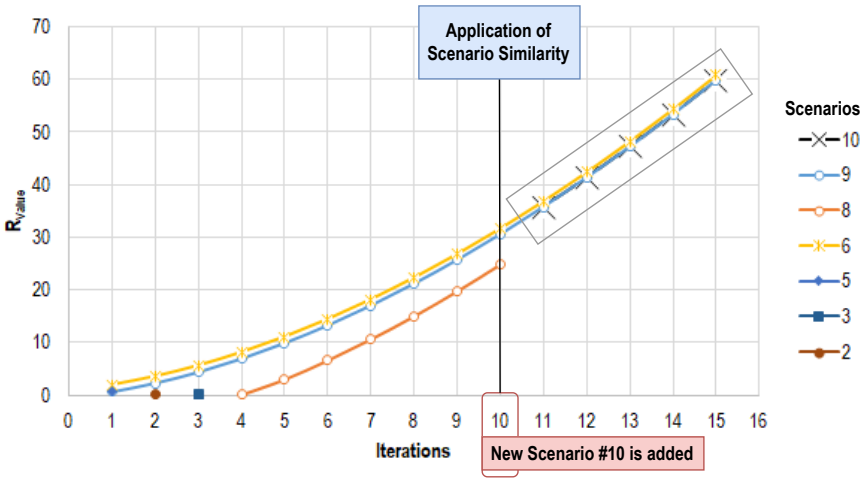

Figure 5.6: Recommendation results considering *Scenario Similarity* measure.

In this case, to mitigate this problem, the *Scenario Similarity* $(sim(a_{10}, a_j))$ between this new what-if scenario and the other rated what-if scenarios was calculated by calculating the scenario similarity, it was possible to conclude that the most similar scenario is scenario #9 $(sim(a_{10}, a_9) = 0,989849)$. Therefore, to have additional initial values to perform the recommendation, the Q-values from the most similar scenario are used as the initial values for the new what-if scenario. Based on this and the trust profile of u_1, the system recommends the new scenario as one of the three best scenarios to apply in the eleventh iteration.

Comparing the graph in Figure 5.5 with the one in Figure 5.6, it is possible to conclude that when the system does not have embedded mitigation techniques for the cold-start problem, the system cannot recommend the new scenario as fast as considering a system with *Scenario Similarity*. This proves that using the *Scenario Similarity* effectively handles the new scenarios cold-start problem within the proposed RS.

Considering the ***Experiment CSE2***, the implemented mitigation techniques for the cold-start problem are *User Similarity* and *User Reputation*. Figure 5.7 illustrates the recommendation results for three different users with different trust rating profiles, considering the iteration of the system throughout time.

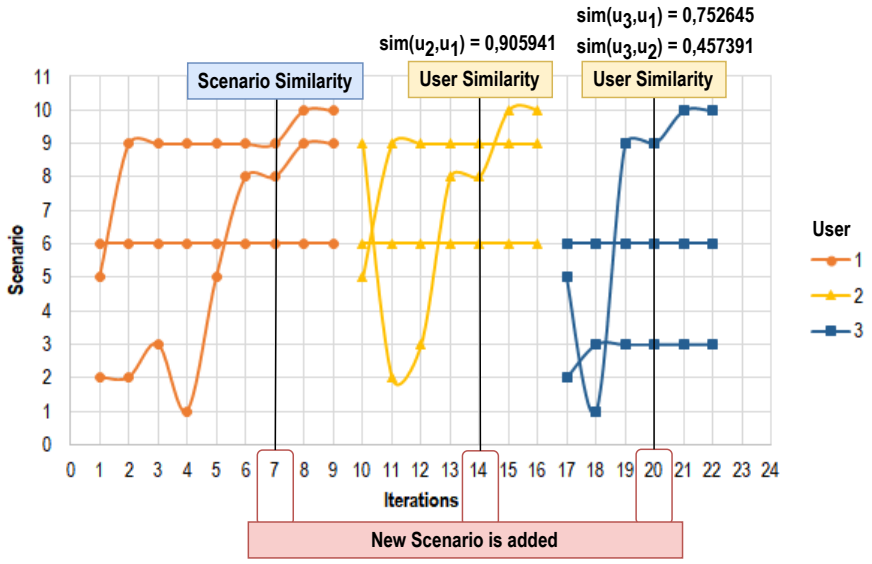

Figure 5.7: Recommendation results considering *User Similarity* measure.

The initial recommendations until the ninth iteration were made for u_1 for which the new scenario was added at the seventh iteration, and since there was no history on this scenario, it was calculated the *Scenario Similarity*. Following this, u_2 started requesting recommendations, and the new scenario (#10) was added at the fourteenth iteration. Since there is a trust rating history from u_1, it is possible to calculate the user similarity $(sim(u_2,u_1) = 0,905941)$. Since the used similarity value is greater than 0.5, the Q-values of the new scenario from u_1 are set as the initial values for u_2. If the user similarity value is less than 0.5, the scenario similarity would be calculated since 0.5 was the established threshold for considering the admissible user similarity value. Lastly, u_3 starts requesting recommendations, and the new scenario (#10) is added at the twentieth iteration. At this moment, the user similarity between u_3 ant the other two users is calculated $(sim(u_3,u_1) = 0,752645, sim(u_3,u_2) = 0,457391)$. This means that the user with higher similarity to u_3 is u_1. Therefore, the system assigns the u_1 Q-values from the new scenario to u_3 as the initial values.

Figure 5.8 presents the results from a specific experiment in which the application of the user similarity is not enough to decide which user is more similar, and it is necessary to apply the *User Reputation*. For this experiment, it was considered the same trust rating profile for u_1 and u_2.

The recommendation for u_3 started at the twenty-first iteration, and the new scenario (#10) was added at the twenty-fourth iteration. Since the user similarity between u_3 and the other two users are the same, i.e., $sim(u_3,u_1) = sim(u_3,u_2) = 0,819810$, the user reputation is applied as a tie-breaking measure. Considering the values from the social trust network, the reputation of u_1 is $rep(u_1) = 0,168013$, and the reputation of u_2 is $rep(u_2) = 0,301$, which means that the reputation of the u_2 is the higher value and the Q-values of the new scenario from the u_2 are assigned as the initial values for u_3 new scenario, mitigating the cold-start problem.

Considering the results of these experiments, it is possible to conclude that applying mitigation

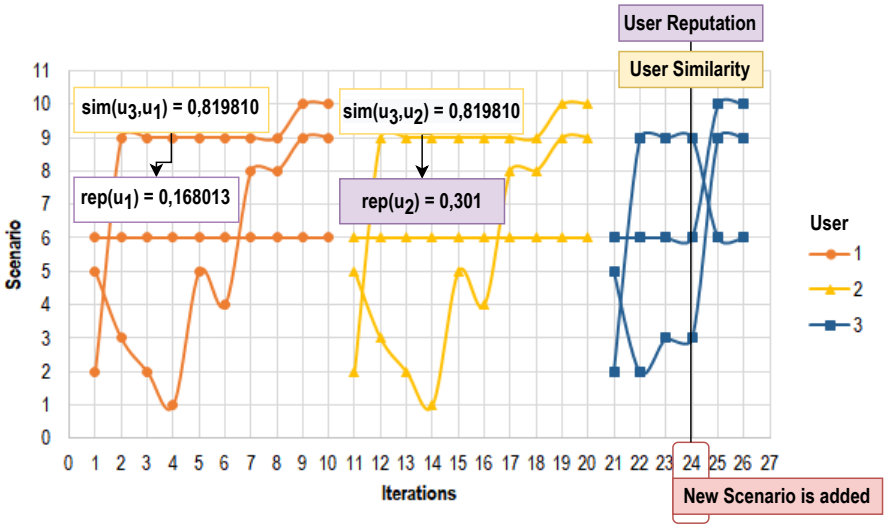

Figure 5.8: Recommendation results considering *User Reputation* measure.

techniques as similarity and trust measures improves the performance in cold-start situations of the *SimQL* recommendation approach, i.e., converging faster and more accurately to the desired system configurations. Particularly, the recommendation approach can adapt to the user trust changes in the trust rating profile (taking three to five iterations to learn the new trust tendency of the user). In cold-start situations, the system can provide recommendations more efficiently by applying similarity measures, such as scenario and user similarity, instead of the RL algorithm's random initialisation of the q-values.

5.2 Comparison with State-of-the-Art Approaches

This set of experiments aims to evaluate the metrics of the proposed recommendation approach by comparing it with the state-of-the-art approaches. For these experiments, it was considered the model of the case study, considering also the set of simulated what-if scenarios attained based on the ***Experiment WT-IF2*** performed in Subsection 5.1.1. The *SimQL* recommendation approach is used to recommend the best logistical scenario from the set of generated scenarios (***Recommendation Scenario 2***). The what-if scenarios are evaluated in terms of the best number of AGVs operating in the line, considering the analysis of the results from the simulation, the user trust rating, the trust history in the RS, and the user social trust network.

The main purpose of the *SimQL* approach is to recommend the best scenarios according to the individual users' trust rating profile. The created datasets used to support this evaluation, represented in Table 5.3, are divided into sub-datasets with a different number of users, scenarios, density, and sparsity levels, and it was applied a k-fold cross-validation technique, 5-fold cross-validation. This technique is usually used to evaluate the performance of a model, where the dataset is split into k number of folds, where k refers to the number of groups the data sample is split into.

Table 5.3: Characterisation of the experimental datasets.

Dataset	Users	Scenarios	Ratings	Avg.N°Ratings/User	Avg.N°Ratings/Scenario	Density	Sparsity
D1	2	23	31	15,50	1,35	67,39%	32,61%
D2	3	26	48	16,00	1,85	61,54%	38,46%
D3	6	47	56	9,33	1,19	19,86%	80,14%

The three datasets were created, including users' feedback for different scenarios modelled according to a specific user bot defined with a different trust rating profile for each user. The datasets were constituted in a way that enabled the evaluation of the different approaches on different sparsity levels and several cold-start users and scenarios. In order to attain different sparsity levels, it was necessary to establish datasets with different numbers of users, scenarios and ratings. The initial dataset had 540 scenarios, which, at the time of recommendation, would become computationally heavy and time-consuming, making it necessary to apply scenario reduction techniques based on the simulation results. Since it is an industrial environment, the presence of few users is a recurrent variable in these systems, which is necessary for a RS capable of working with a small dataset with few users and rating information.

The density level is the ratio between the number of actual ratings (Act_R) and the number of possible ratings (Pos_R) that can be calculated by multiplying the number of users by the number of scenarios. The Equation 5.1 calculates the density of a dataset.

$$Density = \frac{Act_R}{Pos_R} \times 100 \tag{5.1}$$

The sparsity level can be calculated based on the density, as the sum of the two has to be 100%. The Equation 5.2 calculates the sparsity level of the dataset.

$$Sparsity = 100 - Density \tag{5.2}$$

The experimental validation metrics that are usually used to evaluate the predictive rating accuracy for a RS approach are the MAE and RMSE, defined in Subsection 4.2.2.

5.2.1 Comparison between *SimQL* and *QL* Algorithms

A simple version of the *SimQL* recommendation approach, hereafter called *QL*, was also implemented to serve as a comparison. The *QL* algorithm only considers the Q-learning algorithm reporting only on the user trust rate and acceptability, and it does not consider any similarity measures or different predicting rating calculation equations. This model was defined as an intermediate approach highlighting the benefits of applying similarity and trust measures to address the cold-start and data sparsity problems.

In order to verify what are the actual performance differences between the two approaches, *SimQL* and *QL*, a study was conducted comparing the two approaches. Considering that the proposed approach is an iterative method, an experiment was performed for which the dataset ran

continuously, performing recommendations for one user at a time and observing the performance of the models in terms of rating prediction accuracy. This study allowed to compare how the integration of the similarity and trust measures and the dynamic predicting of trust rating changes the performance of the proposed algorithm. Figure 5.9 illustrates the results of the RMSE for the *SimQL* and *QL* models in twenty-five iterations and considering the dataset D1, with the lower level of sparsity and the lower number of cold-start users and scenarios.

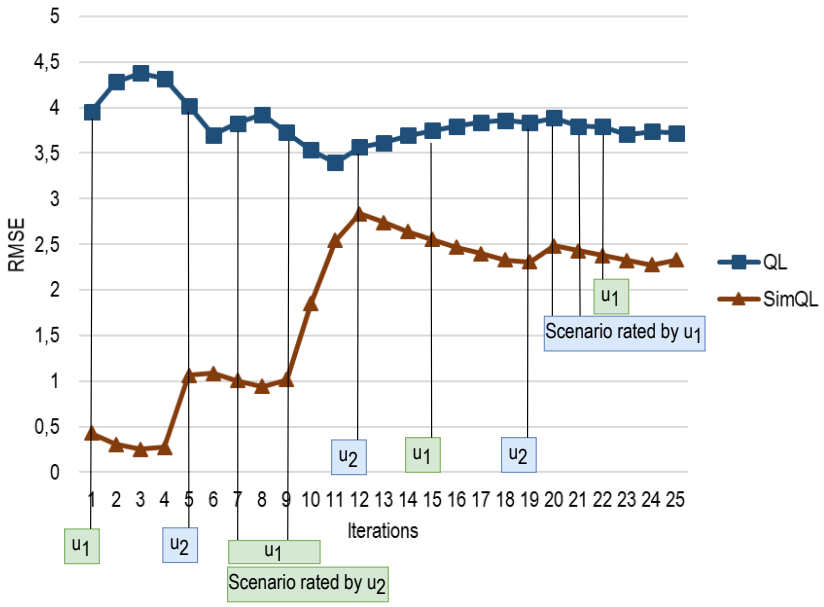

Figure 5.9: Comparison of the performance of *SimQL* with *QL* for dataset D1.

In the RMSE graph for dataset D1, it is possible to observe a significant increase in the RMSE value at the fifth and from the ninth to the twelfth iteration. In the fifth iteration, this is due to the first change of user, introducing a cold-start user with no previous rating history and starting to rate never-rated scenarios. The system uses the scenario similarity measure, but the scenario may be too different, influencing a less accurate prediction. The first scenario, already rated by another user, is introduced in the ninth iteration. The user similarity is calculated, and since there is little historical information for both users, similarity calculation may not be very accurate, changing the following predictions based on this result, which can be classified as an outlier. The significant increase in the RMSE measure may be due to its susceptibility to outliers. In the twentieth iteration, there is again a user similarity calculation, but now with more historical information, which translates into an insignificant increase in the RMSE.

Figure 5.10 illustrates the results of the RMSE for the *SimQL* and *QL* models in twenty-five iterations and considering the dataset D3, with the higher level of sparsity and the higher number of cold-start users and scenarios.

In the dataset D3 graph, the introduction of the users is performed until the twelfth iteration, which means that the algorithm will have more basic information to perform the recommendations. This increases the RMSE from the third to the sixth iteration, possibly due to new scenarios that

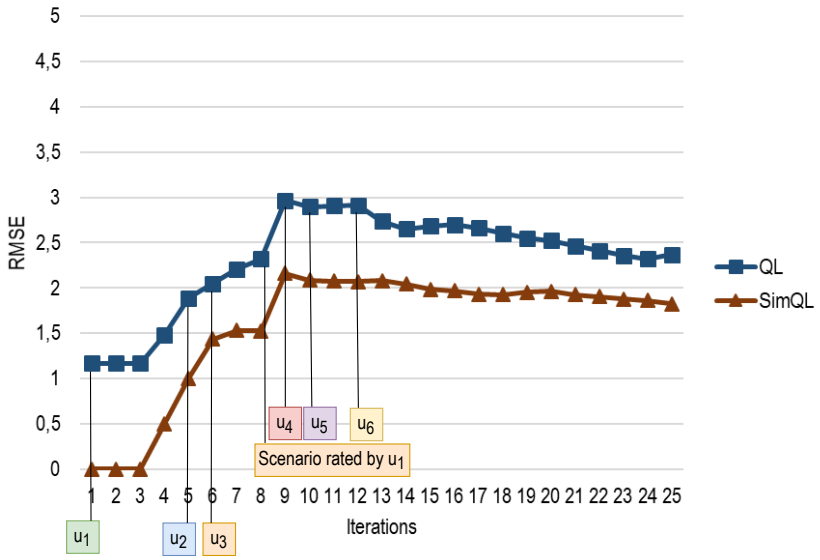

Figure 5.10: Comparison of the performance of *SimQL* with *QL* for dataset D3.

are very different from those with historic rating information. In the eighth iteration, the *User 3* is rating a scenario already rated by *User 1*, applying the user similarity and significantly increasing the RMSE.

For the dataset D1, the RMSE of the *QL* model is significantly higher than the *SimQL* model, which means that the *SimQL* model performs better than the *QL*. Considering the performance of both models, the *SimQL* outperforms the *QL* model in the two situations analysed. Implementing the similarity, reputation and trust measures within a RL algorithm for recommendation significantly contributes to handling cold-start users/scenarios and data sparsity problems. Particularly, the *SimQL* model, on average, performs better in a dataset with more users and scenarios and can handle the data sparsity problem without sacrificing performance.

Regarding datasets D1 and D3, there are significant differences in their constitution, with dataset D1 having only two users and dataset D3 having six users; the number of scenarios to be evaluated is also different, being 23 and 47, respectively, and consequently, the sparsity levels are also quite different, 32% and 80%. These differences are also noted in the evolution of the RMSE over the iterations, and for the dataset D3, the values of the *SimQL* model stabilised earlier. Although dataset D3 has more users, which means it is more likely to be subjected to cold-start users, and the sparsity level is higher, the fact that all users are introduced early in the recommendation cycle until the twelfth iteration makes the system more stable and on a path of continuous improvement.

5.2.2 Comparison of *SimQL* with State-of-the-Art Approaches

The *SimQL* was also compared with the state-of-the-art recommendation approaches, namely two merely based on ratings models, *UserCF* (Resnick et al., 1994) and *SVD++* (Koren, 2010), three early TBR models, *SocialRec* (Ma et al., 2008), *SocialRSTE* (Ma et al., 2009), *SocialReg* (Ma

et al., 2011), and three latest state-of-the-art TBR models, *SocialMF* (Jamali and Ester, 2010), *TrustWalker* (Jamali and Ester, 2009), and *TrustSVD* (Guo et al., 2015). These methods are already mentioned in Subsection 2.3.2, but a summary overview will be provided as a reminder. *UserCF*, was one of the first CF models, being based on the user ratings on the users' similarity of preferences (Resnick et al., 1994). Based on the CF approach is the *SVD++*, being classified as a latent factor model, which bases its recommendations on the matrix factorisation, including a set of factors that model the item-item relations, and the users' implicit feedback (Koren, 2010). The *SocialRec* integrates the concept of social regularisation into a matrix factorisation model, using a user-feature matrix factorised by ratings and trust (Ma et al., 2008). The *SocialRSTE* proposes a social trust ensemble method to linearly combine a basic matrix factorisation model and a trust-based neighbourhood model (Ma et al., 2009). *SocialReg* uses a user-specific vector to calculate the average of their trusted users. This average is then used to create a new matrix factorisation model that leverages social relationships between users to improve its performance (Ma et al., 2011). The *SocialMF* is based on the principles of the *SocialRec*, reformulating the use of trusted users to form the active user's user-specific vector and enabling the trust propagation property. In this model, a user's features depend on the features of its direct neighbours, and recursively, the features of the direct neighbours are also dependent on its direct neighbours. This method combines matrix factorisation with trust propagation to produce recommendations (Jamali and Ester, 2010). The *TrustSVD* model is an extension of the *SVD++* model that includes a trust-based matrix factorisation technique, which uses rating explicit and implicit feedback, and the explicit and implicit user social trust data. The model was adapted with a weighted regularisation to regularise the latent feature vectors of the user and items (Guo et al., 2015). Lastly, the *TrustWalker* model is based on a random walk model that combines an item-based ranking method and a trust-based nearest neighbour model. The model considers the ratings of the target item and of the similar items, the probability of using the rating of the similar item is directly affected by the length of the walk. With the *TrustWalker*, it is possible to calculate the confidence of the made predictions (Jamali and Ester, 2009). These models were implemented following the implementation provided by Zhang et al. (2018), using an Intel Core M-5y71 1.20 GHz CPU with 8 GB RAM to run all the approaches on a Windows 10 Pro system.

Table 5.4 summarises the achieved MAE and RMSE results for the experimental tests considering the different approaches and the three datasets. Note that the *Deviation* parameter indicates the improvement of the performance of the *SimQL* model relative to the analysed model, which is calculated through Equation 5.3.

$$Deviation = \frac{V_M - V_{SimQL}}{V_M} \times 100 \tag{5.3}$$

where the V_{SimQL} represents the RMSE or MAE value of the *SimQL* model, and V_M is the value for the model to be compared.

Considering the results presented in Table 5.4, each dataset's best and worst models are identified with the RMSE values in bold. Regarding the state-of-the-art models and the recommendation

Table 5.4: Comparison of the performance of *SimQL* model with the state-of-the-art recommendation models.

	Model	Dataset D1		Dataset D2		Dataset D3	
		RMSE Deviation	**MAE Deviation**	**RMSE Deviation**	**MAE Deviation**	**RMSE Deviation**	**MAE Deviation**
Proposed Model	**SimQL**	**1,523**	**1,004**	**1,734**	**1,295**	**1,562**	**1,367**
	QL	3,994 61,87%	3,536 63,36%	3,522 51,14%	3,088 58,05%	3,328 53,05%	2,876 52,48%
Traditional	UserCF	3,569 57,33%	2,989 56,66%	2,565 32,90%	1,932 32,96%	2,240 30,24%	1,639 16,63%
	SVD++	**3,663** **58,42%**	3,117 56,14%	2,734 37,04%	2,238 42,12%	2,474 36,85%	2,066 33,87%
Social	SocialRec	3,031 49,76%	2,504 48,27%	**2,469** **30,28%**	1,939 33,19%	2,056 24,00%	1,603 14,74%
	SocialRSTE	3,057 50,18%	2,724 52,45%	2,882 40,28%	2,574 49,68%	**2,574** **39,30%**	2,275 39,93%
	SocialMF	3,637 58,13%	3,123 58,53%	**2,917** **41,00%**	2,468 47,52%	2,502 37,54%	2,118 35,49%
	SocialReg	3,191 52,27%	2,697 51,97%	2,695 36,13%	2,200 41,12%	2,212 29,38%	1,814 24,65%
	TrustWalker	**2,453** **37,92%**	2,113 38,70%	2,695 36,13%	2,298 43,62%	2,484 37,10%	2,103 35,00%
	TrustSVD	2,821 46,02%	2,428 46,64%	2,560 32,76%	2,067 37,35%	**1,918** **18,54%**	1,498 8,80%

accuracy, for the dataset D1, the *TrustWalker* was the best-performing model with an RMSE 2,453. This method uses the rating prediction of the user-scenario rating information and the user trust data from which a trust network is built. One of the reasons this model is the best-performing model is that it performs simulated random walks on the trust network for each user, collecting information about the direct and indirect relationships between users. The use of trust measures in RSs has a direct correlation with improvement in terms of performance. In this case, dataset D1 presents only two users, making the trust network less extensive, consequently improving the model's accuracy since the trust network loses accuracy with the increase in size. In the dataset D2, the best-performing model was *SocialRec*, with an RMSE of 2,469. One of the main reasons for this is that the model uses social influence in the recommendation process, considering the users' individual and friends' preferences. This is achieved by fusing the user-rating matrix with the user's social network using a probabilistic matrix factorisation. The *TrustSVD* model was the best-performing method for the dataset D3, with an RMSE of 1,918. Although the implicit information about the user-scenario rating is not present in this dataset or the other two datasets, this method outperforms the other methods. Despite the lack of explicit and implicit information by the *TrustSVD* model, this method also uses a weighted regularisation technique to avoid the over-fitting of the model learning and the user trust for the rating prediction, and it uses the trust network of users considering only the direct relationships between the users. Although these methods are all social recommendation models, there are fundamental differences between them and how they perform recommendations, which results in different best-performing models for

each dataset with different conditions.

Regarding worst methods, in the dataset D1, the *SVD++* model was the worst performing method, with an RMSE of 3,663. For this model to work well, it requires a large dataset with a large set of implicit user-scenario rating information since this information is the basic information used for the rating prediction calculation in this model. This model works better under sparse data, which does not happen in this dataset (sparsity = 32,61%). For the dataset D2, the worst method was the *SocialMF* with an RMSE of 2,917, which uses the matrix factorisation-based model for recommendation in social rating networks, incorporating trust propagation. This model needs relevant and a large amount of social information to efficiently incorporate it into the recommendations, which may be difficult in a dataset with three users. One of the main characteristics of the method is its ability to reduce the RMSE significantly for cold-start users, which in the dataset D2 is not present since there is a rate of 61,54% of dataset density. In the dataset D3, the worst performing method is the *SocialRSTE* with an RMSE of 2,574, which uses a factorised user-scenario rating matrix and the social trust network then applied in a probabilistic framework and gradient descent objective function. One possible reason this method is the worst is that it performs best in very large datasets, the approach scales linearly with the number of observations, and the provided dataset is of a small dimension. The quality and quantity of the user-item interactions and available social information can influence the recommendation models' performance.

In a high-level analysis, on average, the social recommendation models perform better than the traditional recommendation methods. The differences in performance start to decrease with the increase in the dataset size, considering the number of users, scenarios and ratings. From the results presented in the table, the proposed *SimQL* consistently outperforms the tested state-of-the-art methods, presenting a stable performance throughout the three datasets in terms of RMSE and MAE. For the dataset D1, the model improves 37,92% regarding the *TrustWalker*, for the dataset D2 improves 30,28% relative to *SocialRec*, and for the dataset D3 improves 18,54% regarding the *TrustSVD*. In summary, the *SimQL* model presents a high percentage of performance improvement in all datasets and can alleviate data sparsity problems and cold-start users/scenarios. The main difference between the *SimQL* model and the other social recommendation models is the combination of a RL algorithm with similarity measures, social trust data for performing recommendations, and a dynamic system for rating prediction calculation.

Considering the *QL* approach, from a general perspective and compared with the state-of-the-art methods, the results show that the *QL* approach is the worst performing method for the three datasets. This method only considers the user-item rating and the user-acceptability information, which limits the method's performance in rating prediction. Since the method is based on RL, its performance is also limited by the definition of the parameters, such as the reward function, learning rate, and discount factor for future rewards. The model's performance is also dependent on the quality of the interaction with the user since this is a sequential model dependent on the user interaction.

Taking everything into consideration all the presented results, there are several advantages of applying the *SimQL* approach to the specified case study, namely a faster identification of the

best scenarios allowing for a near real-time intervention, a selection of the best scenarios not only based on the simulation results but also on the human knowledge of the system, and the capability of handling problems as cold-start and data sparsity which are recurrent in recommendation environments.

To summarise, the *SimQL* model outperforms the *QL* model and the other state-of-the-art models, having the capability to handle problems such as cold-start users and data sparsity. These preliminary results reveal that the combination of similarity, reputation measures, and trust measures with a learning algorithm can deal with the most recurrent problems of traditional approaches.

5.3 Sensitivity Analysis of the RS Parameters

Sensitivity analysis is usually performed to understand how the changes in the input parameters of a system can affect the system's output. Considering the proposed RS model, a sensitivity analysis was performed to evaluate the impact of variations in input parameters on the performance and robustness of the model, which could contribute to the improvement of the accuracy and quality of the recommendations made by the RS (Maida and Obwegeser, 2012). The acceptance and adoption of the RS depends on the quality and accuracy of the recommendations produced by the system. In this case, a fuzzy logic approach was proposed to determine the optimal operating conditions and the most influential parameters for the proposed model regarding the recommendation quality.

In the case of the RS field, the definition of the parameters involved in developing a RS can have a high level of impreciseness and uncertainty since it can be domain-dependent. For example, the vagueness of what can be considered a small or large number of users for a system is very dependent on the application domain. In the case of the manufacturing domain, 100 users can be a large number of users, but for e-commerce, this is a small number of users. Considering the uncertainty level associated with the RS design, the Fuzzy Logic can assess the system, particularly focusing on the fuzzy sensitivity analysis (Jain and Gupta, 2018).

5.3.1 Definition of the Fuzzy System

The sensitivity analysis is going to examine the recommendation module's *RL algorithm*, *Similarity Measures*, and *Trust Measures* using a fuzzy logic approach, which has proven to be a flexible method suitable for both types of scope. The general fuzzy system illustrated in Figure 5.11 was combined into four systems to simplify the process and reduce the number of fuzzy rules and system complexity.

The system is going to evaluate how the design variables, like *Dataset Conditions*, which is the first fuzzy system (e.g., cold-start, data sparsity, normal conditions), *trust factor*, the second fuzzy system, (e.g., trust measures, user similarity, user reputation), and the *Learning factor*, the third fuzzy system, (e.g., learning rate, discount factor, and the number of iterations), impact the *Recommendation Quality* of the model outputs, the fourth fuzzy system. For each system, the input variables' fuzzy sets were considered to be established using trapezoidal Matrix Factorisation

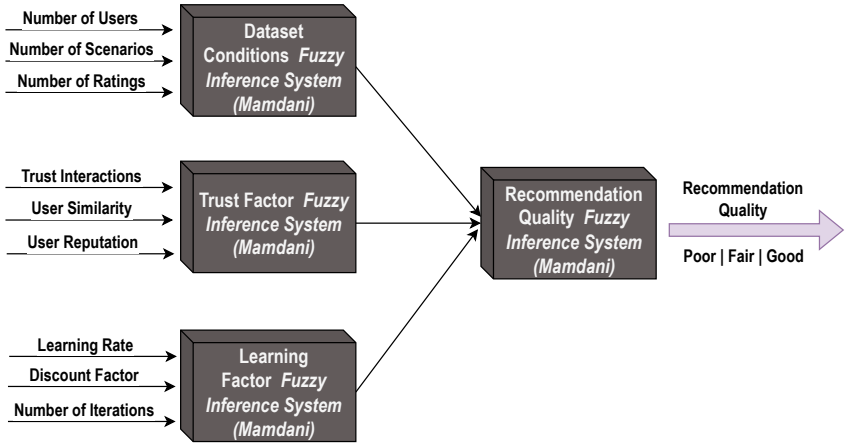

Figure 5.11: General fuzzy system, with input variables and the four inference systems.

(MF)s, and each input variable has a specified range. The variable range for the MFs was determined based on the RS experiment's performance and expected outcomes. The output variables of each fuzzy system were similarly defined, with values ranging from 0 to 1. Figure 5.12 illustrates the first fuzzy system to be analysed.

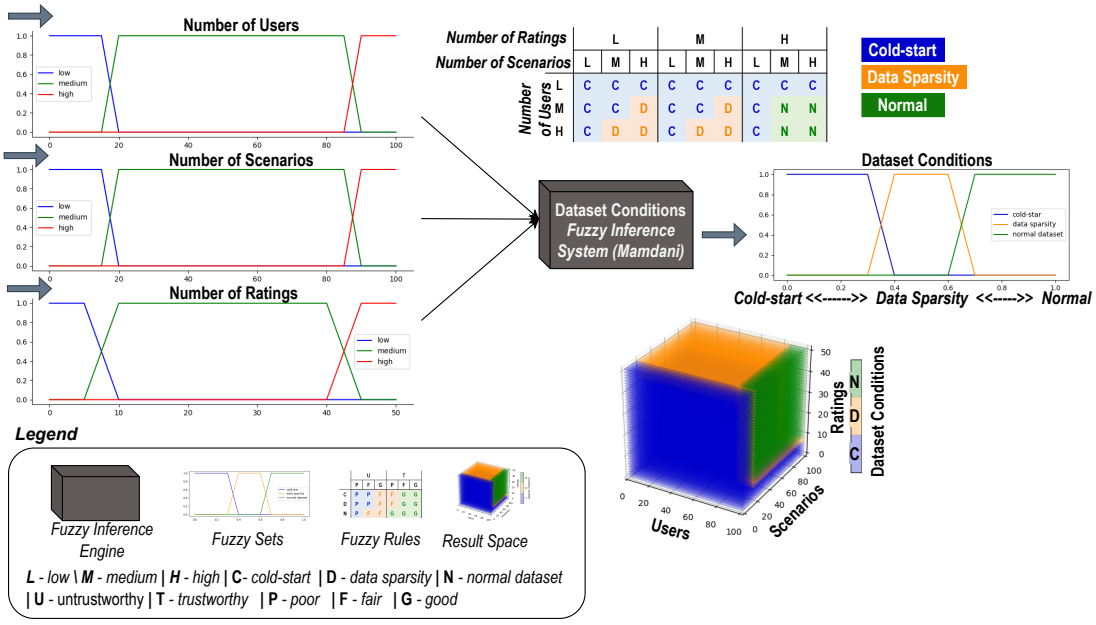

Figure 5.12: Fuzzy system for the *Dataset Conditions*: fuzzy sets, membership functions, fuzzy rules inference systems and solution space.

This system considers as input variables the *number of users*, *number of scenarios*, and *number of ratings*. For each of these input variables, a fuzzy set was defined based on the following linguistic terms: *low (L)*, *medium (M)*, and *high (H)*. In the case of the outputs of the system, these were the *dataset conditions* with which the RS is faced based on the input variables as *cold-start*, *data sparsity*, or *normal*. According to the established decision table, illustrated in Figure 5.12, a set of rules was established for the fuzzy inference system in the form of IF-THEN rules, mapping

the MFs of the input variables to the MFs of the output variables. Each cell of the decision table represents the result of an AND logical operation between the input variables.

IF number of users = *L* **AND** number of scenarios = *H* **AND** number of ratings = *L* **THEN** dataset conditions = *cold-start*

This set of defined rules leads to a resulting space illustrated as a 3D cube. In these graphs, the possible relationships between the variables and the consequent result can be observed.

Figure 5.13 illustrates the second fuzzy system related to the similarity and trust measures of the *SimQL* approach.

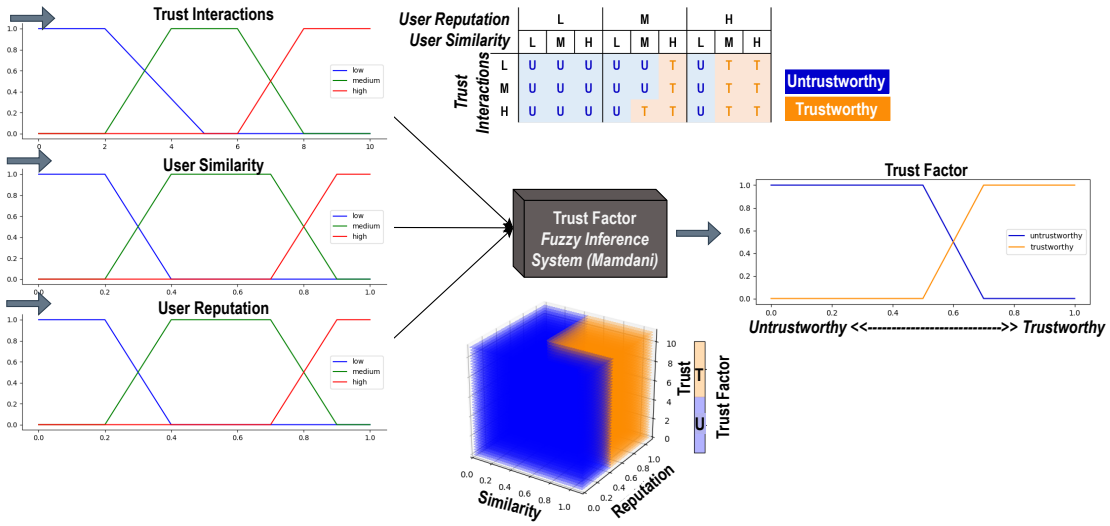

Figure 5.13: Fuzzy system for the *Trust Factor*: fuzzy sets, membership functions, fuzzy rules inference systems and solution space.

This system considers as input variables the *trust interactions*, *user similarity*, and *user reputation*, and these variables follow the linguistic terms established for the first system. In this case, the trust interactions are related to the user interactions in their social network, which are directly connected to the user reputation calculation. The output of this system will be a *trust factor*, which verifies if the measures used fall into the scope of *trustworthy* or *untrustworthy*.

IF trust interactions = *H* **AND** user similarity = *H* **AND** user reputation = *H* **THEN** trust factor = *trustworthy*

Following this, Figure 5.14 presents the third fuzzy system, which refers to the learning capabilities of the RL algorithm of the *SimQL* approach.

This system comprehends as input variables the *learning rate*, *discount factor*, and *number of iterations*, which are three of the main variables involved in the learning process of the algorithm. These variables follow the linguistic terms established for the other two systems. The output that results is a *learning factor* of the algorithm, translating into the learning capabilities of the system. These can be *good*, *fair* or *poor*.

IF learning rate = *M* **AND** discount factor = *H* **AND** number of interactions = *M* **THEN** learning factor = *good*.

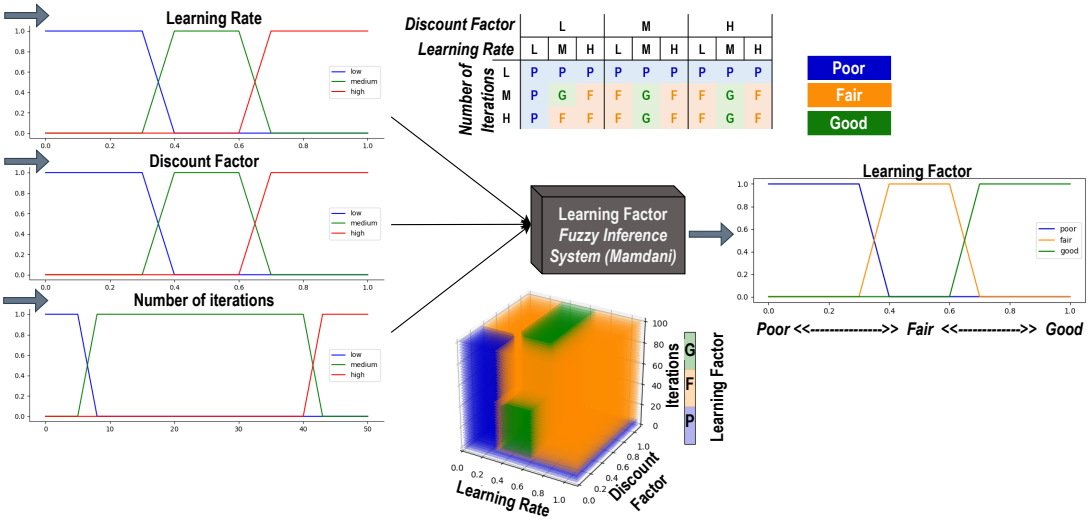

Figure 5.14: Fuzzy system for the *Learning Factor*: fuzzy sets, membership functions, fuzzy rules inference systems and solution space.

Finally, Figure 5.15 presents the fourth fuzzy system, which considers as inputs the outputs of the other three systems, focusing on the resulting recommendation quality.

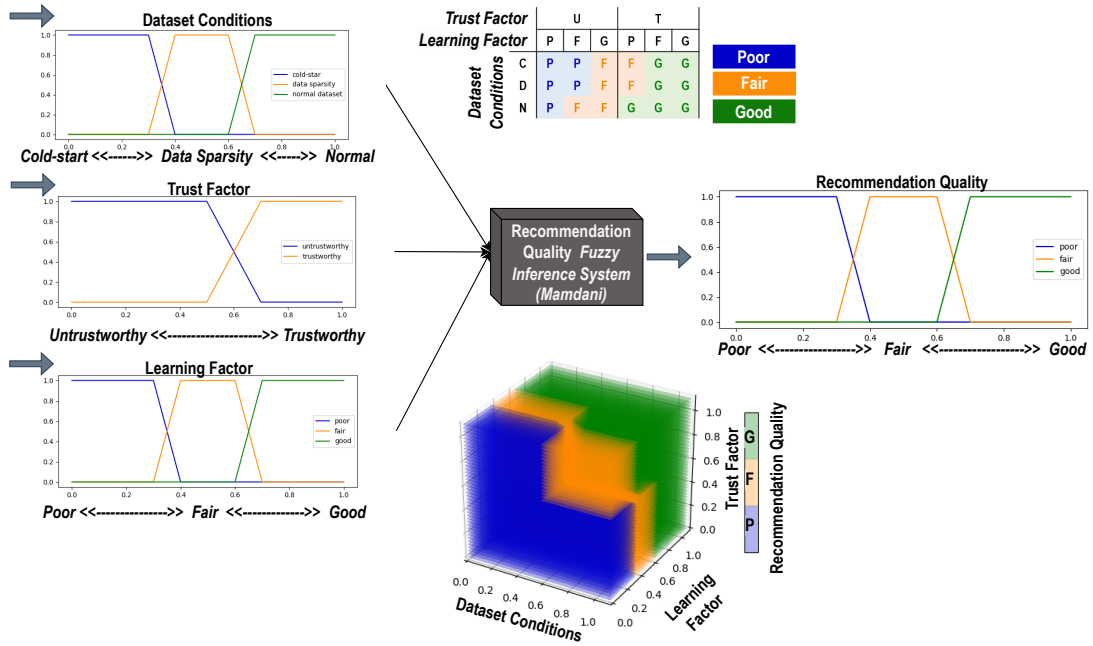

Figure 5.15: Fuzzy system for the *Recommendation Quality*: fuzzy sets, membership functions, fuzzy rules inference systems and solution space.

The fourth fuzzy system's output values determine the quality of the system's recommendations. The recommendation quality can be classified into three types based on the environment presented. For instance, an output of 0 indicates that the recommendation quality is *poor*, whereas an output of 0.5 indicates the recommendation quality is *fair*, and an output of 0.8 indicates that

the recommendation quality is *good*. Each *Recommendation Quality* level is defined according to Equation 5.4, Equation 5.5, Equation 5.6, and Equation 5.7, ranging from 0 to 1.

$$x = R(d,t,l) \tag{5.4}$$

$$O(poor) = \begin{cases} 1 & for\ x \leq 0.3 \\ \frac{0.4-x}{0.1} & for\ 0.3 \leq x \leq 0.4 \\ 0 & for\ x \geq 0.4 \end{cases} \tag{5.5}$$

$$O(fair) = \begin{cases} 0 & for\ x \leq 0.3\ and\ x \geq 0.7 \\ \frac{0.4-x}{0.1} & for\ 0.3 \leq x \leq 0.4 \\ \frac{0.7-x}{0.1} & for\ 0.6 \leq x \leq 0.7 \\ 1 & for\ 0.4 \leq x \leq 0.6 \end{cases} \tag{5.6}$$

$$O(good) = \begin{cases} 0 & for\ x \leq 0.6 \\ \frac{0.7-x}{0.1} & for\ 0.6 \leq x \leq 0.7 \\ 1 & for\ x \geq 0.7 \end{cases} \tag{5.7}$$

where, x is the crisp value of the *recommendation quality* fuzzy inference system, $R(d,t,l)$ is the *recommendation quality* fuzzy inference system with the input variables d,t,l being the *dataset conditions*, *trust factor* and *learning factor* respectively, and $O(quality)$ is the applicability of each given x for each given quality level $\in [Poor, Fair, Good]$.

The decision table is also illustrated in Figure 5.15, presenting the rules for the fuzzy inference system for the *Recommendation Quality*. Considering the previous examples of IF-THEN rules and the obtained results, in this case of recommendation quality, the one rule is as follows.

```
IF dataset conditions = cold-start AND learning factor = good AND
trust factor = trustworthy THEN recommendation quality = good
```

After analysing the resulting space for the recommendation quality system, it is possible to conclude that a good learning factor and a trustworthy trust factor enable a good-quality recommendation even if the dataset conditions are not ideal.

The author relied on acquired experience and knowledge to establish the configuration and definition of the MFs, fuzzy sets and fuzzy rules. The relationships between variables were set through empirical methods and computational experiments.

5.3.2 Validation of the Fuzzy System

The fuzzy sensitivity system was implemented in Python, using the library *skfuzzy*, which implements the Mamdani type fuzzy, having used the centroid method to perform the defuzzification process. To determine if the fuzzy sensitivity system accurately translates recommendation quality

and identifies the main variables that affect it, an experimental validation was performed regarding two experiments:

- ***Experiment FS1:*** the trust factor was kept in *trustworthy* values, and the learning factor (i.e., *good, fair, poor*) and dataset conditions (*.e., cold-start, data sparsity, normal*) were changed iteration after iteration.
- ***Experiment FS2:*** the trust factor was kept at *untrustworthy* values, and the learning factor and dataset conditions were changed as in ***Experiment FS1***.

Figure 5.16 presents the results obtained for the *Recommendation Quality* fuzzy inference system for ***Experiment FS1***, and Figure 5.17 presents the results obtained for the *Recommendation Quality* fuzzy inference system for ***Experiment FS2***.

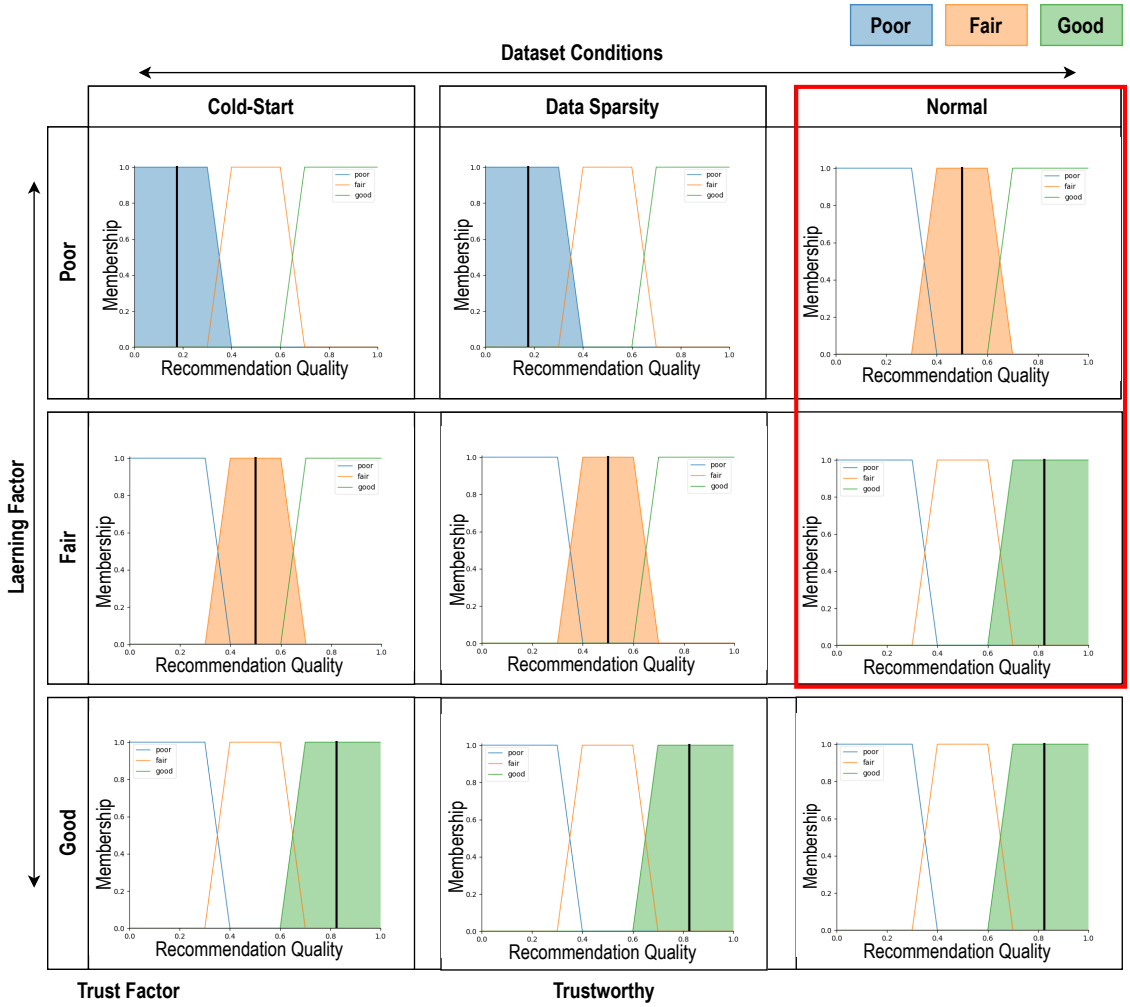

Figure 5.16: Results of the *Recommendation Quality* fuzzy inference system for ***Experiment FS1***.

After an in-depth analysis of the results, it was possible to verify that the fuzzy inference system performs according to the established rules. Additionally, trust and learning factors are two

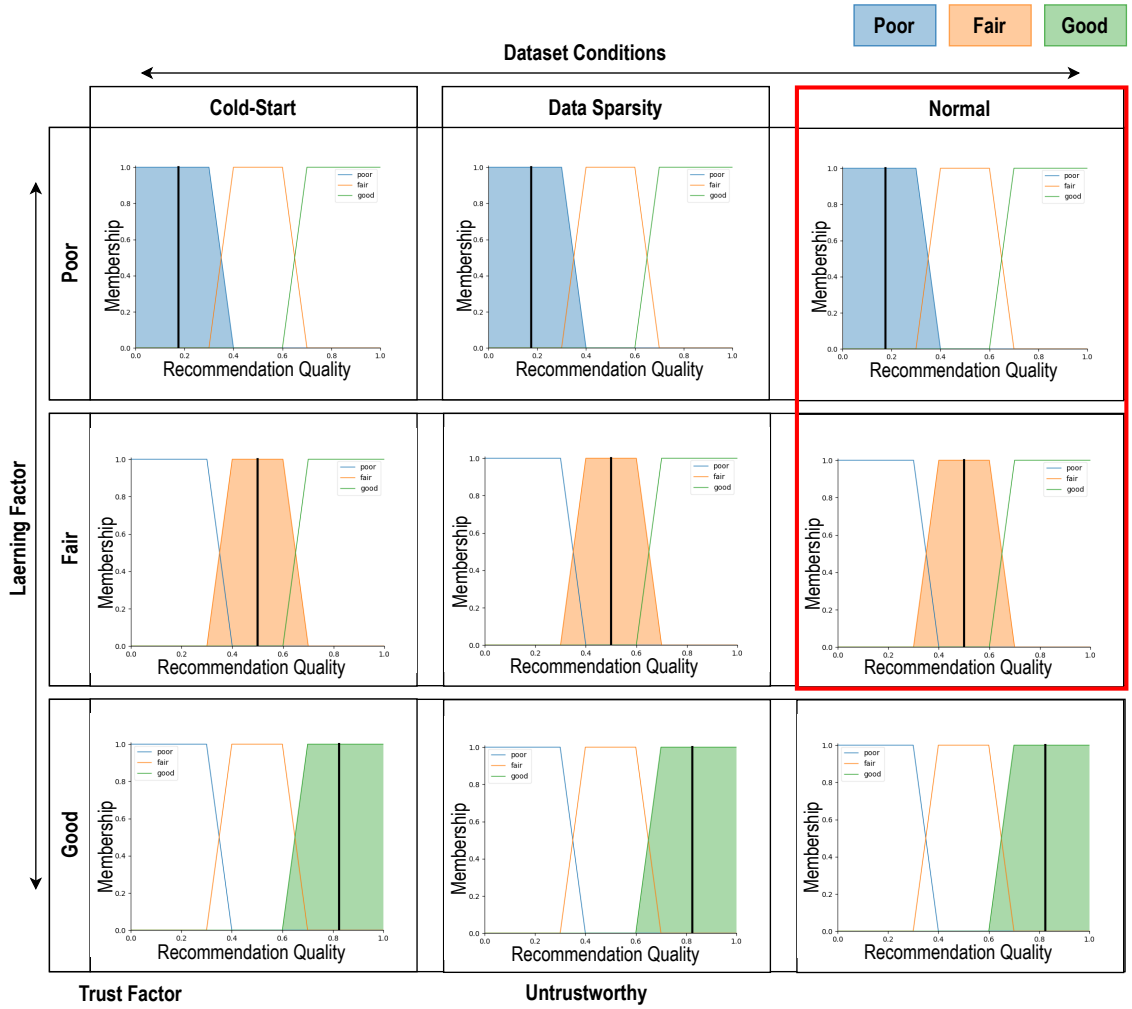

Figure 5.17: Results of the *Recommendation Quality* fuzzy inference system for ***Experiment FS2***.

of the three variables that most influence the recommendation quality of the system. This can be determined by the changes in the recommendation quality from ***Experiment FS1*** to ***Experiment FS2*** for the dataset conditions *normal* and for the learning factor *poor* and *fair*. For the ***Experiment FS2***, with a *untrustworthy* trust factor, the recommendation quality is *poor* and *fair*. By changing this factor to *trustworthy* (***Experiment FS1***), the recommendation quality improves by being *fair* and *good*, respectively. This fact shows that the trust data influences the quality of the recommendations of the *SimQL* model.

Regarding the learning factor, it is possible to observe that the influence of this variable is mostly on the recommendation quality of the cold-start and data sparsity cases since it is very difficult for the recommendation model to perform well with these dataset conditions. A *fair* and *good* learning factor produce recommendations with *fair* and *good* quality, even with an *untrustworthy* trust factor. This is due to the RL algorithm behind the recommendation model, enabling the learning process interactively.

The dataset conditions can also influence the recommendation quality since the most signifi-

cant changes in the recommendation quality from ***Experiment FS1*** to ***Experiment FS2*** are in the *normal* dataset conditions. The recommendation model can produce *fair* and *good* recommendation even with a *poor* and *fair* learning factor, respectively. This shows that a dataset with optimal operating conditions makes the recommendation model perform well, even at the expense of the other variables.

5.4 Summary

Through the presentation of the preliminary experiments, it was possible to validate each part of the recommendation approach individually, from the what-if simulation to the proposed recommendation algorithm, Q-Learning, and the application of the similarity and trust measures.

After the initial validation, the proposed recommendation approach *SimQL* was compared with its simpler version *QL* approach and with state-of-the-art approaches. According to the results, the *SimQL* was the best-performing approach compared to the tested methods. These results also demonstrated that the combination of an AI-algorithm with similarity and trust measures largely influences the *SimQL* performance since the *QL* approach is the worst performing approach.

With the results obtained from the performed sensitivity analysis through the use of a fuzzy logic system, it was possible to conclude that from the analysed parameters, the ones that most influence the recommendation quality of the model are the trust and learning factors.

Chapter 6

Conclusions and Future Work

Due to the rise of digitalisation technologies, the manufacturing industry has become more reliant on data and ICT. However, manufacturers are pressured to respond quickly to market demands, making the performance of traditional decision-making during production processes more difficult. Within the manufacturing industry, the decision-making process primarily depends on the decision-maker's analysis and knowledge. However, this approach can often be time-consuming and potentially inaccurate when faced with large quantities of data or when the decision-maker has no prior experience with the presented issue. In the given context, the Digital Twin concept offers an intelligent decision support platform by creating a virtual replica of a physical system, process or entity, integrating real-time monitoring, data analysis, and simulation to support decision-makers. By integrating a RS into the Digital Twin, it will be possible to amplify the benefits of both technologies, providing a more personalised, adaptive and efficient approach to decision support in dynamic and complex environments. However, one of the major problems in having a highly dynamic, flexible and complex environment is the lack of historical and initial data to perform recommendations, also known as the cold-start and data sparsity problem.

Considering this, this thesis proposes a Digital Twin architecture for decision support based on an innovative RS approach called *SimQL*. The approach integrates trust, similarity, and intelligence to minimise the effects of the cold-start and data sparsity problems when supporting the decision-making for new users or items.

This chapter presents a comprehensive dissertation overview, including final remarks and accomplished contributions. The proposed research questions have been answered based on the developed work, and the thesis statement has been proven. Finally, the research challenges that remain open or have emerged are discussed and presented as future work.

The remainder of this chapter is organised as follows:

- Section 6.1: presents the main contributions obtained throughout the development of the research work to achieve this thesis's main objectives, answer the defined research questions, and prove the proposed thesis statement.
- Section 6.2: discusses potential areas for future research, highlighting promising opportunities for improving and expanding the presented work.

6.1 Main Contributions

The research work presented in this document introduces two significant contributions to the field of decision support systems, namely a new recommendation approach, entitled *SimQL*, in which the main innovation lies in the incorporation of an AI-based algorithm with similarity and trust measures to tackle the challenges posed by cold-start users and items, as well as data sparsity problems. The second significant contribution involves the integration of a RS into a Digital Twin, which enables real-time monitoring, data analysis and what-if simulation to enhance the decision support, combining this knowledge with user trust rating (i.e., feedback). The results and main contributions attained throughout the developed work to achieve the objectives of this research and answer the previously established research questions (Section 1.2) will be discussed next.

RQ 1: *In which way the integration of AI-based algorithms and trust models can enhance the RS to improve personalised recommendations for flexible and dynamic manufacturing environments?*

The answer to this question was attained by defining a Digital Twin architecture capable of performing decision support through the integration of a RS model, as the *SimQL* trust-based recommendation model, based on a AI-based algorithm and trust model. In this case, a RL algorithm, more specifically the Q-Learning, with similarity and trust measures, enabled the generation of more accurate recommendations than the state-of-the-art recommendation models. This was measured and validated by calculating RMSE and MAE parameters in the IML case study, making it possible to validate in an offline manner the effectiveness and accuracy in the generation of recommendations.

RQ 2: *In which way do the similarity measures, focusing both on items and decision-makers, can accelerate the learning process in cold-start environments?*

Throughout the experimental phase, a comparison was conducted between the *SimQL* recommendations utilising similarity measures and those without. The results clearly demonstrate that integrating similarity measures resulted in faster and more efficient recommendations. For instance, when a new scenario is added to the system, i.e., cold-start scenario, scenario similarity measures are utilised to assess its similarity to other recommended scenarios. If the similarity score is high, the system recommends a new scenario to the user based on the most similar scenario. Using scenario similarity measures significantly accelerates the learning process of the recommendation algorithm for the case of cold-start scenarios.

The application of user similarity measures follows a similar principle, as the RS will fast-track the learning based on the similarity score between the users. For example, when a new user enters the system, i.e., a cold-start user, and if there are already other users in the system, it is possible to apply the user similarity measure. If the user similarity score is high enough, the recommendation algorithm will elaborate its recommendations on the cold-start user based on the information of the similar user. If there is a tie between similar users, the tiebreaker will be carried out based on the reputation of the users, which is calculated based on the trust between the users' social networks.

Integrating user and scenario similarity measures enables the mitigation of the cold-start challenge in terms of both perspectives, accelerating the learning of the recommendation algorithm.

Based on the answers to the initially established research questions, the outcomes of this study have made significant contributions towards archiving the objectives of the thesis. Furthermore, the findings have confirmed the hypothesis initially defined in Section 1.2.

Hypothesis: *Developing a Digital Twin-based architecture that integrates recommendation systems to enable personalised, interactive trust-based and intelligent decision support, capable of generating accurate, flexible, agile and reliable recommendations by mitigating the cold-start problem.*

This research has significantly contributed to several scientific publications in international conferences and journals. These IEEE-sponsored peer-reviewed conferences are indexed by either Scopus or Web of Science. The publications can be summarised as four conference papers, three book chapters, and two journal articles (both in journals with a quartile score of Q1). Additionally, two conference papers received awards for the best presentation and best paper.

6.2 Future Work

Remember that the research outlined in this thesis is not final, and there is always room for improvement through new research developments. Therefore, this section identifies some possible research directions for future work, extending the developed research work.

Online Testing and Scalability

The validation of the proposed approach performed in the research work resulted in a proof-of-concept, proving that the *SimQL* can perform recommendations offline, recurring to datasets. The next step in validating the proposed approach is the online validation and scalability of the method. This will enable the validation of the method under conditions more similar to the real world. Although it is a necessary step towards increasing the TRL level of the system, this raises several questions as it is required to collect information on human users in terms of recommendation feedback and social network information, being necessary to account for the privacy concerns, since gathering feedback for RS often involves collecting personal data. Another problem is user engagement, which is challenging to have users actively engage with the system and provide feedback.

Improve the Similarity Measures

In the proposed recommendation approach, the use of similarity measures is proposed as a mitigation strategy for the cold-start problem associated with elaborating recommendations to new users and scenarios. Although it was possible to validate the application of these measures, which improved the system's capacity to perform recommendations, there is still room to improve, for example, by implementing an adaptable similarity measure system. This system would have a

collection of different similarity measures (e.g., PCC, COS, JD, ED, among others), both for users and scenarios, and would be able to choose the best measure from the range of measures depending on the presented challenge, cold-start or data sparsity.

Development of an Explanation Engine

The integration of an explanation engine based on AI-algorithms capable of providing explanations on how the recommendation was performed helps users to understand from which insights resulted from the generated recommendation, which helps in the effectiveness, transparency, and trustworthiness of the RS. The development of an explanation engine can be the next step in the research work following this work, verifying how integrating a system like this influences the RS accuracy and the user's trust in the system and the generated recommendations.

Strategy Definition for the Assessment of Social Trust

Social trust is a way of incorporating trust relationships into the social network of a RS, which can improve the accuracy and fairness of the recommendations. The main challenges in this topic are the assessment, measurement, and application of social trust. None of these challenges were explored in this thesis, and it was assumed that the social trust values already existed. It could be of interest in the future to explore the existing methods of trust assessment to verify if they are and how efficient they are and try to develop a new unified trust assessment strategy and measure that would improve the quality of the recommendations.

Exploration of more advanced RL algorithms

Applying a RL algorithm in the context of recommendation systems was one of the main achievements of this work. In this case, for the development of the recommendation approach *SimQL*, a simple method was applied, Q-Learning. As future work, in order to perform significant improvements in the performance and in the scalability of the systems it should be considered the exploration of more powerful RL algorithms (e.g., Deep Q-Network).

Bibliography

"Digital Twin Consortium Defines Digital Twin," 2020. [Online]. Available: https://www.digitaltwinconsortium.org/2020/12/digital-twin-consortium-defines-digital-twin/

"Scopus now includes 90 million + content records!" 2023. [Online]. Available: https://blog.scopus.com/posts/scopus-now-includes-90-million-content-records

M. Aamir and M. Bhusry, "Recommendation System: State of the Art Approach," *International Journal of Computer Applications*, vol. 120, no. 12, pp. 25–32, 2015. [Online]. Available: https://doi.org/10.5120/21281-4200

A. Abdul-Raham and S. Hailes, "Distributed Trust Model," in *Proceedings of the 1997 Workshop on New Security Paradigms*, 1998, pp. 48–60. [Online]. Available: https://api.semanticscholar.org/CorpusID:18218821

A. Abdul-Rahman and S. Hailes, "Supporting trust in virtual communities," in *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, 2000, p. 9 vol.1. [Online]. Available: https://api.semanticscholar.org/CorpusID:1873238

G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734–749, 2005. [Online]. Available: https://doi.org/10.1109/TKDE.2005.99

M. M. Afsar, T. Crump, and B. Far, "Reinforcement Learning based Recommender Systems: A Survey," *ACM Computing Surveys*, vol. 55, no. 7, pp. 1–38, 2022. [Online]. Available: https://doi.org/10.1145/3543846

H. J. Ahn, "A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem," *Information Sciences*, vol. 178, no. 1, pp. 37–51, 2008. [Online]. Available: https://doi.org/10.1016/j.ins.2007.07.024

H. Ahuett-Garza and T. Kurfess, "A brief discussion on the trends of habilitating technologies for Industry 4.0 and Smart manufacturing," *Manufacturing Letters*, vol. 15, pp. 60–63, 2018. [Online]. Available: https://doi.org/10.1016/j.mfglet.2018.02.011

D. Anand and K. K. Bharadwaj, "Utilizing various sparsity measures for enhancing accuracy of collaborative recommender systems based on local and global similarities," *Expert Systems with Applications*, vol. 38, no. 5, pp. 5101–5109, 2011. [Online]. Available: http://doi.org/10.1016/j.eswa.2010.09.141

APRE and CDTI, "Guiding notes to use the TRL self-assessment tool," 2022. [Online]. Available: https://horizoneuropencpportal.eu/store/trl-assessment

S. Bag, S. K. Kumar, and M. K. Tiwari, "An efficient recommendation generation using relevant jaccard similarity," *Information Sciences*, vol. 483, pp. 53–64, 2019. [Online]. Available: https://doi.org/10.1016/j.ins.2019.01.023

I. Barjasteh, R. Forsati, D. Ross, A. H. Esfahanian, and H. Radha, "Cold-start recommendation with provable guarantees: A decoupled approach," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, pp. 1462–1474, 6 2016. [Online]. Available: https://doi.org/10.1109/TKDE.2016.2522422

Z. Batmaz, A. Yurekli, A. Bilge, and C. Kaleli, "A review on deep learning for recommender systems: challenges and remedies," *Artificial Intelligence Review*, vol. 52, pp. 1–37, 6 2019. [Online]. Available: https://doi.org/10.1007/s10462-018-9654-y

H. Bauer, C. Baur, G. Campole, K. George, G. Ghislanzoni, W. Huhn, D. Kayser, M. Löffler, A. Tschiesner, A. E. Zielke, J. Cattell, L. Giet, R. Kelly, S. Muschter, F. Soubien, D. Wee, M. Breunig, M. Dufour, J. Lim, J. Bromberger, K. P. Rath, B. Saß, and K. Jüngling, "Industry 4.0 How to navigate digitization of the manufacturing sector," McKinsey&Company, Tech. Rep., 2015. [Online]. Available: https://www.mckinsey.com/capabilities/operations/our-insights/industry-four-point-o-how-to-navigae-the-digitization-of-the-manufacturing-sector

J. Beel and S. Langer, "A comparison of offline evaluations, online evaluations, and user studies in the context of research paper recommender systems," in *International Conference on Theory and Practice of Digital Libraries*, 2014. [Online]. Available: https://doi.org/10.1007/978-3-319-24592-8_12

A.-J. Berre, N. Figay, C. Guglielmina, and D. R. Karlsen, *The ATHENA interoperability framework*, 2007. [Online]. Available: https://doi.org/10.1007/978-1-84628-858-6

A. M. Bisantz and Y. Seong, "Assessment of Operator Trust in and Utilization of Automated Decision Aids under Different Framing Conditions," *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 44, no. 1, pp. 5–8, 2000. [Online]. Available: https://doi.org/10.1177/154193120004400102

J. Bloem, M. van Doorn, S. Duivestein, D. Excoffier, R. Maas, and E. van Ommeren, "The Fourth Industrial Revolution Things to Tighten the Link Between IT and OT," Sogeti, Tech. Rep., 2014. [Online]. Available: https://www.sogeti.ie/globalassets/global/special/sogeti-things3en.pdf

J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender systems survey," *Knowledge-Based Systems*, vol. 46, pp. 109–132, 2013. [Online]. Available: https://doi.org/10.1016/j.knosys.2013.03.012

M. Bohanec and J. S. Institute, "What is decision support?" in *Proceedings of the 4th International Multi-conference Information Society*, 2001, pp. 86–89. [Online]. Available: https://www.researchgate.net/publication/248421433

S. Boschert and R. Rosen, "Digital Twin - The Simulation Aspect," in *Mechatronic Futures*, 2016, pp. 59–74. [Online]. Available: https://doi.org/10.1007/978-3-319-32156-1

D. M. Botín-Sanabria, S. Mihaita, R. E. Peimbert-García, M. A. Ramírez-Moreno, R. A. Ramírez-Mendoza, and J. d. J. Lozoya-Santos, "Digital Twin Technology Challenges and Applications: A Comprehensive Review," *Remote Sensing*, vol. 14, no. 6, pp. 1–25, 2022. [Online]. Available: https://doi.org/10.3390/rs14061335

R. Burke, "Knowledge-Based Recommender Systems," *Encyclopedia of library and information systems*, no. August, pp. 167–197, 2013.

E. Çano and M. Morisio, "Hybrid recommender systems: A systematic literature review," *Intelligent Data Analysis*, vol. 21, no. 6, pp. 1487–1524, 2017. [Online]. Available: https://doi.org/10.3233/IDA-163209

S. Chakraborty, S. Hoque, N. R. Jeem, M. C. Biswas, D. Bardhan, and E. Lobaton, "Fashion recommendation systems, models and methods: A review," *Informatics*, vol. 8, pp. 1–34, 2021. [Online]. Available: https://doi.org/10.3390/informatics8030049

J. C. Chaplin, G. Martinez-arellano, and A. Mazzoleni, "Digital Twins and Intelligent Decision Making," in *Digital Manufacturing for SMEs*, 2020, pp. 159–187. [Online]. Available: https://doi.org/10.17639/91WZ-CP62

C. C. Chen, Y. H. Wan, M. C. Chung, and Y. C. Sun, "An effective recommendation method for cold start new users using trust and distrust networks," *Information Sciences*, vol. 224, pp. 19–36, 2013. [Online]. Available: http://doi.org/10.1016/j.ins.2012.10.037

H. Chen, H. Sun, M. Cheng, and W. Yan, "A recommendation approach for rating prediction based on user interest and trust value," *Computational Intelligence and Neuroscience*, vol. 2021, pp. 1–9, 2021. [Online]. Available: https://doi.org/10.1155/2021/6677920

Z. Chen, L. Shen, F. Li, and D. You, "Your neighbors alleviate cold-start: On geographical neighborhood influence to collaborative web service qos prediction," *Knowledge-Based Systems*, vol. 138, pp. 188–201, 12 2017. [Online]. Available: https://doi.org/10.1016/j.knosys.2017.10.001

P.-a. Chirita, W. Nejdl, and C. Zamfir, "Preventing Shilling Attacks in Online Recommender Systems," in *Proceeding of the 7th annual ACM International workshoop on Weeb Information and Data Management*, 2005, pp. 64–74. [Online]. Available: https://doi.org/10.1145/1097047.1097061

E. Commission, "European Interoperability Framework," European Commission, Tech. Rep., 2017. [Online]. Available: https://doi.org/10.2799/78681

A. Corallo, V. Del Vecchio, M. Lezzi, and P. Morciano, "Shop floor digital twin in smart manufacturing: A systematic literature review," *Sustainability (Switzerland)*, vol. 13, no. 23, 2021. [Online]. Available: https://doi.org/10.3390/su132312987

N. Donthu, S. Kumar, D. Mukherjee, N. Pandey, and W. M. Lim, "How to conduct a bibliometric analysis: An overview and guidelines," *Journal of Business Research*, vol. 133, no. April, pp. 285–296, 2021. [Online]. Available: https://doi.org/10.1016/j.jbusres.2021.04.070

A. Elisa, C. Sotos, S. Vanhoof, W. van den Noortgate, and P. Onghena, "The transitivity misconception of pearson's correlation coefficient," *Statistics Education Research Journal*, vol. 8, no. 2, pp. 33–55, 2009. [Online]. Available: https://doi.org/10.52041/serj.v8i2.394

Z. Fayyaz, M. Ebrahimian, D. Nawara, A. Ibrahim, and R. Kashef, "Recommendation systems: Algorithms, challenges, metrics, and business opportunities," *Applied Sciences*, vol. 10, no. 21, p. 7748, 2020. [Online]. Available: https://doi.org/10.3390/app10217748

A. Felsberger, B. Oberegger, and G. Reiner, "A Review of Decision Support Systems for Manufacturing Systems," in *SamI40 workshop at i-KNOW'16*, 2016. [Online]. Available: https://api.semanticscholar.org/CorpusID:7416287

F. Fkih, "Similarity measures for collaborative filtering-based recommender systems: Review and experimental comparison," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, pp. 7645–7669, 2022. [Online]. Available: https://doi.org/10.1016/j.jksuci.2021.09.014

K. K. Fletcher, "A method for dealing with data sparsity and cold-start limitations in service recommendation using personalized preferences," in *2017 IEEE International Conference on Cognitive Computing (ICCC)*, 2017, pp. 72–79. [Online]. Available: https://doi.org/10.1109/IEEE.ICCC.2017.17

A. Forcina and D. Falcone, "The role of Industry 4.0 enabling technologies for safety management: A systematic literature review," in *International Conference on Industry 4.0 and Smart Manufacturing*, vol. 180. Elsevier B.V., 2021, pp. 436–445. [Online]. Available: https://doi.org/10.1016/j.procs.2021.01.260

F. Franke, S. Franke, and R. Riedel, "AI-based Improvement of Decision-makers' Knowledge in Production Planning and Control," *IFAC-PapersOnLine*, vol. 55, no. 10, pp. 2240–2245, 2022. [Online]. Available: https://doi.org/10.1016/j.ifacol.2022.10.041

S. Frémal and F. Lecron, "Weighting strategies for a recommender system using item clustering based on genres," *Expert Systems with Applications*, vol. 77, pp. 105–113, 2017. [Online]. Available: https://doi.org/10.1016/j.eswa.2017.01.031

G. Gabrani, S. Sabharwal, and V. K. Singh, "Artificial intelligence based recommender systems: A survey," in *Communications in Computer and Information Science*, vol. 721. Springer Verlag, 2017, pp. 50–59. [Online]. Available: https://doi.org/10.1007/978-981-10-5427-3_6

J. Gaillard, "Recommender Systems: Dynamic Adaptation and Argumentation," Ph.D. dissertation, Universite D'Avignonetdes Pays Devaucluse, 2014. [Online]. Available: https://theses.hal.science/tel-01168476c

M. Geravanchizadeh and H. Roushan, "Dynamic selective auditory attention detection using rnn and reinforcement learning," *Scientific Reports*, vol. 11, pp. 1–12, 2021. [Online]. Available: https://doi.org/10.1038/s41598-021-94876-0

A. Gharahighehi, K. Pliakos, and C. Vens, "Addressing the cold-start problem in collaborative filtering through positive-unlabeled learning and multi-target prediction," *IEEE Access*, vol. 10, pp. 117 189–117 198, 2022. [Online]. Available: https://doi.org/10.1109/ACCESS.2022.3219071

M. Ghobakhloo, "The future of manufacturing industry: a strategic roadmap toward Industry 4.0," *Journal of Manufacturing Technology Management*, vol. 29, no. 6, pp. 910–936, 2018. [Online]. Available: https://doi.org/10.1108/JMTM-02-2018-0057

E. H. Glaessgen and D. S. Stargel, "The digital twin paradigm for future NASA and U.S. Air force vehicles," in *53rd Structures, Structural Dynamics and Materials Conference*, 2012, pp. 1–14. [Online]. Available: https://doi.org/10.2514/6.2012-1818

B. T. Gockel, A. W. Tudor, M. D. Brandyberry, R. C. Penmetsa, and E. J. Tuegel, "Challenges with structural life forecasting using realistic mission profiles," in *Collection of Technical Papers - AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, no. April 2012, 2012. [Online]. Available: https://doi.org/10.2514/6.2012-1813

J. Golbeck and J. Hendler, "Inferring binary trust relationships in Web-based social networks," *ACM Transactions on Internet Technology*, vol. 6, no. 4, pp. 497–529, 2006.

J. A. Golbeck, "Computing and applying trust in web-based social networks," Ph.D. dissertation, University of Maryland at College Park, 2005.

D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, "Using collaborative filtering to weave an information tapestry," *Communications of the ACM*, vol. 35, no. 12, pp. 61–70, 1992. [Online]. Available: https://doi.org/10.1145/138859.138867

M. Golfarelli and S. Rizzi, "What-if Simulation Modeling in Business Intelligence," *International Journal of Data Warehousing and Mining*, no. October, 2009. [Online]. Available: https://doi.org/10.4018/jdwm.2009080702

M. Grieves and J. Vickers, *Digital Twin: Mitigating unpredictable, undesirable emergent behavior in complex systems*. Springer, 2017. [Online]. Available: https://doi.org/10.1007/978-3-319-38756-7

G. Guo, "Resolving data sparsity and cold start in recommender systems," in *(eds) User Modeling, Adaptation, and Personalization*, J. Masthoff, B. Mobasher, M. C. Desmarais, and R. Nkambou, Eds., vol. 7379. Springer Berlin Heidelberg, 2012, pp. 361–364. [Online]. Available: https://doi.org/10.1007/978-3-642-31454-4_36

G. Guo, "Integrating trust and similarity to ameliorate the data sparsity and cold start for recommender systems," in *Proceedings of the 7th ACM Conference on Recommender Systems*, 2013, pp. 451–454. [Online]. Available: https://doi.org/10.1145/2507157.2508071

G. Guo, J. Zhang, and N. Yorke-Smith, "A novel bayesian similarity measure for recommender systems," in *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*. AAAI Press, 2013, p. 2619–2625.

G. Guo, J. Zhang, and D. Thalmann, "Merging trust in collaborative filtering to alleviate data sparsity and cold start," *Knowledge-Based Systems*, vol. 57, pp. 57–68, 2 2014. [Online]. Available: https://doi.org/10.1016/j.knosys.2013.12.007

G. Guo, J. Zhang, D. Thalmann, A. Basu, and N. Yorke-Smith, "From ratings to trust: An empirical study of implicit trust in recommender systems," in *Proceedings of the ACM Symposium on Applied Computing*. Association for Computing Machinery, 2014, pp. 248–253. [Online]. Available: https://doi.org/10.1145/2554850.2554878

G. Guo, J. Zhang, and N. Yorke-Smith, "Trustsvd: Collaborative filtering with both the explicit and implicit influence of user trust and of item ratings," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 29, 2015, pp. 123–129. [Online]. Available: https://doi.org/10.1609/aaai.v29i1.9153

L. Guo, J. Liang, Y. Zhu, Y. Luo, L. Sun, and X. Zheng, "Collaborative filtering recommendation based on trust and emotion," *Journal of Intelligent Information Systems*, vol. 53, no. 1, pp. 113–135, 2019. [Online]. Available: https://doi.org/10.1007/s10844-018-0517-4

S. Gupta and M. Dave, "An overview of recommendation system: Methods and techniques," in *Advances in Computing and Intelligent Systems. Algorithms for Intelligent Systems.*, H. Sharma, K. Govindan, R. Poonia, S. Kumar, and W. El-Medany, Eds. Springer, 2020. [Online]. Available: https://doi.org/10.1007/978-981-15-0222-4_20

S. Gupta and S. Nagpal, "An empirical analysis of implicit trust metrics in recommender systems," in *2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2015, pp. 636–639. [Online]. Available: https://doi.org/10.1109/ICACCI.2015.7275681

M. K. Habib and C. Chimsom I., "Industry 4.0: Sustainability and Design Principles," in *2019 20th International Conference on Research and Education in Mechatronics (REM)*, 2019, pp. 1–8. [Online]. Available: https://doi.org/10.1109/REM.2019.8744120

C. A. Haydar, "Les systèmes de recommendation à base de confiance," Ph.D. dissertation, Université Lorraine, 2014. [Online]. Available: https://hal.univ-lorraine.fr/tel-01751172/document

J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Trans. Inf. Syst.*, vol. 22, no. 1, p. 5–53, jan 2004. [Online]. Available: https://doi.org/10.1145/963770.963772

M. Hermann, T. Pentek, and B. Otto, "Design Principles for Industrie 4.0 Scenarios: A Literature Review," Business Engineering Institute St. Gallen, Tech. Rep. 1, 2015. [Online]. Available: https://doi.org/10.13140/RG.2.2.29269.22248

F. Hu, "Three-segment similarity measure model for collaborative filtering," in *Data Mining and Big Data*, Y. Tan, Y. Shi, and Q. Tang, Eds. Cham: Springer International Publishing, 2018, pp. 138–148. [Online]. Available: https://doi.org/10.1007/978-3-319-93803-5_13

L. Huang, M. Fu, F. Li, H. Qu, Y. Liu, and W. Chen, "A deep reinforcement learning based long-term recommender system," *Knowledge-Based Systems*, vol. 213, p. 106706, 2021. [Online]. Available: https://doi.org/10.1016/j.knosys.2020.106706

C.-S. Hwang and Y.-P. Chen, "Using trust in collaborative filtering recommendation," in *New Trends in Applied Artificial Intelligence*, H. G. Okuno and M. Ali, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 1052–1060. [Online]. Available: https://doi.org/10.1007/978-3-540-73325-6_105

W. Y. Hwang and C. H. Jun, "Supervised learning-based collaborative filtering using market basket data for the cold-start problem," *Industrial Engineering and Management Systems*, vol. 13, pp. 421–431, 12 2014. [Online]. Available: https://doi.org/10.7232/iems.2014.13.4.421

F. O. Isinkaye, Y. O. Folajimi, and B. A. Ojokoh, "Recommendation systems: Principles, methods and evaluation," *Egyptian Informatics Journal*, vol. 16, no. 3, pp. 261–273, 2015. [Online]. Available: https://doi.org/10.1016/j.eij.2015.06.005

ISO 23247, "Automation systems and integration - Digital twin framework for manufacturing - Part 1: Overview and general principles," ISO, 2021. [Online]. Available: https://www.iso.org/standard/75066.html

A. Jain and C. Gupta, "Fuzzy logic in recommender systems," *Studies in Computational Intelligence*, vol. 749, pp. 255–273, 2018. [Online]. Available: https://doi.org/10.1007/978-3-319-71008-2_20

G. Jain and T. Mahara, "An efficient similarity measure to alleviate the cold-start problem," in *2019 15th International Conference on Information Processing: Internet of Things*, 2019, pp. 1–8. [Online]. Available: https://doi.org/10.1109/ICInPro47689.2019.9092250

M. Jamali and M. Ester, "TrustWalker: A random walk model for combining trust-based and item-based recommendation," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2009, pp. 397–405. [Online]. Available: https://doi.org/10.1145/1557019.1557067

M. Jamali and M. Ester, "A matrix factorization technique with trust propagation for recommendation in social networks," in *Proceedings of the 4th ACM Conference on Recommender Systems*, 2010, pp. 135–142. [Online]. Available: https://doi.org/10.1145/1864708.1864736

G. K. Jha, M. Gaur, P. Ranjan, and H. K. Thakur, "A survey on trustworthy model of recommender system," *International Journal of System Assurance Engineering and Management*, vol. 14, pp. 789–806, 7 2023. [Online]. Available: https://doi.org/10.1007/s13198-021-01085-z

K. Ji and H. Shen, "Addressing cold-start: Scalable recommendation with tags and keywords," *Knowledge-Based Systems*, vol. 83, pp. 42–50, 2015. [Online]. Available: https://doi.org/10.1016/j.knosys.2015.03.008

D. Jones, C. Snider, A. Nassehi, J. Yon, and B. Hicks, "Characterising the Digital Twin: A systematic literature review," *CIRP Journal of Manufacturing Science and Technology*, vol. 29, pp. 36–52, 2020. [Online]. Available: https://doi.org/10.1016/j.cirpj.2020.02.002

J. Jooa, S. Bangb, and G. Parka, "Implementation of a recommendation system using association rules and collaborative filtering," *Procedia Computer Science*, vol. 91, pp. 944–952, 2016, promoting Business Analytics and Quantitative Management of Technology: 4th International Conference on Information Technology and Quantitative Management (ITQM 2016). [Online]. Available: https://doi.org/10.1016/j.procs.2016.07.115

A. Josang, R. Ismail, and C. Boyd, "A Survey of trust and reputation systems for online service provision," *Decision Support Systems*, vol. 43, no. 2, pp. 618–644, 2007. [Online]. Available: https://doi.org/10.1016/j.dss.2005.05.019

H. Kagerman, W. Wahlster, and J. Helbig, "Securing the future of German manufacturing industry Recommendations for implementing the strategic initiative Industrie 4.0 Final Report of the Industrie 4.0 Working Group," ACATECH, Tech. Rep. April, 2013. [Online]. Available: https://doi.org/10.1109/REM.2019.8744120

G. Ke, H. L. Du, and Y. C. Chen, "Cross-platform dynamic goods recommendation system based on reinforcement learning and social networks," *Applied Soft Computing*, vol. 104, 6 2021. [Online]. Available: https://doi.org/10.1016/j.asoc.2021.107213

M. Khan, X. Wu, X. Xu, and W. Dou, "Big data challenges and opportunities in the hype of Industry 4.0," *IEEE International Conference on Communications*, pp. 1–6, 2017. [Online]. Available: https://doi.org/10.1109/ICC.2017.7996801

H. N. Kim, A. T. Ji, I. Ha, and G. S. Jo, "Collaborative filtering based on collaborative tagging for enhancing the quality of recommendation," *Electronic Commerce Research and Applications*, vol. 9, no. 1, pp. 73–83, 2010. [Online]. Available: https://doi.org/10.1016/j.elerap.2009.08.004

Y. Koren, "Factorization meets the neighborhood: a multifaceted collaborative filtering model," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2008, pp. 426–434. [Online]. Available: https://doi.org/10.1145/1401890.1401944

Y. Koren, "Factor in the neighbors: Scalable and accurate collaborative filtering," *ACM Transactions on Knowledge Discovery from Data*, vol. 4, pp. 1–24, 2010. [Online]. Available: https://doi.org/10.1145/1644873.1644874

M. Kunath and H. Winkler, "Integrating the Digital Twin of the manufacturing system into a decision support system for improving the order management process," in *51st CIRP Conference on Manufacturing Systems*, vol. 72. Elsevier B.V., 2018, pp. 225–231. [Online]. Available: https://doi.org/10.1016/j.procir.2018.03.192

N. Lathia, S. Hailes, and L. Capra, "Trust-based collaborative filtering," in *Trust Management II*, Y. Karabulut, J. Mitchell, P. Herrmann, and C. Jensen, Eds., 2008, pp. 119–134. [Online]. Available: https://doi.org/10.1007/978-0-387-09428-1_8

J. Lee, E. Lapira, S. Yang, and A. Kao, "Predictive manufacturing system - Trends of next-generation production systems," *Society of Manufacturing Engineers (SME)*, vol. 1, pp. 38–41, 2013. [Online]. Available: https://doi.org/10.3182/20130522-3-BR-4036.00107

P. Leitao, F. Pires, S. Karnouskos, and A. W. Colombo, "Quo Vadis Industry 4.0? Position, Trends, and Challenges," *IEEE Open Journal of the Industrial Electronics Society*, vol. 1, no. October, pp. 298–310, 2020. [Online]. Available: https://doi.org/10.1109/OJIES.2020.3031660

J. Leng, D. Wang, W. Shen, X. Li, Q. Liu, and X. Chen, "Digital twins-based smart manufacturing system design in Industry 4.0: A review," *Journal of Manufacturing Systems*, vol. 60, no. May, pp. 119–137, 2021. [Online]. Available: https://doi.org/10.1016/j.jmsy.2021.05.011

C. Li, Y. Chen, and Y. Shang, "A review of industrial big data for decision making in intelligent manufacturing," *Engineering Science and Technology, an International Journal*, vol. 29, p. 101021, 2022. [Online]. Available: https://doi.org/10.1016/j.jestch.2021.06.001

T. P. Liang, "Recommendation systems for decision support: An editorial introduction," *Decision Support Systems*, vol. 45, no. 3, pp. 385–386, 2008. [Online]. Available: https://doi.org/10.1016/j.dss.2007.05.003

Y. Lin, Y. Liu, F. Lin, L. Zou, P. Wu, W. Zeng, H. Chen, and C. Miao, "A survey on reinforcement learning for recommender systems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 14, pp. 1–21, 9 2021. [Online]. Available: http://doi.org/10.1109/TNNLS.2023.3280161

H. Liu, Z. Hu, A. Mian, H. Tian, and X. Zhu, "A new user similarity model to improve the accuracy of collaborative filtering," *Knowledge-Based Systems*, vol. 56, pp. 156–166, 2014. [Online]. Available: http://doi.org/10.1016/j.knosys.2013.11.006

Z. Liu, N. Meyendorf, and N. Mrad, "The role of data fusion in predictive maintenance using digital twin," in *AIP Conference Proceedings*, vol. 1949, no. April 2018. AIP Publishing LLC, 2018, p. 020023. [Online]. Available: https://doi.org/10.1063/1.5031520

Y. Lu, "Industry 4.0: A survey on technologies, applications and open research issues," *Journal of Industrial Information Integration*, vol. 6, pp. 1–10, 2017. [Online]. Available: http://doi.org/10.1016/j.jii.2017.04.005

H. Ma, H. Yang, M. R. Lyu, and I. King, "SoRec: Social Recommendation Using Probabilistic Matrix Factorization," in *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, 2008, pp. 931–940. [Online]. Available: https://doi.org/10.1145/1458082.1458205

H. Ma, I. King, and M. R. Lyu, "Learning to recommend with social trust ensemble," in *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2009, pp. 203–210. [Online]. Available: https://doi.org/10.1145/1571941.1571978

H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King, "Recommender systems with social regularization," in *Proceedings of the 4th ACM International Conference on Web Search and Data Mining*, 2011, pp. 287–296. [Online]. Available: https://doi.org/10.1145/1935826.1935877

X. Ma, H. Lu, Z. Gan, and J. Zeng, "An explicit trust and distrust clustering based collaborative filtering recommendation approach," *Electronic Commerce Research and Applications*, vol. 25, pp. 29–39, 2017. [Online]. Available: http://doi.org/10.1016/j.elerap.2017.06.005

M. Macchi, I. Roda, E. Negri, and L. Fumagalli, "Exploring the role of Digital Twin for Asset Lifecycle Management," *IFAC-PapersOnLine*, vol. 51, no. 11, pp. 790–795, 2018. [Online]. Available: https://doi.org/10.1016/j.ifacol.2018.08.415

P. Madhavan and D. A. Wiegmann, "Similarities and differences between human-human and human-automation trust: An integrative review," *Theoretical Issues in Ergonomics Science*, vol. 8, no. 4, pp. 277–301, 2007. [Online]. Available: https://doi.org/10.1080/14639220500337708

A. M. Madni, C. C. Madni, and S. D. Lucero, "Leveraging digital twin technology in model-based systems engineering," *Systems*, vol. 7, no. 1, pp. 1–13, 2019. [Online]. Available: https://doi.org/10.3390/systems7010007

M. Maida and N. Obwegeser, "The effect of sensitivity analysis on the usage of recommender systems," *RecSys Decisions 2012: Workshop on Human Decision Making in Recommender Systems In conjunction with the 6th ACM Conference on Recommender Systems*, pp. 15–18, 2012. [Online]. Available: https://api.semanticscholar.org/CorpusID:1768453

S. Malik, A. Rana, and M. Bansal, "A survey of recommendation systems," *Information Resources Management Journal*, vol. 33, no. 4, pp. 53–73, 2020. [Online]. Available: https://doi.org/10.4018/IRMJ.2020100104

J. Manyika, J. Sinclair, R. Dobbs, G. Strube, L. Rassey, J. Mischke, J. Remes, C. Roxburgh, K. George, D. O'Halloran, and S. Ramaswamy, "Manufacturing the future: The next era of global growth and innovation," 2012. [Online]. Available: https://www.mckinsey.com/capabilities/operations/our-insights/the-future-of-manufacturing

U. Marung, N. Theera-Umpon, and S. Auephanwiriyakul, "Applying memetic algorithm-based clustering to recommender system with high sparsity problem," *Journal of Central South University*, vol. 21, pp. 3541–3550, 9 2014. [Online]. Available: https://doi.org/10.1007/s11771-014-2334-4

M. Mashaly, "Connecting the twins: A review on digital twin technology its networking requirements," in *International Conference on Ambient Systems, Netwroks and Technologies*, vol. 184. Elsevier B.V., 2021, pp. 299–305. [Online]. Available: https://doi.org/10.1016/j.procs.2021.03.039

P. Massa and P. Avesani, "Trust-aware collaborative filtering for recommender systems," in *OTM Confederated International Conferences On the Move to Meaningful Internet Systems*, vol. 3290, 2004, pp. 492–508. [Online]. Available: https://doi.org/10.1007/978-3-540-30468-5_31

P. Massa and P. Avesani, "Trust-aware Recommender Systems," in *Proceedings of the 2007 ACM conference on Recommender systems*, 2007, pp. 17–24. [Online]. Available: https://doi.org/10.1145/1297231.1297235

P. Massa and P. Avesani, "Trust metrics in recommender systems," in *Computing with Social Trust*, J. Golbeck, Ed. London: Springer London, 2009, pp. 259–285. [Online]. Available: https://doi.org/10.1007/978-1-84800-356-9_10

A. Menk Dos Santos, "Personality-based recommendation: human curiosity applied to recommendation systems using implicit information from social networks," Ph.D. dissertation, Universitat Politècnica de València, 2018. [Online]. Available: https://dialnet.unirioja.es/servlet/tesis?codigo=250496&info=resumen&idioma=SPA%0Ahttps://dialnet.unirioja.es/servlet/tesis?codigo=250496

F. Meyer, "Recommender systems in industrial contexts," Ph.D. dissertation, Université de Grenoble, 2012. [Online]. Available: http://arxiv.org/abs/1203.4487

A. Moeuf, R. Pellerin, S. Lamouri, S. Tamayo-Giraldo, and R. Barbaray, "The industrial management of SMEs in the era of Industry 4.0," *International Journal of Production Research*, vol. 56, no. 3, pp. 1118–1136, 2018. [Online]. Available: https://doi.org/10.1080/00207543.2017.1372647

M. H. Mohamed, M. Khafagy, and H. Elbeh, "Sparsity and cold start recommendation system challenges solved by hybrid feedback," *International Journal of Engineering Research and Technology*, vol. 12, pp. 2734–2741, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:261465538

P. Moradi, S. Ahmadian, and F. Akhlaghian, "An effective trust-based recommendation method using a novel graph clustering algorithm," *Physica A: Statistical Mechanics and its Applications*, vol. 436, pp. 462–481, 2015. [Online]. Available: http://doi.org/10.1016/j.physa.2015.05.008

M. Nanthini and K. Pradeep Mohan Kumar, "Cold start and data sparsity problems in recommender system: A concise review," in *International Conference on Innovative Computing and Communications*, D. Gupta, A. Khanna, S. Bhattacharyya, A. E. Hassanien, S. Anand, and A. Jaiswal, Eds. Singapore: Springer Nature Singapore, 2023, pp. 107–118. [Online]. Available: https://doi.org/10.1007/978-981-19-2821-5_9

S. Natarajan, S. Vairavasundaram, S. Natarajan, and A. H. Gandomi, "Resolving data sparsity and cold start problem in collaborative filtering recommender system using linked open data," *Expert Systems with Applications*, vol. 149, 7 2020. [Online]. Available: https://doi.org/10.1016/j.eswa.2020.113248

D. Nawara and R. Kashef, "IoT-based recommendation systems - An overview," in *2020 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS)*, 2020, pp. 1–7. [Online]. Available: https://doi.org/10.1109/IEMTRONICS51293.2020.9216391

E. Negri, L. Fumagalli, and M. Macchi, "A Review of the Roles of Digital Twin in CPS-based Production Systems," in *27th International Conference on Flexible Automation and Intelligent Manufacturing*, vol. 11, no. June. The Author(s), 2017, pp. 939–948. [Online]. Available: http://doi.org/10.1016/j.promfg.2017.07.198

A. A. Neto, B. S. Carrijo, J. G. Romanzini Brock, F. Deschamps, and E. P. de Lima, "Digital twin-driven decision support system for opportunistic preventive maintenance scheduling in manufacturing," *Procedia Manufacturing*, vol. 55, no. C, pp. 439–446, 2021. [Online]. Available: https://doi.org/10.1016/j.promfg.2021.10.060

H. Nouinou, E. Asadollahi-Yazdi, I. Baret, N. Q. Nguyen, M. Terzi, Y. Ouazene, F. Yalaoui, and R. Kelly, "Decision-making in the context of Industry 4.0: Evidence from the textile and clothing industry," *Journal of Cleaner Production*, vol. 391, no. July 2022, p. 136184, 2023. [Online]. Available: https://doi.org/10.1016/j.jclepro.2023.136184

J. O'Donovan and B. Smyth, "Trust in recommender systems," in *Proceedings of the 10th International Conference on Intelligent User Interfaces*, 2005, pp. 167–174. [Online]. Available: https://doi.org/10.1145/1040830.1040870

Oracle, "Digital Twins for IoT Applications: A Comprehensive Approach to Implementing IoT Digital Twins," Oracle Corporation, Tech. Rep. January, 2017. [Online]. Available: https://www.oracle.com/assets/digital-twins-for-iot-apps-wp-3491953.pdf

S. Pal, A. Hill, T. Rabehaja, and M. Hitchens, "A blockchain-based trust management framework with verifiable interactions," *Computer Networks*, vol. 200, p. 108506, 2021. [Online]. Available: https://doi.org/10.1016/j.comnet.2021.108506

T. Y. Pang, J. D. P. Restrepo, C.-t. Cheng, A. Yasin, H. Lim, and M. Miletic, "Developing a Digital Twin and Digital Thread Framework for 'Industry 4.0' Shipyard," *Applied Sciences*, vol. 11, no. 1097, pp. 1–22, 2021. [Online]. Available: https://doi.org/10.3390/app11031097

A. Panteli and B. Boutsinas, "Addressing the cold-start problem in recommender systems based on frequent patterns," *Algorithms*, vol. 16, 4 2023. [Online]. Available: https://doi.org/10.3390/a16040182

M. Papagelis, D. Plexousakis, and T. Kutsuras, "Alleviating the sparsity problem of collaborative filtering using trust inferences," in *International Conference on Trust Management*, vol. 125, no. 5, 2005, pp. 224–239. [Online]. Available: https://doi.org/10.4018/ijssmet.2020100102

A. Parrott and W. Lane, "Industry 4.0 and the digital twin," *Deloitte University Press*, pp. 1–17, 2017. [Online]. Available: https://dupress.deloitte.com/dup-us-en/focus/industry-4-0/digital-twin-technology-smart-factory.html

K. Patel and H. B. Patel, "A state-of-the-art survey on recommendation system and prospective extensions," *Computers and Electronics in Agriculture*, vol. 178, no. August, p. 105779, 2020. [Online]. Available: https://doi.org/10.1016/j.compag.2020.105779

K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chartterjee, "A Design Science Research Methodology for Information Systems Research," *Journal of Management Information Systems*, vol. 24, no. 3, pp. 45–78, 2007. [Online]. Available: https://doi.org/10.2753/MIS0742-1222240302

R. S. Peres, A. Dionisio Rocha, P. Leitao, and J. Barata, "IDARTS – Towards intelligent data analysis and real-time supervision for industry 4.0," *Computers in Industry*, vol. 101, no. October, pp. 138–146, 2018. [Online]. Available: https://doi.org/10.1016/j.compind.2018.07.004

V. Peterka, "Predictor-Based Self-Tuning Control," *Automatica*, vol. 20, pp. 39–50, 1984. [Online]. Available: https://doi.org/10.1016/s1474-6670(17)63123-9

F. Pires, J. Barbosa, and P. Leitão, "Quo Vadis Industry 4.0: An overview Based on Scientific Publications Analytics," in *IEEE 27th International Symposium on Industrial Eletronics (ISIE'18)*, 2018, pp. 663–668. [Online]. Available: https://doi.org/10.1109/ISIE.2018.8433868

F. Pires, A. Cachada, J. Barbosa, A. P. Moreira, and P. Leitao, "Digital twin in industry 4.0: Technologies, applications and challenges," in *2019 IEEE 17th International Conference on Industrial Informatics (INDIN)*, vol. 2019-July, 2019, pp. 721–726. [Online]. Available: https://doi.org/10.1109/INDIN41052.2019.8972134

F. Pires, B. Ahmad, A. P. Moreira, and P. Leitao, "Recommendation system using reinforcement learning for what-if simulation in digital twin," in *IEEE International Conference on Industrial Informatics (INDIN)*, 2021, pp. 1–6. [Online]. Available: https://doi.org/10.1109/INDIN45523.2021.9557372

F. Pires, B. Ahmad, A. P. Moreira, and P. Leitao, "Digital twin based what-if simulation for energy management," in *2021 4th IEEE International Conference on Industrial Cyber-Physical Systems (ICPS)*, 2021, pp. 309–314. [Online]. Available: https://doi.org/10.1109/ICPS49255.2021.9468224

F. Pires, P. Leitão, A. P. Moreira, and B. Ahmad, "Reinforcement learning based trustworthy recommendation model for digital twin-driven decision-support in manufacturing systems," *Computers in Industry*, vol. 148, no. November 2022, 2023. [Online]. Available: https://doi.org/10.1016/j.compind.2023.103884

G. Pitsilis and L. F. Marshall, "A model of trust derivation from evidence for use in recommendation systems," 2004. [Online]. Available: https://api.semanticscholar.org/CorpusID:20542764

I. Portugal, P. Alencar, and D. Cowan, "The use of machine learning algorithms in recommender systems: A systematic review," *Expert Systems with Applications*, vol. 97, pp. 205–227, 2018. [Online]. Available: https://doi.org/10.1016/j.eswa.2017.12.020

A. Rasheed, O. San, and T. Kvamsdal, "Digital twin: Values, challenges and enablers from a modeling perspective," *IEEE Access*, vol. 8, pp. 21 980–22 012, 2020. [Online]. Available: https://doi.org/10.1109/ACCESS.2020.2970143

P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens: An Open Architecture for Collaborative Filtering of Netnews," in *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, 1994, pp. 175–186. [Online]. Available: https://doi.org/10.1145/192844.192905

F. Ricci, L. Rokach, and B. Shapira, "Introduction to recommender systems handbook," in *Recommender Systems Handbook*. Springer, 2010, pp. 1–35. [Online]. Available: https://doi.org/10.1007/978-0-387-85820-3

F. Ricci, L. Rokach, and B. Shapira, "Recommender systems handbook." Springer, 2015. [Online]. Available: https://doi.org/10.1007/978-1-4899-7637-6

K. V. Rodpysh, S. J. Mirabedini, and T. Banirostam, "Employing singular value decomposition and similarity criteria for alleviating cold start and sparse data in context-aware recommender systems," *Electronic Commerce Research*, vol. 23, pp. 681–707, 6 2023. [Online]. Available: https://doi.org/10.1007/s10660-021-09488-7

R. Rosen, G. Von Wichert, G. Lo, and K. D. Bettenhausen, "About the importance of autonomy and digital twins for the future of manufacturing," *IFAC-PapersOnLine*, vol. 28, no. 3, pp. 567–572, 2015. [Online]. Available: http://doi.org/10.1016/j.ifacol.2015.06.141

F. Rosin, P. Forget, S. Lamouri, and R. Pellerin, "Enhancing the Decision-Making Process through Industry 4.0 Technologies," *Sustainability (Switzerland)*, vol. 14, no. 1, 2022. [Online]. Available: https://doi.org/10.3390/su14010461

D. Roy and M. Dutta, "A systematic review and research perspective on recommender systems," *Journal of Big Data*, vol. 9, 12 2022. [Online]. Available: https://doi.org/10.1186/s40537-022-00592-5

R. A. Rupasingha and I. Paik, "Alleviating sparsity by specificity-aware ontology-based clustering for improving web service recommendation," *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 14, pp. 1507–1517, 10 2019. [Online]. Available: https://doi.org/10.1002/tee.22970

M. Rüßmann, M. Lorenz, P. Gerbert, M. Waldner, J. Justus, P. Engel, and M. Harnisch, "Industry 4.0: The Future of Productivity and Growth in Manufacturing Industries," Boston Consulting, Tech. Rep. April, 2015. [Online]. Available: https://doi.org/10.1007/s12599-014-0334-4

A. Said, "Evaluating the accuracy and utility of recommender systems," Ph.D. dissertation, Technischen Universität Berlin, 2013. [Online]. Available: https://api.semanticscholar.org/CorpusID:5226642

B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the 10th International Conference on World Wide Web, WWW 2001*, 2001, pp. 285–295. [Online]. Available: https://doi.org/10.1145/371920.372071

V. K. Sejwal and M. Abulaish, "A hybrid recommendation technique using topic embedding for rating prediction and to handle cold-start problem," *Expert Systems with Applications*, vol. 209, 12 2022. [Online]. Available: https://doi.org/10.1016/j.eswa.2022.118307

A. Selmi, Z. Brahmi, and M. M. Gammoudi, "Trust-based recommender systems: An overview," in *Proceedings of the 27th International Business Information Management Association Conference (IBIMA)*, 2016, pp. 371–380.

Y. Seong, A. M. Bisantz, and G. J. Gattie, *Trust, Automation, and Feedback: An Integrated Approach*. Oxford University Press, 2006. [Online]. Available: https://api.semanticscholar.org/CorpusID:64438523

M. Shafto, M. Conroy, R. Doyle, E. Glaessgen, C. Kemp, J. LeMoigne, and L. Wang, "Modeling, Simulation, Information Technology & Processing Roadmap," *National Aeronautics and Space Administration*, 2010. [Online]. Available: https://api.semanticscholar.org/CorpusID:197862517

Q. Shambour and J. Lu, "A trust-semantic fusion-based recommendation approach for e-business applications," *Decision Support Systems*, vol. 54, no. 1, pp. 768–780, 2012. [Online]. Available: https://doi.org/10.1016/j.dss.2012.09.005

G. Shani and A. Gunawardana, "Evaluating Recommendation Systems," *Recommender Systems Handbook*, pp. 257–297, 2011. [Online]. Available: https://doi.org/10.1007/978-0-387-85820-3_8

M. Sharma and S. Mann, "A Survey of Recommender Systems: Approaches and Limitations," *International Journal of Innovations in Engineering and Technology*, vol. 2, no. 2, pp. 8–14, 2013. [Online]. Available: https://pdfs.semanticscholar.org/fa41/dc4b60eccedf1c41e2ae488044827dd79384.pdf

R. Sharma and R. Singh, "Evolution of recommender systems from ancient times to modern era: A survey," *Indian Journal of Science and Technology*, vol. 9, no. 20, pp. 1–12, 2016. [Online]. Available: https://doi.org/10.17485/ijst/2016/v9i20/88005

G. Shaw, Y. Xu, and S. Geva, "Using association rules to solve the cold-start problem in recommender systems," in *Advances in Knowledge Discovery and Data Mining: 14th Pacific-Asia Conference, PAKDD 2010, Hyderabad, India, June 21-24, 2010. Proceedings. Part I 14*, M. Zaki, J. Yu, B. Ravindran, and V. Pudi, Eds., vol. 6118. Springer, 2010, pp. 340–347. [Online]. Available: https://doi.org/10.1007/978-3-642-13657-3_37

S. Sheibani, H. Shakeri, and R. Sheibani, "Four-dimensional trust propagation model for improving the accuracy of recommender systems," *Journal of Supercomputing*, vol. 79, pp. 16 793–16 820, 10 2023. [Online]. Available: https://doi.org/10.1007/s11227-023-05278-0

W. Shi, L. Wang, and J. Qin, "Extracting user influence from ratings and trust for rating prediction in recommendations," *Scientific Reports*, vol. 10, 12 2020. [Online]. Available: https://doi.org/10.1038/s41598-020-70350-1

L. C. Siafara, H. A. Kholerdi, A. Bratukhin, N. TaheriNejad, A. Wendt, A. Jantsch, A. Treytl, and T. Sauter, "Samba: A self-aware health monitoring architecture for distributed industrial systems," in *IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society*. IEEE Press, 2017, p. 3512–3517. [Online]. Available: https://doi.org/10.1109/IECON.2017.8216594

A. Simeone, Y. Zeng, and A. Caggiano, "Intelligent decision-making support system for manufacturing solution recommendation in a cloud framework," *International Journal of Advanced Manufacturing Technology*, vol. 112, no. 3-4, pp. 1035–1050, 2021. [Online]. Available: https://doi.org/10.1007/s00170-020-06389-1

P. K. Singh, P. K. D. Pramanik, G. Ahuja, A. Nayyar, V. Pandey, and P. Choudhury, "Mitigating sparsity and cold start problem in collaborative filtering using cross-domain similarity," in *2020 8th International Conference on Orange Technology, ICOT 2020*. Institute of Electrical and Electronics Engineers Inc., 12 2020. [Online]. Available: https://doi.org/10.1109/ICOT51877.2020.9468770

H. Sobhanam and A. Mariappan, "Addressing cold start problem in recommender systems using association rules and clustering technique," in *2013 International Conference on Computer Communication and Informatics*. IEEE, 2013, pp. 1–5. [Online]. Available: https://doi.org/10.1109/ICCCI.2013.6466121

L. H. Son, "Dealing with the new user cold-start problem in recommender systems: A comparative review," *Information Systems*, vol. 58, pp. 87–104, 2016. [Online]. Available: http://doi.org/10.1016/j.is.2014.10.001

H. Song, Q. Pei, Y. Xiao, Z. Li, and Y. Wang, "A Novel Recommendation Model Based on Trust Relations and Item Ratings in Social Networks," in *2017 International Conference on Networking and Network Applications, (NaNA)*, 2017, pp. 17–23. [Online]. Available: https://doi.org/10.1109/NaNA.2017.17

M. Soori, B. Arezoo, and R. Dastres, "Artificial intelligence, machine learning and deep learning in advanced robotics, a review," *Cognitive Robotics*, vol. 3, pp. 54–70, 2023. [Online]. Available: https://doi.org/10.1016/j.cogr.2023.04.001

R. K. Sorde and S. N. Deshmukh, "Comparative Study on Approaches of Recommendation Systems," *International Journal of Computer Applications*, vol. 118, no. 2, pp. 10–14, 2015. [Online]. Available: https://doi.org/10.1007/978-981-15-0947-6_72

P. K. Sowell, "The C4ISR Architecture Framework: History, Status, and Plans for Evolution," MITRE Corporation Mclean VA, Virginia, Tech. Rep., 2006. [Online]. Available: http://www.mitre.org/work/tech_papers/tech_papers_00/sowell_evolution/sowell_evolution.pdf%5Cnhttp://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA456187

R. Stark, S. Kind, and S. Neumeyer, "Innovations in digital modelling for next generation manufacturing system design," *CIRP annals*, vol. 66, no. 1, pp. 169–172, 2017. [Online]. Available: https://doi.org/10.1016/j.cirp.2017.04.045

T. Stock and G. Seliger, "Opportunities of Sustainable Manufacturing in Industry 4.0," in *13th Global Conference on Sustainable Manufacturing in Industry 4.0*, vol. 40. Elsevier B.V., 2016, pp. 536–541. [Online]. Available: http://doi.org/10.1016/j.procir.2016.01.129

M. Sun, F. Li, and J. Zhang, "A multi-modality deep network for cold-start recommendation," *Big Data and Cognitive Computing*, vol. 2, pp. 1–15, 3 2018. [Online]. Available: https://doi.org/10.3390/bdcc2010007

M. Sutrisna, "Research methodology in doctoral research: Understanding the meaning of conducting qualitative research," in *Association of Researchers in Construction Management (ARCOM) Doctoral Workshop*, vol. 2, 2009, pp. 48–57.

R. S. Sutton and A. G. Barto, "Reinforcement learning: An introduction," in *MIT Press*. MIT Press, 1998. [Online]. Available: https://doi.org/10.1108/k.1998.27.9.1093.3

J. Tang, X. Hu, and H. Liu, "Social recommendation: a review," *Social Network Analysis and Mining*, vol. 3, no. 4, pp. 1113–1133, 2013. [Online]. Available: 10.1007/s13278-013-0141-9

F. Tao and M. Zhang, "Digital Twin Shop-Floor: A New Shop-Floor Paradigm Towards Smart Manufacturing," *IEEE Access*, vol. 5, pp. 20 418–20 427, 2017. [Online]. Available: https://doi.org/10.1109/ACCESS.2017.2756069

F. Tao, Q. Qi, L. Wang, and A. Y. Nee, "Digital Twins and Cyber–Physical Systems toward Smart Manufacturing and Industry 4.0: Correlation and Comparison," *Engineering*, vol. 5, no. 4, pp. 653–661, 2019. [Online]. Available: https://doi.org/10.1016/j.eng.2019.01.014

F. Tao, H. Zhang, A. Liu, and A. Y. Nee, "Digital Twin in Industry: State-of-the-Art," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 4, pp. 2405–2415, 2019. [Online]. Available: https://doi.org/10.1109/TII.2018.2873186

J. K. Tarus, Z. Niu, and G. Mustafa, "Knowledge-based recommendation: a review of ontology-based recommender systems for e-learning," *Artificial Intelligence Review*, vol. 50, no. 1, pp. 21–48, 2018. [Online]. Available: https://doi.org/10.1007/s10462-017-9539-5

F. J. Tey, T. Y. Wu, C. L. Lin, and J. L. Chen, "Accuracy improvements for cold-start recommendation problem using indirect relations in social networks," *Journal of Big Data*, vol. 8, no. 98, 2021. [Online]. Available: https://doi.org/10.1186/s40537-021-00484-0

D. Thomas, "National institute of standards and technology (nist)," 2023. [Online]. Available: https://www.nist.gov/el/applied-economics-office/manufacturing/manufacturing-economy/total-us-manufacturing

B. Thorat, R. M. Goudar, and S. Barve, "Survey on Collaborative Filtering, Content-based Filtering and Hybrid Recommendation System," *International Journal of Computer Applications*, vol. 110, no. 4, pp. 31–36, 2015. [Online]. Available: https://doi.org/10.5120/19308-0760

E. J. Tuegel, A. R. Ingraffea, T. G. Eason, and S. M. Spottswood, "Reengineering aircraft structural life prediction using a digital twin," *International Journal of Aerospace Engineering*, vol. 2011, pp. 1–14, 2011. [Online]. Available: https://doi.org/10.1155/2011/154798

M. C. Urdaneta-Ponte, A. Mendez-Zorrilla, and I. Oleagordia-Ruiz, "Recommendation systems for education: Systematic review," *Electronics*, vol. 10, no. 14, p. 1611, 2021. [Online]. Available: https://doi.org/10.3390/electronics10141611

S. Vaidya, P. Ambad, and S. Bhosle, "Industry 4.0 - A Glimpse," in *2nd International Conference on Materials Manufactruing and Design Engineering*, vol. 20. Elsevier B.V., 2018, pp. 233–238. [Online]. Available: https://doi.org/10.1016/j.promfg.2018.02.034

E. VanDerHorn and S. Mahadevan, "Digital Twin: Generalization, characterization and implementation," *Decision Support Systems*, vol. 145, no. February, p. 113524, 2021. [Online]. Available: https://doi.org/10.1016/j.dss.2021.113524

P. Victor, M. De Cock, and C. Cornelis, "Trust and Recommendations," in *Recommender Systems Handbook*. Springer, 2011, pp. 645–675. [Online]. Available: https://doi.org/10.1007/978-0-387-85820-3_20

A. L. Vizine Pereira and E. R. Hruschka, "Simultaneous co-clustering and learning to address the cold start problem in recommender systems," *Knowledge-Based Systems*, vol. 82, pp. 11–19, 2015. [Online]. Available: http://doi.org/10.1016/j.knosys.2015.02.016

J. Wei, J. He, K. Chen, Y. Zhou, and Z. Tang, "Collaborative filtering and deep learning based recommendation system for cold start items," *Expert Systems with Applications*, vol. 69, pp. 1339–1351, 2017. [Online]. Available: http://doi.org/10.1016/j.eswa.2016.09.040

L.-T. Weng, "Information enrichment for quality recommender systems," Ph.D. dissertation, Queensland University of Technology, 2008. [Online]. Available: https://api.semanticscholar.org/CorpusID:60663299

L. Xiao, "Trust quantitative model with multiple decision factors in trusted network," *Chinese Journal of Computers*, 2009. [Online]. Available: https://api.semanticscholar.org/CorpusID:63629790

Y. Xin, "Challenges in recommender systems: Scalability, privacy, and structured recommendations," Ph.D. dissertation, Massachusetts Institute of Technology, 2015. [Online]. Available: http://hdl.handle.net/1721.1/99785

L. D. Xu, E. L. Xu, and L. Li, "Industry 4.0: state of the art and future trends," *International Journal of Production Research*, vol. 56, no. 8, pp. 2941–2962, 2018. [Online]. Available: https://doi.org/10.1080/00207543.2018.1444806

B. Yang, Y. Lei, J. Liu, and W. Li, "Social collaborative filtering by trust," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 8, pp. 1633–1647, 2017. [Online]. Available: https://doi.org/10.1109/TPAMI.2016.2605085

L. Yinggang and T. Xiangrong, "Social recommendation system based on multi-agent deep reinforcement learning," in *2022 IEEE 8th International Conference on Cloud Computing and Intelligent Systems (CCIS)*, 2022, pp. 371–377. [Online]. Available: https://doi.org/10.1109/CCIS57298.2022.10016386

Yokogawa, "Digital Twin: The Key to Effective Decision-Making," p. 12, 2019. [Online]. Available: https://web-material3.yokogawa.com/15/28517/overview/DigitalTwinWhitePaperX09.pdf

W. Yuan, D. Guan, Y. K. Lee, S. Lee, and S. J. Hur, "Improved trust-aware recommender system using small-worldness of trust networks," *Knowledge-Based Systems*, vol. 23, pp. 232–238, 4 2010. [Online]. Available: https://doi.org/10.1016/j.knosys.2009.12.004

D. Zhang, C. H. Hsu, M. Chen, Q. Chen, N. Xiong, and J. Lloret, "Cold-start recommendation using bi-clustering and fusion for large-scale social recommender systems," *IEEE Transactions on Emerging Topics in Computing*, vol. 2, pp. 239–250, 6 2014. [Online]. Available: https://doi.org/10.1109/TETC.2013.2283233

H. Zhang, G. Liu, and J. Wu, "Social collaborative filtering ensemble," in *PRICAI 2018: Trends in Artificial Intelligence*, X. Geng and B.-H. Kang, Eds. Springer International Publishing, 2018, pp. 1005–1017. [Online]. Available: https://doi.org/10.1007/978-3-319-97304-3_77

Q. Zhang, J. Lu, and Y. Jin, "Artificial intelligence in recommender systems," *Complex Intelligent Systems*, vol. 7, pp. 439–457, 2021. [Online]. Available: https://doi.org/10.1007/s40747-020-00212-w

W. Y. Zhang, S. Zhang, Y. G. Chen, and X. W. Pan, "Combining social network and collaborative filtering for personalised manufacturing service recommendation," *International Journal of Production Research*, vol. 51, pp. 6702–6719, 11 2013. [Online]. Available: https://doi.org/10.1080/00207543.2013.832839

X. Zhang, J. Cheng, S. Qiu, G. Zhu, and H. Lu, "Dualds: A dual discriminative rating elicitation framework for cold start recommendation," *Knowledge-Based Systems*, vol. 73, pp. 161–172, 1 2015. [Online]. Available: https://doi.org/10.1016/j.knosys.2014.09.015

Z. Zhang, Y. Zhang, and Y. Ren, "Employing neighborhood reduction for alleviating sparsity and cold start problems in user-based collaborative filtering," *Information Retrieval Journal*, vol. 23, pp. 449–472, 8 2020. [Online]. Available: https://doi.org/10.1007/s10791-020-09378-w

H. Zheng, D. Wang, Q. Zhang, H. Li, and T. Yang, "Do clicks measure recommendation relevancy? an empirical user study," in *Proceedings of the Fourth ACM Conference on Recommender Systems*, ser. RecSys '10. New York, NY, USA: Association for Computing Machinery, 2010, p. 249–252. [Online]. Available: https://doi.org/10.1145/1864708.1864759

Y. Zhou, Z. Tang, L. Qi, X. Zhang, W. Dou, and S. Wan, "Intelligent service recommendation for cold-start problems in edge computing," *IEEE Access*, vol. 7, pp. 46 637–46 645, 2019. [Online]. Available: https://doi.org/10.1109/ACCESS.2019.2909843

Y. Zuo, J. Zeng, M. Gong, and L. Jiao, "Tag-aware recommender systems based on deep neural networks," *Neurocomputing*, vol. 204, pp. 51–60, 2016, big Learning in Social Media Analytics. [Online]. Available: https://doi.org/10.1016/j.neucom.2015.10.134

Appendix A

This section contains the URL for the GitHub repository where the SimQL approach code that was developed is presented (https://gitfront.io/r/fpires1993/mvp6m5t1UTBp/SimQL/).